IVC Course Code : 319

COMPUTER SCIENCE & ENGINEERING
SECOND YEAR

(w.e.f.2019 – 20)

Intermediate Vocational Course

Paper I : OOPS & JAVA

Paper II : RELATIONAL DATABASE & MANAGEMENT SYSTEM

Paper III : DATA COMMUNICATIONS AND COMPUTER NETWORKS

STATE INSTITUTE OF VOCATIONAL EDUCATION

BOARD OF INTERMEDIATE EDUCATION, A.P.

Text Book Development Committee

Paper -I OOPS & JAVA

AUTHOR

N.SRIRAMA MURTHY
Junior Lecturer in CS & E
Govt. Junior College For Girls,
CHIRALA

Paper-II RELATIONAL DATABASE MANGAGEMENT SYSTEM

AUTHOR

V.R.S.K.S. SASTRY TANIKELLA
Junior Lecturer in CS & E
Govt. Junior College Kothapeta
EAST GODAVARI

Paper-III DATA COMMUNICATIONS AND COMPUTER NETWORKS

AUTHOR

I. **Murali Krishna,** M.Tech
Asst.Professor
PSCMR College of Engineering And Technology,
VIJAYAWADA

EDITOR

Dr.A.Pathanjali Sastri
Professor & HOD of CS & E
PSCMR College of Engineering & Technology
VIJAYAWADA

COMPUTER SCIENCE & ENGINEERING

TEXT BOOK DEVELOPMENT COMMITTEE

S.No	Name	Designation	Signature
1	Dr. A. Pathanjali Sastri Professor & HOD of CS & E PSCMR College of Engineering & Technology VIJAYAWADA	Editor	
2	N.Srirama Murthy Junior Lecturer in CS & E Govt. Junior College For Girls, CHIRALA	Author	
3	V.R.S.K.S.Sastry Tanikella Junior Lecturer in CS & E Govt. Junior College Kothapeta EAST GODAVARI	Author	
4.	I. Murali Krishna Asst.Professor PSCMR College of Engineering And Technology, VIJAYAWADA	Author	

Smt. B.UDAYA LAKSHMI, I.A.S.

Commissioner & Secretary

Intermediate Education

ANDHRA PRADESH

GUNTUR

S.I.V.E Co – Ordinating Committee

Sri. Podili Yerraiah, **M.Sc., B.Ed.**
Professor
State Institute of Vocational Education
Commissioner of Intermediate Education, Guntur

Sri. B. Nageswara **B.Com, B.L.,**
Joint Secretary (Vocational)
Board of Intermediate Education, Vijayawada

Sri. Dr.G.V.S.R. Murthy, **M.SC., Ph.D.**
Lecturer
State Institute of Vocational Education
Commissioner of Intermediate Education, Guntur

DTP

Thuraka Ravi Kumar ***BA,B.Ed., PGDCA***

COMPUTER SCIENCE & ENGINEERING

Paper – I

OOPS & JAVA

INDEX

UNIT 1

Object Oriented Programming And Java

Object Oriented Programming and Java
History of Java
Need of Object Oriented Programming
OOP Concepts
- Object
- Class
- Inheritance
- Polymorphism
- Data abstraction
- Encapsulation

Application of Object Oriented Programming Language.
Features of Java
- Rubust and Security
- Distributed
- Multi threading
- Dynamic Binding

JVM, JRE, JIT, Bite code

JVM

JRE

JIT

Byte Code

Java and Internet

Java and World Wide Web

Steps of Communicate Applets

Web Browsers

- Hot Java
- Chrome Opera
- Mozilla Firefox
- Netscape Navigator

Java Environment

Documentation Section

Package Statements

Import Statements

Interface Statements

Class Definition

1.8.1. Java Simple Program

Installation of JDK

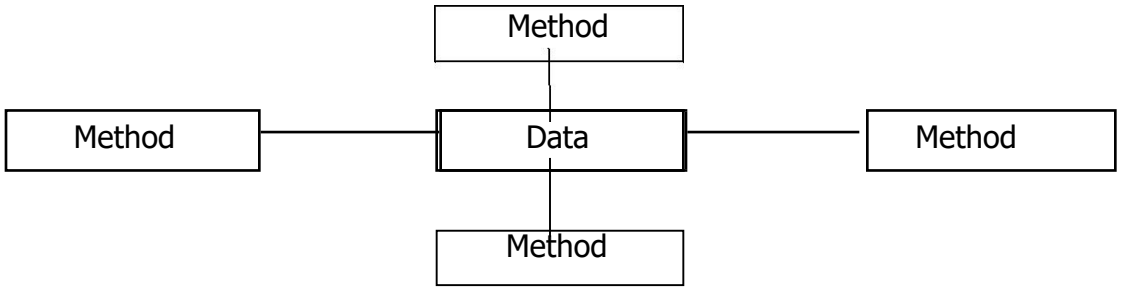
To Install Java

Introduction

Object oriented programming approach in eliminates some of the errors encountered in the procedure programming approach object oriented program threats that's data as a critical element in the program development and does not allow it to flow freely around the system it binds data more closely to the functions that operate on it and protects it from outside. Object oriented programming allows us to decompose a problem into a number of entities called object and then build data and functions around these entities.

The data of an object can be accessed by line methods associated with that object. The methods of the object access the methods of other object. Some of the features of the object oriented programs are:

(a) Programs are divided into where are known as objects
(b) Data structures are designed such that they characterize the objects.
(c) Methods that Operate on the data of an object are joined together in the data Structure.
(d) Data is hidden and cannot be accessed by external functions.
(e) Objects may communicate with each other through methods.
(f) New data and methods can be easily added whenever necessary.
(g) It follows bottom-up approach in program design



Object = data + methods

History of JAVA:

The history of Java starts from "Green team, Java team members initiate a revolutanary task to develop a language for digital devices such as set-top boxes, televisions etc. for the Green team members, it was an advance concept at that time, currently Java is used in internet programming mobile device, games, e-business solutions etc. there are given the major points that describe the history of Java. **"James Gosling" "Mike Sheridan"** and **Patrick "Naughton"** initiated the Java Language Project in June 1991. The small team of Sun Engineers called **"Green Team"**. Java is a general – purpose object – oriented programming language developed by "Sun Micro Systems" of U.S.A. in 1991. Originally it is called **OAK** by **James**

Gosling is one of the inventors of the Language Java that was designed for the development of Software for consumer Electronic Mechines. In 1995 Oak was renamed "Java" Sun Micro System released Java Development Kit 1.0 in 1999. Then released Java 2 plat form standard edition (J2SE) and Enterprise Edition (J2EE). In 2004 J2SE with JDK 5.0 was released it is called as J2SE 5.0. The most striking feature of the Language is that a plat form – independent language. This means the Java Program can be run on any platform without any modification of the code.

Need of Object Oriented Programming :

An object Oriented Programming is a way which enables programmers to think they are working with reality entities or objects. In OOP, Objects have fields to store knowledge or State or Data and can do various works and methods. It helps to reduce the complexity and also improves the maintainability of the system. When combined with the concepts of the Encapsulation and polymorphism, abstraction, give more power to the object oriented programming languages.

Benefits of oops

Oops of refers several benefits of both programer and user.

a. It provides good productivity better quality of soft ware and lesser maintenance cost.

b. Through inheritance, we can eliminate redundant code and extend line use of existence code.

c. The principle of data hiding helps the programmer to build secure programs that can not be involved by code to coexist without any interference.

d. It is possible to have multiple objects in the problem domain to those objects in the program.

e. The data centered design approach enables us to capture more details of a model in an implemented form .

f. Systems can be easily upgraded from small to large system.

g. Message passing techniques for communication between objects make the interface descriptions with external systems much simpler.

h. Software complexity can be easily managed.

OOP's Concepts :

OBJECT ORINENTED PROGRAMMING CONCEPTS:

In this we will learn about the basics of OOP's provides many concepts such as object means a real world entity such a pen chair, table, computer watch etc. object oriented programming is methodology or Para diagram to design a program using classes and objects. It simplifies the software development and maintains by providing same concepts.

Object
Class
Inheritance
Polymorphism
Abstraction
Encapsulation

Object : An entity that has state and behavior is known as an object. For example a chair, pen, table key board. Bike etc. it can be physical or logical. An object can be defined as an instance of a class. An object contains an address and takes up some space in memory. Objects can communicated without knowing the details of each other's data or code. The only necessary thing is the type of message accepted and the type of response returned by the object.
Example: A dog is an object because it her states like color, name, breed etc. As well as behaviors like wagging the trail, barking, eating etc.

Class : collection of objects is called class. It is a logical entity.
A class can also be defined as a blueprint from which you can create an individual object class doesn't consume any space.

Inheritance : when one object acquires all the properties and behavior of parent object. It is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism : If one task is performed by different ways. It is known as polymorphism. For example to convince the customer differently to draw some thing. For example : shape, triangle, rectangle etc.

In java example we use method overloading and method overloading to achieve polymorphism.

Data abstraction : There may be a class contains a lot of data and the user does not need the entire data. The user require only some part of the available data. In this case, we can hide the unless any data from the user and expose only that data is of interest to user. This is called abstraction. A good example for abstraction is a car. Any car will have some parts like engine radiator, mechanical and electrical equipment etc. The user of the car should known how to drive the car and does not acquire any knowledge of these parts.

Encapsulation : Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a giving procedure call is not known until the time of the call at runtime.

An OOPs program consists of a set of object that communicate with each other by using following steps.

a. Creating classes that define objects and their behavior.

b. Creating object from class definition.

c. Establishing communication among object.

d. Object communicate with one another by sending and receiving information much the same way as people pals message to one another.

   Example:-salary particular of a employee.

   Employee – Object

   Salary – Message

   Name – information

1-5 Application of object oriented programming language

Object oriented programming language (OOP) is a style of programming just like functional programming or procedural programming you can apply a style to any programming problem you are trying to solve.

OOP is a most appropriate for writing large complex software. That's because it was expressly created to help you manage program complexity. It does this by dividing you program into smaller, more manageable units that consists of private state (known as instant variable)and public behavior(represented by functions called methods). Objects collaborate with one another by calling each other methods (this is typically done by sending message).If choose to employee inheritance, you can also improve the reusability of your code. This can make also employee aggregation or composition.

1.6 Features of java

Java programs can be easily moved from one computer system to another anywhere and anytime changes and upgrades in the system will not effect to change to source code. By this reason java has become so popular for programming on internet we can download applet from a remote computer into our local system via internet and execute it locally. Java ensure portability in two ways

Java compiler generates byte code instruments that can be implemented on any machine.

The size of the primitive data types are machine implemented.

Simple: Java is a small and simple language many feature of c and c++ that are either redundant or source of unreliable code are not part of java. Java uses many constructs of c and c++.

Object: Java is true object oriented language . Almost everything in java is an object. All program code and data reside within objects and classes. Java comes within are extensive set of classes, arranged in package. The object model in java simple java and easy to extend.

Robust and Security : Java is robust language. It provides many sage guards to ensure reliable code. It has strict compile time and runtime checking for data types. It maintains garbage collection. Which is used to maintain memory efficiently and exception handling in used to capture errors and eliminates any risk of crashing system.

Java system not only verifies memory access but also ensure that no viruses are communicated with an applet. The absence of painters in java ensures that programs cannot gain access without proper authorization.

Distributed: Java is designed as distributed for creating applications on networks. It has ability to share both data and programs. Java applications can open and access remote objects an Internet as easily as they can do in a local system.

Platform independent: java source code can run on all operating systems.

Multithreading: Multithreading means handling multiple tasks simultaneously. This means that we need not wait for the application to finish on task before beginning another. This features increases the interactive performance of graphic applications High performance according to sun Micro System Java speed is similar to native C or C++. Java architecture is also designed to reduce over heads during runtime.

Dynamic Binding: Java is a dynamic language. It is capable of dynamically linking is new class libraries, methods and objects. Java program support function written in other languages like C and C++ these functions are known as native method s. these are Binding or linked dynamically at run time.

Interpreted: Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an invented and light weight process.

Platform Independent : Java Programs can be easily moved from one computer system to another, anywhere and any time. Changes and upgrades in the system will not effect to change the source code. By this reason java has become so popular for

programming on Internet. We can download applet from a remote computer into our local system via Internet and execute it locally. Java ensures portability in two ways.

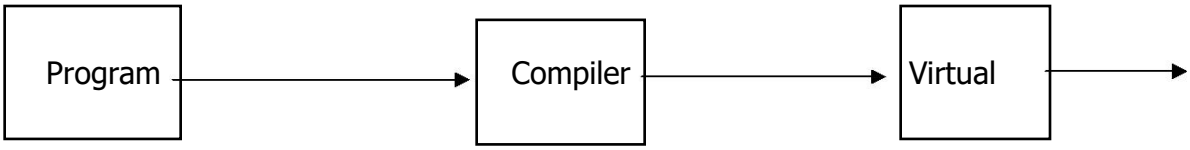
Java compiler generates byte code instructions that can be implemented on any machine.

The size of the primitive data types are machine independent.

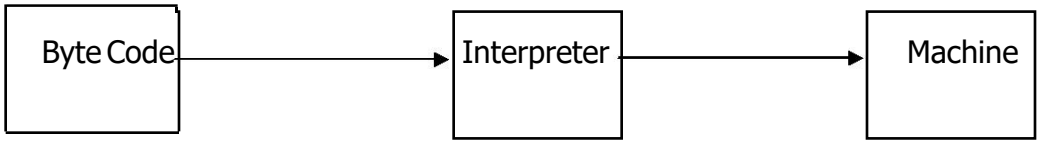
JVM, JRE, JIT, BITE CODE:

JVM : Java virtual machine is virtual machine that can executes Java byte Code. It is the Code execution component of the Java Software platform. The Java virtual machine is responsible for interpreting Java bite code and translating. Java compiler produce an intermediate code known as byte code for a machine that do not exist. This machine is called the java virtual machine and it exists only inside the computer with the computer memory. It is simulated computer with the computer and does it major functions of a real computer.

The virtual machine code is not machine specific. The machine specific code is generated by the Java interpreter by as mediator between the virtual machine and real machine. The interpreter is different machine.



Source Program:



JRE : Java run time environment is also called Java run time. It is a part of Java development kit (JDK). A set of programming tools for developing Java Application. Java runtime environment provides the minimum requirements for executing a Java Application. It consists of Java virtual machine (JVM) and classes are supporting files. It support Libraries. JRE was originally developed by Sun Micro Systems. JRE was originally developed by Sun Micro Systems. The Java Runtime Environment is a software layer that runs on top of a computer's operating system, providing additional services specific to Java.

JRE consists of the following components:

a. Deployment technologies, including deployment, Java Web Start and Java Plug-in.
b. User interface toolkits, including Abstract Window Toolkit (AWT), Swing, Java 2D, Accessibility, Image I/O, Print Service, Sound, drag and drop (DnD) and input methods.
c. Integration libraries, including Interface Definition Language (IDL), Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI), Remote Method Invocation (RMI), Remote Method Invocation Over Internet Inter-Orb Protocol (RMI-IIOP) and scripting.
d. Other base libraries, including international support, input/output (I/O), extension mechanism, Beans, Java Management Extensions (JMX), Java Native Interface (JNI), Math, Networking, Override Mechanism, Security, Serialization and Java for XML Processing (XML JAXP).
e. Lang and util base libraries, including lang and util, management, versioning, zip, instrument, reflection, Collections, Concurrency Utilities, Java Archive (JAR), Logging, Preferences API, Ref Objects and Regular Expressions.
f. Java Virtual Machine (JVM), including Java HotSpot Client and Server Virtual Machines.

J.I.T : In computing JIT compilation also known as dynamic Translation, it a method to improve to runtime performance of computer programs based on the byte code. Since byte code is interpreted it executes more slowly than compiled machine code, unless it is actually compiled to machine code, which could be performed before the execution, making the program loading slow.

In Java programming language and environment a just-in-time. Compiler is a program that turns java byte code. (a program that contains instructions that must be interpreted) into instructions that can be sent directly to the processor. After you have written a Java program, the source language statements are compiled by Java Compiler into byte code rather than into code that contains instructions that match a particular hardware platform's processor.

The byte code is platform independent code that can be sent to any platform and run on that platform. Most of program written in any languages have to be recompiled and sometimes rewritten for each computer platform. One of the biggest advantages of java is that you only have write and compile a program once. The java on any platform will interpret the compiled byte code into instructions understandable by the particular processor. However the virtual machine handles one byte code instructions at a time. Using the java just-in-time compiler at the particular system platform compiles the byte code has been re-compiles by the JIT compiler, it will usually run more quickly is the computer.

The just- in-time compiler comes with the virtual machine and is used optionally. It compiles the bytecode into platform. Specific executable code that is immediately executed. The Sun Micro Systems that it's usually faster to select the JIT compiler option, especially if the method executable is repeatedly reused.

1.7.4: Byte Code: Java byte code is the form of instruction of JVM executes. Each by code is one byte lengths although some requires parameter resulting is some multi-byte instructions.

Java byte code is the result of the compilation of a Java program. It is intermediate representation of that program. This is machine independent. The java bytecode gets processed by the Java virtual machine (JVM) instead of the processor. It is the job of the JVM to make the necessary resource calls to the processor. It is in order to run by bytecode. Java byte code is the resulting compiled object run in any platform which has a Java installation in it. Java program does not have to be knowledgeable about the quick's.

Java and Internet : Java is strongly associated with the internet because of the fact that the fact that first application program written in Java was Hot java, a web browser to run applet on internet. Internet users can use java to create applets programs and run them locally using a "Java enabled browser" such as Hot java. They can also use Java enables browser to download an applet located on a computer anywhere in the internet and run it on his local computer. Internet users can also setup their web sites containing java applets that could be used by other remote users of internet. Java is popularly known as Internet language.

Java an World Wide Web : It is an open-ended information retrieval system design to be used internet's distributed environment. The system contains web pages that provide both information and controls. We can navigate a new document in any directions. This is made with the help of HTML.HTML tags enable us to find, retrieve, manipulate and display document worldwide.

Java could be easily incorporated into web system. Before Java the WWW was limited to display images and texts. By using Java animations, graphics, games etc. Increase the range of www.Java communicates with the web pages through a special tag call <APPLET>

Steps to Communicate Applets:

a. User send requests for an HTML document to remote computer web server. Web server is a program that accepts request, process the request and sends the required document.
b. To the user's browser the document contains applet tag which defines the applet.
c. The Java enable browser on the computer interprets to byte code and provides output.
d. The corresponding applet byte code is returns to the user computer.
e. The user may have further interaction with the applet but with no further down loading from the provider's web server. This is because the byte code contains the all the information necessary to interpret the applet.

Web Browsers: Web browsers are used to pass through the information found on the net. They allow us to retrieve information across the internet and display it using the HTML.

Example of Browsers:
- Hot Java
- Chrome Opera
- Mozilla Firefox
- Internet Explorer

Hot Java : It is the web browser from Sun Micro System that enables to the display of interactive content on the using the java language. It demonstrates the capability of Java PL. Hot java is currently available for the SPARC/Solar platforms as well as windows 95, NT, XP.

Chrome Opera : Opera is a web browser and internet suite developed by opera software. It began in 1994.

Mozilla Firefox : It is another popular browser developed by Micro soft for Windows-95, NT and XP work stations.

Both the Navigator and explorer used tool bars, icons menus and dialog boxes for easy navigation. Explorer uses just in time compiler; which increase the speed of execution.

Netscape navigator : This web browser from Netscape communication corporation is a general – purpose browser that can run java applet. It is available for Windows -95, NT, XP, Solar and apple Macinthosh. It is one of the mostly used browser. The main features is displaying the information about downloading process and indication of numbers of bytes downloaded. It is also support Java script used in HTML document.

7.9. Java Environment : It includes large number of development tools and hundreds of classes and methods. The development tools are known as JDK (Java Development Kit) classes and methods are the part of Java standard Library (JSL) also known as API (Application Programming Interface).

JDK : It is a collection of tools for developing and running the Java programs.

Tool	Description
Applet Viewer	Enables us to run java applet.
Java	Java interpreter, which runs applets and applications
Javac	Java compiler translates Java source code into byte code.
Javadoc	Creates HTML format documentation from source code.
Javah	Produce header files for use with native methods
Javap	Java dissembler converts by the byte code into program description.
jdp	Java debugger helps us to find errors in the program.

Overview of Java Languages :

Java is a general purpose OOPL. Are can develop 2 types.

a). Standard Applications.

b). Web applets.

Standard – alone Applications : are programs written in Java to carry out certain tasks on a stand-alone local computer. Executing a stand-alone java program involves two steps.

Compiling source code into byte code using javac compiler.

Executing the byte code program using java interpreter.

Applets :

Applets are small Java programs developed for Internet Applications. An applet located in a server can be downloaded through Internet and executed on a client using a Java enabled browser Applets are embedded in an HTML document and run inside a web page.

Java Programming Structure :

A Java program may contain many classes of which only one class defines a main method. Classes contain data members and methods that operate on the data members of the class. To write Java program may contain one or more sections as shown below.

Documentation Section.
Package Statement.
Import Statements.
Interface Statements.
Class definitions
Main method class.
{
Main method definition
}

Documentation Section : This section contains set of comment lines giving the name for the program, the author name, software and hardware requirements.

Package statements : The first statement allowed in Java program is a package statement. This statement declares a package name and informs the compiler that the classes defined here belong to this package.

Import Statement : This is similar to the include statement in C. we can decline number of import statements in the program.

Interface Statements : An interface is like a class but includes a group of method declarations. This is also optional section.

Class Definition : Java program may have multiple system definitions. Classes are the primary and essential element of a java program. These classes are used to map the object to real world problems.

Main method class : Every java stand-alone program requires a main method as its starting point; this class is the essential part of a java program. The main method creates objects of various classes and establishes communication between them. On reaching the end of main, the program terminates and the control passes back to the operating system.

Java simple Program : In java every stand – alone application must have a main method inside a class. If main () is not written in java program JVM will not execute it. Main () is starting point for start execution of a java program. Generally method performs mainly 2 actions, they are :

It can accept some data from outside.
It can also generate some result.

```
Ex :class sample
        {
        public static void main (String args[])
        {
        System.out.println("GJC CHIRALA");
        }
            }
```

The first line class sample declares a class which is an object – oriented construct. Java is a true object – oriented language and therefore, everything must be placed inside a class.

Class is keyword and Sample is a java identifier that specifics a new class named **sample** was defined.
Every class definition in java begins with an opening brace "{" and end with a closing brace "}", appearing in the last line.
The third line public static void main (string args []) defines a method named main. Every java application program must include the main () method. This is the starting point for the interpreter to begin the executing of the program. The keyword **Public** indicates that the main () method can be accessed by the outside environment (JVM) and doesn't return any value. So void should write before that method's name, main is also a method. So we have to call that method with the help of an object. So we have to create object to calling the main method. But how is it possible to create an object before calling and belongs to the entire class.
Static : These methods are the methods, which can be called and executed without creating the objects. So we should declare the main () method as **static** one.
String args[] : Which is an array to store the values passed to main() method. These values passed to main () are called arguments. These values are stored in args[] array in the form of strings.

The only executable statement in the program is :

system.out.println("GJC, Chirala");

Implementing a Java Program :

Implementation of a java program involves a series of steps are :

create the program
compiling the program
Running the program

Creating the program :

We can create a program using any Text editor. The notepad is very convenient and small text editor. Enter the following program. Class sample

```
{
public static void main (String args[])
{
System.out.println("Vocational Courses");
}
}
```

We must save this program is a file as "Sample.java". This file is called the source file. All java source files will have the extension ."java". The file name must be the class name of the containing the main method.

Compiling the program : To compile the program, we must run the java compiler "javac" with the name of the source file on the command line, like **javac sample.java.** If everything is OK, the javac compiler creates a file called sample. Class containing the byte code of the program compiler automatically names the byte code file as <source filename>. Class

Running the program :

We need to use the java interpreter to run a stand – alone program. At the command prompt type: Java sample

Now, interpreter looks for the main method in the program and begins execution from there. Note that we simply type "test" at the command line and not "sample class" or "sample.java".

Machine neutral:

The compiler converts the source code files into byte code files. Those codes are machine – independent and therefore can be run on any machine. Java interpreter reads the byte code files and translates them into machine code for the specific machine on which the java program is running. The interpret is therefore specially written for each type of machine.

Installation of JDK :

To install the JDK software and set JAVA – HOME on a windows system.

Install the JDK software a) Goto

http ://java.sun.com/javase/downloads/index.jsp: b) Select the appropriate JDK software and click download. The JDK software is installed on your computer for example at

\programfiles\files\ java\jdk 1.6.0-02. You can move the JDK software to another location if desired.

To set Java-Home :

Right click my computer and select properties.

On the advanced tab, select environment variables and then edit java home to point to where the JDK software is located, for example.

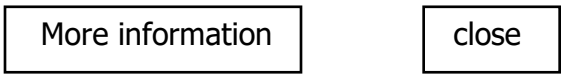
C: \program files\java\jdk1.6. -0.2.

To install java, we need to perform the following steps.

Double click the .exe file to initiate the installation procedure. The screen appears as Java SE Development kit 7 update 9(64bit)

Please wait while windows configures. Java SE development kit 7 update 9(64-bit) Gathering required information.

In the welcome screen, if we click on NEXT, starts the installation of Java. Once the installation is completed, the complete screen arrears as shown.



The screen describes the successful installation of java on the computer system.

Configuring Java :

Once the Java is installed, we need to configure it by adding the java path to the environment variable. PATH. This eliminates the restriction to use java only from its current working directory. To set the PATH variable to the java directory, we need to perform the following steps.

Right click on the "My Computer" or "Computer" icon

Open

Manage

Map network drive

Disconnect network drive

Create shortcut

Delete

Rename

Properties

Select the properties option from the drop down means the system properties dialogue box appears on screen select the advanced system settings option to display the advanced page displayed on screen.
Click the "Environment Variables" button to display the Environment variables dialogue box shown on screen.
The environment variable dialog box is divided into two sessions.

User variables
System variables

Under system variable sections, select the path option below the variable column and click the edit button. The edit system variable dialogue box appears shown on screen. By default, the path variable is already set to multiple locations.
To set java director path variable, append to directory path in the variable value text box separated by a semicolon (;) shown on the screen.
Click OK to close the Edit System
Variable dialogue box
Click Ok the close Environment variable dialogue box.
Click OK the close system properties dialogue box and complete the process of configuring java.

Short answer type questions

1. **Define OOP.**
2. **Who invented Java.**
3. **Write the Need of object oriented program.**
4. **What is a class?**
5. **What is an object?**
6. **What is data abstraction?**
7. **Expand the terms JVM, JRE, JIT, OOP.**
8. **What is JVM?**
9. **What is JRE?**
10. **What is JIT?**
11. **What is Byte code?**
12. **What is Machine neutral or Architectural Neutral?**

Long Answer Type Questions.

1. **Explain OOPS Concept.**
2. **Explain the features of Java.**
3. **Explain the java program, structure.**
4. **Explain implementation a java program.**
5. **Explain the web browsers used to Java.**
6. **How to install JDK software and set java home on window system.**

UNIT 2

THE JAVA PROGRAMMING LANGUAGE

2.1.0.Simple Java Programs.
2.1.1.Implementing a Java Programs.
2.1.2.Input/output Files in Java.
2.1.2.1.Streams
2.1.2.2.- Accepting a Single Character from Keyboard.
Accepting a string from key board.
Accepting integer values from key board.
How to print accepting input from key board by using Buffer.
Print to information as the screen.
Display formatting output with string format.
Variable
Declaration of Variable
Initialization of variable in Java.
Types variables.
Constants
Types of Constants
Java Laterals.
Types of Laterals.
Identifiers.
Separators in Java.
2.2.Data Types.
Primitive data Types
Mixed mode.
Casting Values.
Operation in Java.
Class, Objects and creating objects.
Defining a class.
Fields declaration.
Instance variable.
Methods declarations.
Creating objects.
Accessing class methods.

Creating Method
Method calling
Void Key Word
Method overloading
Over loading vrs. Method overriding.
Passing parameter by values

Static Members
This
Abstract Class Methods

Simple Java Program :

In Java, every stand – alone application must have a main () method inside a class. If main() method is not written in java program JVM will not execute it main() **is the starting point for start execution of a Java Program**. Generally a method performs mainly 2 actions. They are

It can accept some data from outside.
It can also generate some result.

Ex :

```
Class sample
        {
          Public static void main (String args[])
        {
         System.out.println("Government College");
        }
```

The first line **class sample** declares a class which is an object – oriented construct. Java is true object-oriented language and therefore, everything must be placed a class. Class is a key board and the **sample** is a java identifier that specifies a new class named **sample** was defined.

Every class definition in java begins with an opening brace "{" and ends with matching closing brace"}", appearing in the last line.

Public static void main (string args[])

It defines a method named main(). Every java application program must include the main () method. This is the starting point for the interpreter to begin the execution of the program. The keyword public indicates that the main() method can be accessed by the outside environment (JVM), and doesn't return any value. So void should write before that method's name. But main() is also a method. So we have to call that method with the help of an object. So we have no create the object to calling that main method. But how it is it possible to create an object belongs to the entire class. Such method called static method and should be declared as static. Static methods are the methods, which can be called and executed without creating the objects. So we should declare the main() method as static one string args() which is an array to store the values passed to main() method. These values passed to main() are called arguments. These values are stored in args [] array in the form of string.

The only executable statement in the program is System.out.println ("Government College"); There is similar to the printf() statement in c.

Implementing a Java Program :

Implementation of a java program involves a series of steps. They include.

Creating the program
Compiling the program
Running program

Creating the Program :

We can create a program using text editor. Entered the following program.

```
Class sample
{
        Public static void main (String args[])
        {
        System.out.println("welcome to java world");
        }
}
```

We must save this program in a file as "sample.java". this file is called the source file. All java source files will have in extension "Java". The file name must be class name of class containing the main method.

Compiling the Program : To compile the program, we must run the java compiler "Javac" with the name of the source file on the command line, like.

Java.sample.java

If everything is OK, the Javac compiler creates a file called sample. Class containing the byte code file as <source filename>.class.

Running the program : We need to use the java interpreter to run a stand – alone program. At the command prompt, type **java – sample** Now, interpreter looks for the main method in the program and begins execution from there. Note that we simply type "Sample" at the command line and not "Sample. class" or "Sample.java".

Machine neutral : The compiler converts the source code files into byte code files. Those codes are machine – independent and therefore can be run on any machine. Java interpreter reads the byte code files and translates them into machine code for the specific machine on which the java program is running. The interpreter is therefore specially written for each type of machine.

Input / Output Files in Java.

Introduction and Concepts of Statements :

*Input and output (I/O) are the most commonly used operations in any language. Reading data from keyboard and writting data on the monitor.
Some times require reading data from files and writing data to files. *Java uses streams to handle data input and output.

Input streams act as **source data** of data.
Output streams at as **destination** of data.

The **Java.IO** package defines various classes and methods for reading and writing files and handling data streams.

: Stream : A flow of data is often referred to as a data stream. Streams facilitate transporting data from one place to another. A stream is an ordered sequence of bytes that has a **source** or destination.

A stream is an object that represents an I/O devices or an device an I/O resource using which we transfer data into out of java program.

We have different kinds of streams in Java:

*One classification is on the direction of data & another classification is based upon the contents under transfer.

Input stream is used to bring data into java program output stream is used to transfer data out of java program.

*Byte / Binary streams are used to transfer binary data.

*Character streams are used to transfer textual/character data.

For Example :

Ata input stream dis=new Data Input Stream (System.in); In the above statement, we are attaching the keyboard to data input stream object.

The key word is represented by system. In.

Now data Input Stream object can read data coming from the key board. Here is a class and in a field in system class. The system class has following three fields.

System.in : It represents input stream object. This object represents the standard input device that is key board by default.

System.out : It represents print stream object. This object by default represents the standard output device that is monitor.

System.err : It represents print screen object. The object by default represents the standard output device that is monitor sometimes we can use system.err to print on the monitor.

Advantages of stream: Java program never performs the I/O operation directly on the I/O devices or I/O resources. Instead of it makes use of their representative i.e streams.

Object oriented benefit.

Device Independent programming is possible.

Java program need not know the functionally by the I/O device.

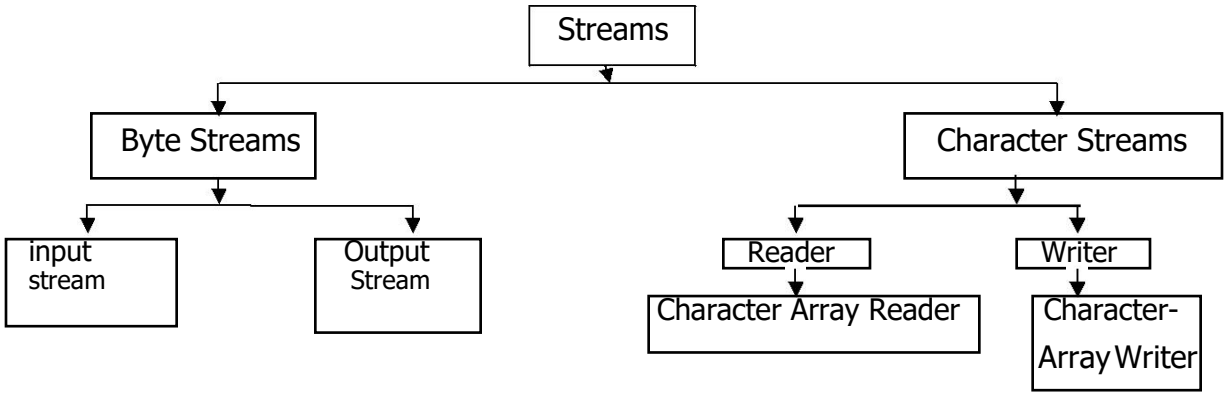
Stream Class: The **Java.IO** package defines a collection of stream classes that support input and output. To use these classes, or program needs to import the java. IO packages, as shown below.

```
Import java.IO.*;
```

Generally streams are classified into two types known as character streams and byte streams.

The Java.IO packages provides two set of classes hierarchies to handle character and byte streams for reading and writing.

Input stream and output stream classes are operated on bytes for reading and writing respectively.
Reader and **writer** classes are operated on character for reading and writting.
There are two other classes that are useful for handling input and output.
These are file class and **Random Access Files** class.



Accepting a single character from the key board:
Create a Buffered Reader class object (br):
Then read a single character from the keyboard using read () method
as char ch = br.Read();
The read () method reads a character from the keyboard but it returns its ASCII number. Which is integer, since integer number can't be stored in character type variable **ch** we should convert into char type by writing (char) before method. So

Char ch = (char) br. Read ();

But in java read() could not accept a character due to some reason like insufficient memory or illegal character then it gives to a runtime error which is called by the name IO Exception where IO standards from input/output and exemption represents runtime error.
So it's our duty to handle that exception, which is nothing but exception handling. But in this stage we don\t have that much of idea on the exception handling. But now to handle this type of exceptions. We just write "throughs "**IOException"** at the side of the method where read () is used.

Accepting a string from keyboard :
To read a string from the key board, read line() method is used by the Buffer Reader object, its also raise an exception like read() method. So what was step we followed to handle read () exception. Here also we are following Ex:

Ex : String Str = br.readline() ;

We have to follow some sequence of steps :
We should accept the integer number from the keyboard as a string using read line

() as String str = str = br. Read line ();

Now the number in str, i.e. in the form of string. Now this should be converted into int by using parseInt () method of **Integer class** as

Int X = Integer. Parse Int (Str);

We can write top two commands in to a single one.
That is int x = integer. Parse in (br.read line().

2.1.7. How to print Accepting input from the key board by using Buffer: A stream is required to accept input from the key board stream represents flow of data from one place to another place. Stream carries data from keyboard to memory. Memory to printer or memory to a file. So stream is mandatory to move the data from one place to another.

In java 2 types of statements. Input stream and output stream input stream is used to read the data out stream is used to write the data. All streams are represented by classes in java. i.o. package.

a. **System.in :** This represents Input stream object. Which is default representation standard input device i.e. key board.
b. **System.Out** :This represents print stream objects which by default represents print stream objects which by default represents standard output device i.e. monitor.
c. **System.err :** The represents print stream object which by default represents standard output device i.e. monitor keyboard is represented by a field called

Syntax : Input stream Reader ob = new Input Stream Reader (System.in). connect input stream reader to buffer reader, which is another input data properly, coming to the stream.
Syntax : Buffered Reader br = new Buffered Reader (ob).
We can get 'br' object in a single steps ob.
Syntax : Buffer Reader .br = new buffered (new input stream Reader (System.out);
Now we can read the data coming from the keyboard using read() and readline() method available in **buffer reader class**.

Print the information on the screen:

In java the print () method is used to display something on the monitor. A method should be called by using object name. so, we should create an object to the class to which print() method belongs. The print() method is belongs to **print stream** class. So we should call print() method by using an object to **print stream** class as

Print stream object.print ("welcome to java");

But it is not possible to create to object to **print stream** class as print stream class directly, an alternative is given to us i.e. System.out. Here **System** is the class name and out is a field in **system** class, when we call this field a

print stream calss object will be created internally. So, finally we can call the print() method as :

```
Class sample
{
Public static void main (String args[])
{
System.out.print("Govt. Junior College for Girls");
}
}
```

\> Javac.sample.java

C:\> Java sample

Output : Govt. Junior College for Girls.

Using Java C command the Java compiler generates the byte code instruction (.class file). This file is executed by JVM. Here JVM is an executed program.

That observing the command "Java Sample". Here JVM is nothing but java.exe program. In fact, JVM is written in **C Language. Formatting output :**

```
Int a = 20, b = 30
    C = a + b;
```

System.out.print ("Sum of the two numbers =" +c);
Hence we are using + to java the string "Sum of the numbers =" and the numeric value

The reason is that print() method cannot display more than one value. We should combine them using "+" operator.

Let int x = 20, y = 30
Now we want the **result is** : sum of the two numbers 50.

If you are specifying format.
System.out.print("sum of the two numbers =", x + y);
Then **result is** : sum of the two numbers = 20 30.

Since left one is string and the next value is also converted as string (x is also) and then next value y also converted as string. This is nothing but string concatenation. To avoid that problem, we are following like this.

System.out.print ("sum of the two numbers =";(x+ y)); **Result as** sum of the two numbers = 50
There is another method System. out. println (string) is also available. It displays the arguments string on the monitor and return to the next line automatically.

Display formatted Output with "String.format" :

Hence we have take the help of format () method of the String class. The format character supported by system. Out. Print() are also usable with format () method. This method return formatted string.

String str = string.format ("name = %s %n Num = %d %n avg = %f", st, n, f); System .out. println (str);

VARIABLES:

A variable is a data name that may be used to store a data value. During execution of a program the values are change is called **variable**. The variable may take different values at different times during execution. For instance, we used the variable Total as Tot is 40 to store the value of total marks. A variable name can be chosen by the programmer in a meaningful way so as to reflect its function or nature in the program.

Example : Avg (Average)
Ht (height)
Tot((Total)
Tot _ stu(Total students)

If you want write to compound words as variable name. You can put the underscore (_) character only.
The length of variable not exceed 31 characters.
The variable name should not be a keyword.
White spaces are not allowed.

Invalid variable names :

Example: 521, 4abc, / ram, 26^{th} etc.

Table – Invalid and valued Examples of variables names

Variable name	Valid / not valid	Remarks
First name	Valid	
Int	Not valid	Int is a key word
Rs. Rate	Not valid	Rs sign is illegal
Average marks	Not valid	Blank space is notallowed
Type def	Not valid	It is a reserve word.
No	Valued	-
Na	Valued	-
Name	Valued	-
Number	Valued	-
My_name	Valued	-

Fig 2.3.3

After assigning variable names we must declare to the compiler. The variable does two things.

(i) It specifies the compiler what the variable name is (ii) It specifies what type of data the variable will hold.

The declaration of variables must be done before. They are used in program.

Variable declaration :

A variable can be used to store a value of any data type. This is the name has nothing to do with its types.

Syntax : Data type var1, var2, var3…….var n;

Var1, var2, var3……..var n are the name of variable. Variables are separated by commas. Variable declaration statement must end with a semicolon(;).

Examples: int a;
Int no; marks
Float avg – marks;
Char name[20];
Double ratio;

In the above example int, float, char, double are data types, a, no, avg marks are numbers. Name or ratio are variable names with character data value and double data value respectively.

Assign the values to variable:

Variable are declared for use in programming statements such as Interest = P*N*R/100;

Values can be assigned to variable using the assignment operator "=" as follows.

Variable – name = constant or variable or expression

An assignment implies that the value of the variable on the left of the equal sign ("=")right side of equal sign"=".

2.1.4. Constant : A constant are fixed values those are not changed during the execution of program Java supports several types of constants those are :

- Integer Constants
- Real Constants
- Single character constants
- Backslash character constants

Constants increase the readability of code and in addition cannot be effected by program errors. C supports several types of constants.

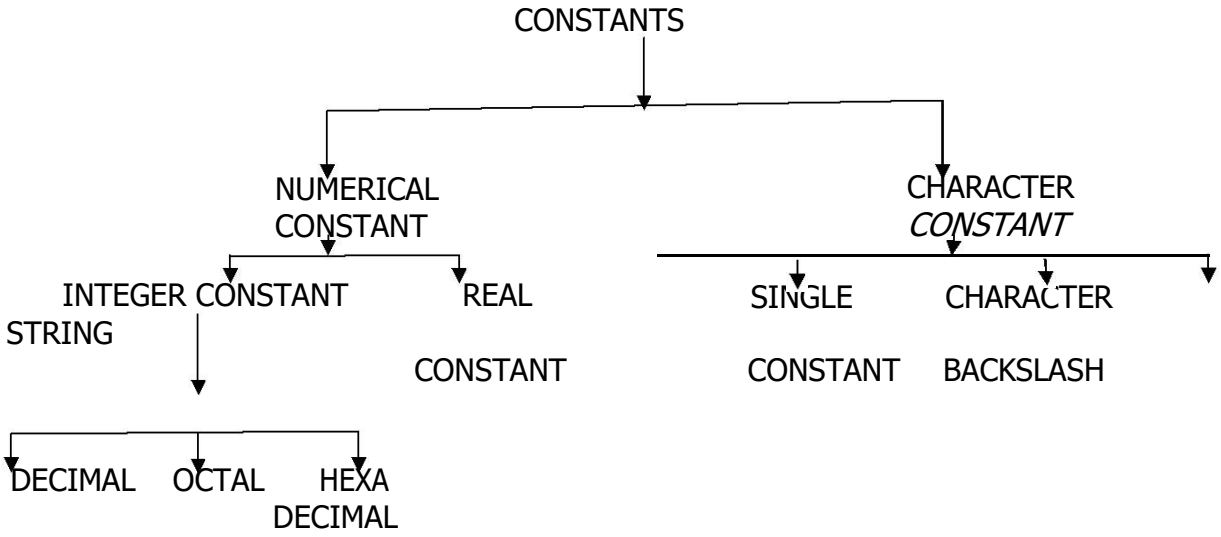


Fig : Constant types

(a) NUMERICAL CONSTANT :

It can be any of decimal integers (base 10) consists one or more digits 0-9

(b)INTEGER CONSTANTS :

An integer constants refers to a sequence of digits. These are three types, namely (i) decimal (ii) octal (iii) hexa decimal

Decimal integer : The decimal integer constants consists of a set of digits 0 (zero) to 9, preceded by an optional + or – sign

Example : 521, -521, 64537, +72

In the formation of integer constants do not maintain spaces between digits don't put an special character between digits like comma (,) Semicolon(;) and non_ digits characters are not permitted between digits.

Example to invalid integer constant:

23,265 (not valued) commas are not allowed between integer constant
265 (not valued) blank spaces are not allowed between integer constant Rs.
(not valued) special symbols are not allowed between integer constant

The above examples are illegal numbers.

Octal:

An octal integer constant consists of any combination of digits from 0 (zero) to 7 with a leading 0 (zero).

Example :0212
0393
0123
0

The + or – sign may have optional. In the absence of sign, the constant is assumed to be positive. Special characters like , (commas) and blank spaces are not permitted do not put the period (.) in between digits.

The in valued octal integers are
05.56decimal point is not permitted
07,89 commas are not permitted
X023 X is a special symbol is not permitted
5 is not a permissible symbol as first character.

Hexadecimal integer :

A sequence of digits preceded by OX. The letters A,B,C,D,E,F (lower case or upper case) represent A or a is equal to 10, B or b is equal to 11, C or c is equal to 12 and D or d is equal to 13, E or e equal to 14 an F or f equal to 15.

Example : 0X2 0X2
0X8C 0X8C
0Xabc 0Xabc
0X24 0XEF
0X24

We rarely use hexadecimal numbers in C programming. The following rules are required to form a hexadecimal number.

It is a sequence of one or more symbols. (0 to 9 and A to F) the symbols of Hexadecimal number system
It may have an optional + or – sign in the absence of sign, the constant is assumed be positive.
It should start with the symbols OX or ox
Commas and blanks spaces are not permitted.
It should not have a period (.) as a part of it.

Some examples for **invalid** hexadecimal integer constants.

OX.421(.) Dot is not permitted.
-0X512- sign is not permitted.
-056.A . the symbols in between the digits.

REAL CONSTANTS :

Integers, numbers containing fraction parts, some numbers are called real (or) floating point.

Example :

-2.321
+523.12
5.2351
0.0282

These numbers are shown in decimal notation having a whole number followed by a decimal point and the fractions part. It is possible to omit digits before the decimal point, digits after the decimal point that is

315 . 6
. 62
. 71
.5

The above are all valid real numbers. A real numbers may also be expressed in exponential notation. For example, the value 315.25 may be written 3.1525e2 in exponential notation e2 means multiply by 102. The general form is

Montissa e exponent:

The mantissa is either a real number expressed in decimal notation or an integer. The exponent is an integer number with an optional plus or minus sign. The letter 'e' separating the mantissa and exponent can be written in either lower_ case(e) or upper case(E).

(d)CHARACTER CONSTANTS :

(a) A single character constants :

A single character constant contains a single character enclosed with in pair of single quotation mark.

Example :'A', '5', ',^*'
Note that the character constants 5 is not same as integer value of '5'. '5' is the value ASCII is a 'blank space'.

(b) String constant :

A string constant is a sequence of character enclosed in double quotes ("").
The characters may be letters, numbers, special characters and blank space..
Examples :"GJC"
"college"

"5+6"
"xyz"
"A"

In the above examples "A" is not equal to 'A'. "A" is a double string constants, 'A' is a single string constant. Further a single string constant does not have an equivalent integer while a character constant has integer value. i.e. (ASCII value), character strings are often used in programs to build meaningful programs.

(c). BACK SLASH CHARACTERS :

'C' supports some special characters constants that are used in output function. For example, the symbol '\n' stands for new line character. Some backslash character constants are shows in Fig 2.2.3.0.

TABLE

Back slash Constant	Meaning
"\a"	Bell
"\b"	Back space
"\f"	Form feed
"\n"	New line
"\r"	Carriage return
"\t"	Horizontal tab
"\v"	Vertical tab
"\0"	Null

2.1.4.Java Literals : Literals in Java are a sequence of characters (digits, letters and other characters) that represent constant values to be stored in variables. There are five types of literals in java.

Integer literals
Float literals
Character Literals
String Literals

Integer Literals : For internal data types byte, short int long we can specify literals.
We can specify literals in 4 ways :

Decimal literals (base 10) : In this digits are allowed (0-9)

Octal Literals (Base 8) in this form the allowed digits are (0-7). The octal number should be prefix with 0.

Hexa – decimal literals (Base – 16) In this form the allowed digits are 0-9 and characters are A to F. We can use both upper and lower case characters. We know that java is a case – sensitive programming language.

Example : OX or ox.

Int x = 0x125 Face.

Binary Literals : From 0 or 1 values should be prefixed with **ob** or **OB**.

Int x = ob 1011.

Float Literals :

For floating point data types we can specify literals in only decimal form and we specify in octal and Hexa decimal forms.

Character Literals : For char data type we can specify literal char and data type as single character, with in single quote.

Ex : Char ch = 'a'

Escape Sequence : Every escape character can be specify as char literals.

Char ch = '\n';

Sting Literals : Any sequence of character within double quotes is treated as string literals.

String s = "College";

String Literals may not contain un escaped New line or linefeed characters.

Boolean Literals : Only two values are allowed for Boolean Literals i.e. True or false

Ex : Boolean b = True.

Mixed Mode : When we are performing concatenation operations, then the values in brackets are concatenated first. Then the values are concatenated from the left to the right. We should be careful when we are mixing character literals and integer in string concatenation operations and this type of operations are known as **mixed mode** operations.

Identifiers : These are programmer –designed tokens. They are used for naming classes, methods, variables, objects, labels, packages and interfaces in a program. Java identifiers follow some rules.

a. They can have alphabet, digits and to underscore.
b. They must not begin with digits.
c. Uppercase and lower case letters are distinct.
d. They can be of any length.

Java developers have following some name convention**:**

a. Names of all public methods and instance variables starts with a leading lower case letters.
b. When more than one word are used in a name, the second and subsequent words are marked with a leading upper case letters.
    i. Ex: print stream
c. All private and local variables are used only one lower case letters combined with underscore. All classes and interfaces start with leading uppercase.
d. Variables that represent constant values are all uppercase letters and underscore between words.

Separator in Java :

Separator help define the structure of a program. The **separators** used in java are

Parenthesis ( )
Braces { }
•Period.
Semicolon ;
Comma ,
Brackets [ ]

Separator help define the structure of a program. The separators used in java are

Parenthesis ( ); Braces { }; Period(.), Semicolon (;), Comma (,), and Brackets [ ]

Parenthesis ( ) : This holds the list of parameters in method definition also used in control statements & type casting.

Braces { } This is used to identify the block of code or statements, class methods.

Period . It is used to separate the package name from sub – package name & class name. it is also used to separate variables or method from its object or instance.

Semicolon ; It is terminates statements the statement in java.

Comma , t is used separate the consecutive identifiers in a variable declaration.

Brackets [ ] It is used in Array declaration.

Data Types :

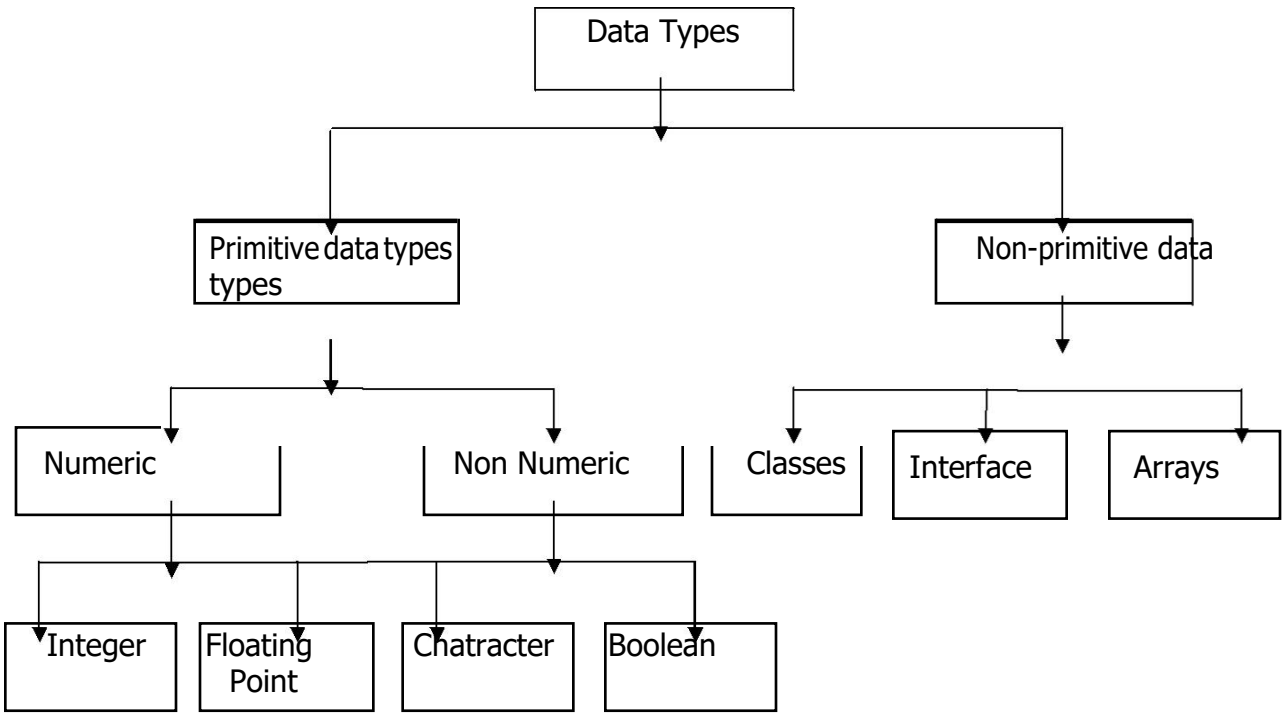
They are two types of data types in java

Primitive data types

Non-primitive data type.

Primitive data types : The primitive data types includes **Boolean, char byte int, short in long, float and double.**

Non Primitive data types : These are classes interfaces and Arrays.



Integer Types : Integer types can hold whole number such as 512, 151,1234. The size of the values that can be stored depends upon the integer data type we choose java

supports 4 types of integer **Byte, Short, int long** . Java does not support the concept of unsigned types. So all java values are signed they can be positive or negative.

Data Type	Memory Size	Min and Max Values
Byte	1 byte	-128 to + 128
Short	2 Byte	- 32768 to + 732.767
Int	4 Bytes	-2147483648 – 214783648
Long	8 Bytes	- 9223372036854775808 to + 9223372036854775808

Floating Point : The float data types is used to hold numbers containing such as 124.56 and -123.45. These are two types

Float type values are single – precision number.

Double Types represents double – precision number.

Floating point numbers are treated as double precision quantities. To force then to be in single precision mode. We must append **f** or **F** to the numbers.

Ex : 1.23 f, 12345.678 .F. etc.

Data Type	**Memory Size**	**Min and Max Values**
Float Double	4 Bytes 8 Bytes	- 3.4 e 38 to -1.4 e – 45 –ve no's 1.4 e 45 to 3.4 e 38 + ve no's

Character Types :

These are used to store character constants in memory. Java provides a character data type called char. Char assumes a size of 2 bytes, but basically, it can hold only single character.

Data Type	**Memory size**	**Min and Max values**
Char	2 bytes	0 – 65535

Boolean type :

It is used when we want to test a particular condition. There are only two values that a Boolean type can be take **True** or **False**.

It is denoted by keyword Boolean and values only one bit of storage.

Casting Values :

The process of converting one data type value into another data type is called casting. It is done is two ways.

Implicit type casting or Automatic conversion.
Explicit Type casting.

Automatic Conversion : We know that the computer, consider one operator at a time involving two operands. If the operands one of different types. The "lower" types is automatically converted to the higher type before the operation processed. The result is higher type this process is called implicit type casting or automatic conversion.

Ex 1 Byte a = 62, int b = a ;
Ex 2 Double avg = (a + b)/ 2.0

Explicit type casting :

When we want to force a type conversion is called explicit type casting. Syntax : type variable 1 = (type) variable 2;

Ex 1: int m = 52;
Byte n = (byte) m;
Ex 2: Byte a = 12345, b = 25000
Int c = (int) a + b;

The four integer types can be cost to any type except Boolean. Similarly, the float and double can be cost to any other Boolean; casting to similar type can result in a class of data.

Operators Java Material :

A. Java provides a rich set of operators to manipulate variables we can devide all the java operators in to the following groups.

a. ARITHMETICAL OPERATORS.
b. RELATIONAL OPERATORS.
c. LOGICAL OPERATORS.
d. BIT WISE OPERATORS.
e. ASIGNMENT OPERATORS.
f. INCREMENT AND DECREMENT OPERATORS.
g. CONDITIONAL OPRATORS.

A. ARITHMATICAL OPERATORS:- Arithematical operators are used in mathematical expressions in the same way that they are used algebra. The following table its arithematical operators .

OPERATORS	purpose
+	addition
_	Subtraction
/*	multiplication
\	division
%	remainder

ARITHMETICAL:

EX:- Suppose a andb are variables and assignment a=5 and b=3 then.

OPERATOR	A.M	RESULT
+	a+b	8
_	a_b	2
/	a/b	1
%	a%b	2
*	a*b	15

REAL ARITHMETIC: Un like , c java allows the modulas to the float values after integer division flere quotient must be integer value .

EX:- Suppose x and y are float variable and assigned x=20.5 and y=10.5 then .

OPERATOR	Arithematic Expression	RESULT
+	a+b	31.0
_	a_b	10.0
/	a/b	215.25
%	a%b	1.9523
*	a * b	10.0

MIXED MODE:-When one of operands is real and the other is integer then the expression is called operands must be converted as integer operands. Then the result will be in integer mode.

EX:- int 12.0/5 gives result 2. Here 12.0. is float operand 5 is integer operand.

B.RELATIONAL OPERATORS:- There are six following relational operation supported by java language.

OPERATORS	PERPOSE
=	It is equal to
!=	It is not equal to
<	It is less than
>	Greater than
>=	Greater than equal to
<=	Less than equal to

EX:- Suppose a, b and c are integer variables and assigned 3, 5 and 10 respectively.

RELATIONAL EXPRESSION	RESULT
a > b	false
(a+b)<c	true
a<=3	true
a!=b	true
b<=9	false
c==10	true

C.LOGICAL OPERATORS:- The following table lists logical operators.

OPERATOR	PURPOSE
&&	AND
ll	OR
!	NOT

a.&&(AND):- It both operators are true then the logical is true.

A	B	A & & B
0	0	0
0	0	0
1	0	0
1	1	1

b.l l(OR):- Either operands is true then the logical or is true.

A	B	A l l B

0	0	0
0	1	1
1	0	1
1	1	1

!(NOT):-If negative the operand.

A	! A
0	1
1	0

EX:-Suppose a, b and c are integer variable assignment 3, 5 and 10 co respectively.

Expression	RESULT
(a>b)&&(c>a)	false
(a>b)ll(c<a)	true
!(a>b)	true
(b>a)&&(b,==c)	true.

D. BIT WISE AND BITSHIFT OPERATORS:- Bitwise and bit sift operators in java

are used to manipulate the content of variable at bit level according to binary format. These format operators perform bitwise and bit shift operator.

OPERATIONS ON INTEGRAL TYPE VARIABLES:-

SYMBOL	NAME OF THE OPERATORS	EXAMPLE
~ & ! ^ << >> >>>	UNARY BITWISE COMPLEMENT BITWISE AND operand2 BITWISE INCLUSIVE OR BITWISE EXECLUSIVE OR SIGNED LEFT SHIFT operand2 SIGNED RIGHT SHIFT UNSIGNED RIGHT SHIFT	~ (0 becomes 1 1 becomes 0) operand1 & operand1 ! operand2 operand1 ^ operand2 operand1 << operand1 >> operand2 operand1 >>> operand2

Lets understand and these operators in brief.

UNARY BITWISE COMPLEMENT("N"):-The unary bitwise compliment operator takes a single bit and inverts in it this case , the value of a bit which is 0.Become 1 and 1 become 0.

1 01

0 11

1 10

BIT SHIFT OPERATORS:- The computer processor has the register including a fixed number of available bits for storing numaricals. So it is possible to "shift out " some bits of the register at one end, and "shift in " from the other end the other number of bits are shifted within the range in mode of 32.

SIGN BIT:-A sign bit found in the left most position of the number and is known as most significant bit (MSB) which indicates the status of a number i.e., the number is positive or negative, is the value of the sign bit is "0" then the number is positive otherwise the number is negative.
Now, let's understand those operators in brief.

SIGNED LEFT SHIFT("<<"):-The signed left shift("<<") operators shifts a bit or bits to the left by the distance specified in the right operand. In this case , the left most digit is shifted at the end of the register and a new "0" is shifted into. The right most position no matter the number is negative or positive. In both of case the leading bit position is always filled with is 0.

MSB	4	3	2	1	0	bit position
	0	0	1	1	1	values
	0	1	1	1	0	

SIGNED RIGHT SHIFT:- The signed right shift("<<") operators shifted a bit or bits to the right by the distance specified in the right operand and files the left most bit by the sign bit . In this case the right most bit or bits is shifted out, and a new 0 is filled with the sign bit into the high-order bits to the left position if the left operand is positive; otherwise 1, if the left operand is negative. This technique is known as sign extension.

MSB LSB

4	3	2	1	0
0	0	1	1	1
0	0	0	1	1

UNSIGNED RIGHT SHIFT:(">>>"):-The unsigned right shift (">>>")operator behave like the signed light shift operator. I.e., it shift a bit or bits to the right. But unlike">>" operator, this operators always shift 0 's into the left most position by the distance specified in the right operand. So the result of applying the >> > operator is always positive.

FOR example , the expression "14 >>> 2"; shifts all bits of the number 14 to the right placing a 0 to the left for each blank place thus the value 1110 becomes 0011 or 3 in decimal.

E. THE ASIGNMENT OPERATORS :- There are following assignment supported by java language.

OPERATOR	EXPRESSION	EQUIVALENT
+=	X+=a	X=X+a
_=	X-=a	X=X-a
*=	X*=a	X=X*a
%=	X%=a	X=X%a
/=	X/=a	X=X/a
<=	X<<=a	X=X<<2
>=	X>>=2	X=X>>2
&=	X&=2	X=&2
^=	X^=2	X=X^2
|=	X|=2	X=X|2

F. INCREMENT AND DECREMENT OPERATORS :- They are two increment and decrement operators ++ and - - . These two operators are unique in that can be written both before the operand there are applied to called prefix increment and decrement (+ + i / - - i) or often called prefix increment, decrement(i + + or i - -) the meaning is different in each case.

G.CONDITIONAL OPERATORS :-Conditional operators is also known as that ternary operators . This operator consists of three operands and is used to evaluate

Boolean expressions the goal of the operators is decide which value should be assigned is the variable.

SYNTAX:- variable x =(conditional expression) ? expression1: expression2

Conditional expression is evaluated first and if it is true expression1 is assigned to the variable x.

If it is false expression2 to is assigned to the variable x.

Class, objects and creating objects :

Java is a true object oriented language and therefore everything is in the form of the classes. Class defines the state and behaviors of the basic program component is

called objects classes creates objects use methods to communicate between them. Class provides a convenient way for packing the logical related data items are called fields and the functions called methods that work on them calling a specific a method in an object is described as sending the object a message. Java provides basic OOP concepts such as encapsulation, inheritance and polymorphism.

Defining a class : A class is a user – defined data type with a Template that service to define its properties. Once the class type has defined, we can create "variable" of that type using declarations that are similar to the basic. In java these variables are called as **instance of classes**. Which are actual objects.

Syntax ClassClass Name [Extends Super class name]

{
Declarations of members
Method declarations
}

Here, class name super class name are any valid java identifiers. The key word **extends** indicates that the properties of the super class name class are extend to the class name class. This is called inheritance.

Field declaration : Data is encapsulated in a class by placing fields inside the body of the class definition. These variables are called instance variables because they are created whenever an object of the class in initiated. **Ex** : class Rectangle

{
Int length;
Int width;
}

The class Rectangle contains two integer type instance variable. It is allowed to declare them in one line as int length, width. These variables are also known as member variables.

Instance variable : It is used in java by object to store their states. Variables which are defined with the **static** key word and are outside any method declaration are object specific and are known as instance variable. There are called so because their values are **instance** specific and are not shared among instances.

```
Class sample
{
Public string studentname;
Private int studentnumber;
}
Package com.jbt;
Public class variables In java
{
Int instance Field
    Static string static Field;
    public void method()
    {
    Final string local variable = "Initial value";
    System.out.println(local variable);
```

```
}
Public static void main (string args[])
{
Variablesinjava obj = new variablesInjava();
System.out.println(obj.instance Field);
System.out.println(obj.staticField);
System.out.println(variablesInjava.static Field);
System.out.println(newvariablesInjave().instanceField);
```

Rules for instance vaiable:

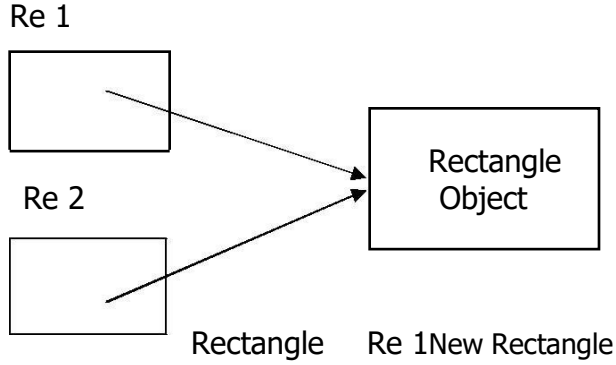
Instance variables can use any of the four access levels.
They can be marked final.
They can be marked transient.
They cannot be marked abstract.
They cannot be marked native.
They cannot be marked static.
They cannot be marked synchronized

Creating Objects : In java object in a block of memory that contain space to store all the instance variables, creating the object is also referred to as initiating an object. Objects are created using the new operator. The new operator creates an object of the specified class and returns.

Example : Rectangle Rect
Rect = new rectangle() ;
Now the variable rect is an object of the rectangle class.

If both statements can be combined into one statement. As 1
Rectangle rect = new rectangle ();
In this way we can create **N** numbers of objects to the Rectangle class.

Each object has its own copy of the instance variable of its class. This means that any changes to the variables of one object have no effect on the variables of another object. It's also possible to create more than one reference to a single object in the program.



Rectangle Re 1New Rectangle ();
Rectangle Re 2= Re1 ;

Methods Declaration :

A class contains fields as well as methods. The methods are necessary for manipulating the data contained in the class. Methods are declared inside the body of the class but immediately after the declaration of the instances variables.

Syntax : type methodname (parameter list)

```
{
Method body ;
}
```

The type specifies the type of value the methods would return. If the method doesn't return any value then we have to put **void** keyword the method name is valid identifier. The parameter list is always enclosed in parameters and these are separated with commas.

Example : class rectangle

```
{
Int length;
Int width;
Void get data (int x, int y)
{
Length = x;
Width = y;
}
}
```

Access class Methods: Object variables and methods are not possible to access from outside of the class. It means we cannot access the properties of the object directly. To do this, we must use concerned object and **dot operator** use. Example :

Object name. variable name = values;

Object name. method name (parameter list);

In the above example object_name is the name of object, variable _ name is the name of the instance variable inside the object that we wish to access. Method name is the name of method that we wish to access, and parameter list is a name the list of "actived values" that must match in type and number with the parameter list of the method name declared in the class.

Example :

```
Rectangle re1 = new rectangle();
Rectangle re2 = new rectangle ();
Re1. Length = 20;
Re1. Width = 30;
Re2. Length = 25;
Re2.width = 40;
```

No two objects re1 and re2 have different values in their fields.

	Re 1	Re2	
Re1.length	20	25	Re2.length
Re1.width	30	40	Re2.width

The above example is the one way of assigning values to instance variables is the object. Another way is more convenient way of assigning values to instance variables is to use method that is declared to class.

The method get data() can be used to do this work. We can call the get data() method on any Rectangle object to set the values of both length and width. This is a method on any rectangle object to set the values of both length and width. That is :

```
Rectangle re1 = new Rectangle();
Re1.getdata (20,30)
```

This code creates re1 objects and passes in the 20,30 for x and y parameters of the method get data(). This method assigns these values to length and width variables respectively void get data (int x, int y)

```
{
        Length = x;
        Width = y;
}
```

Now the object. Re1 contains values for its variables we can compute the area of the object.re1.

The first approach is to access the instant variable using the dot operator and compute to area.

Example int area = re1. Length * re1. Width;

The second approach is to access the instance variable using method rect area(); declared inside the class.

Example : Class Rectangle

```
        {
        Int length, width;
        Void get data (int x, int y);
        {
                Length = x;
                Width = y;
        }
    Void rect area ()
        {
                Int area = length * width;
        }
}
Class rect area
        {
                Public static void main (string args[])
        {
                Int area1, area2;
        Rectangle re1 = new rectangle[];
        Re 1.length =20
        Re 1. Width = 30
    Area1 = re1.length * re1.width;
    Re2.getdata (25,40);
```

```
Area2 = re2.rectarea();
System.out.println("area1=", area1);
System.ouot.println("area2="; area2);
        }
        }
```

Access Specifier :

An accesser specifier is a keyword that specifies has to access the members of a class or a class itself. We can use access specifiers before a class and its members. There are four access specifiers available in java.

Private
Public
Protected
Private Protected
Default

Private : Private members of a class are not accessible any where outside the class. They are accessibly only within the class by the methods of that class.

Public : Public members or a class are accessible every where outside the class. So any other program can read then and use them.

Protected : Protected members of a class are accessible all classes in the same package and derived classes of this class.

Private protected : This type of field visible in all subclasses regardless of what pakage they are in, remember. These fields are not accessible by other classes in the same pakage.

Default : If no access specifier is written by programmer, then the java compiler uses a default access specifier. Default members are accessible outside the class, but with same package.

Constructor : It's special method is used to initialige the instance variables at the time of object creation. It consists of same properties. They are :

Constructor name and class name should be same and its name ends with a pair of simple braces.

A constructor may have or may not have parameters. Parameters are used to receive the data from outside world. It the constructor doesn't have arguments them it is called as default constructor. If it has then that constructor is called as argument constructor.

Constructor are not automatically executed it is not possible to call with objects. Constructor is called and executed only once per object.

Constructor are not returning any value, not even void also. Example : Student S1 = new student ();
Student S2 = new student ("Murthy", "Lecturer");

Constructor vrs. Methods:

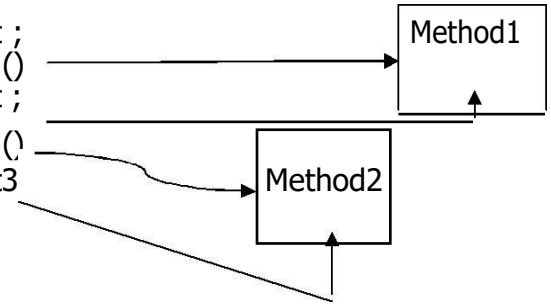
Constructor	Method
* It is used to intialise the instance Variable of a class	* It is used for any general purpose processings or calculations.
* It's name and class should be Equal	* Its name and class name cannot be same
It is called at the time of object Creation	* It can be called for creating the object
* Its called by once per object	* It can be called several times on the object
* It called and executed automatically	* It is execute when we call it

Methods :

What are the Methods?
A method is a set of code which is referred to by name and can be called at any point in a program simply by utilizing to methods name. Think of a method as a subprogram that acts on data and often returns a value. Each method has its own name. when that name is encountered in a program, the execution of the program branches to the body of that method. When the method is finished, execution return to the area of the program code from which it was called and the program continues on the next line of code.

Example :

```
Public static void main (string [] args)
{
        Statement ;
        Method 1 ()
        Statement ;
        Method 2 ()
        Statement3
```



You are using methods when you use system.out.println() or system.out.print() they are two basic types of methods.

2.7.1: How to define a method in java.

Classes and objects in java classes and objects an basic concepts of object oriented programming which revolve around to real life entities. A class is user defined blue print or prototype from which objects are created. It represents the set of properties or methods that are common to all objects or one.

They are two basic types of methods :

Built in : Built in methods are part of the compiler packages, such as system.out.println() and system. Exit (0);

User defined: these methods are created by program. These methods take on names that you assign to them and perform tasks that you create.

Method : Java method is a collection of statements that are grouped together to perform on operation. When you call the system.out.println() method.

For example. The system actually executes several statements in order to display a message on to console.

Now you will learn how to create your own methods with or without return values, invoke a method with without parameter, and apply method abstraction in the program design.

Creating Method : A java method is a collection of statements that are grouped together to perform an operation. When you call the system.out.println () method.

Example : Public static int methodname (int a, int b);

```
{
        Body of the method
}
```

Here

* Public static modifier
  Int return type
  Method name = Name of the method
  A, b = formal parameters
  Int a, int b = list of parameters

Method definition consists of a method headers and a method body. The same is shown in the following syntax :

Syntax :

```
Modifier return type name () method (parameter list())
{
  Body of the method
}
```

Modifier : It defines the access type of the method and it is optional to use.

Return type : Method may return a value.

Name of Method : This is the Method name. The method signature consists the method name and parameters list.

Parameters list : The list of parameter it is the type order and number of parameters of a method. These are optional, method may contain zero parameters.

Method body : The method body defines what the method does with the statements.

Example : Here is the source code of the above defined method called (). This methods takes two parameters Num1 and Num2 and returns the maximum between the two

```
Public static int minfuntion (int N1, int N2)
        {
            Int min;
             If (N1> N2)
            Min = N2
            Else
            Min = N1;
            Return min ;
        }
```

Method Calling : For using a method it should be called. There a two ways in which method is called i.e. method return a value or returning nothing.

The process of method calling in sample, when a program invokes a method, the program control get transferred to the called method. This called method then returns control to the caller in two conditions when –

The return statmement is executed.

It reaches the method ending closing brace.

The methods returning void is considered as call to a statements. Lets consider an example :

```
Int result = sum (0, 9);
```

The following is the example to demonstrate how to define a method and how to call it.

```
Public class min number
    {
    Public static void main (string[] args)
    {
    Int a = 10
    Int b = 12
    Int c = min Function (a,b);
    System.out.println("Minimum value =" +c);
}
Public static int minfunction(int N1, int N2)
    {
        Int min;
        If (N1> N2)
        Min = N2;
```

```
        Else
        Min = N1;
        Return min;
        }
}
```

Output : 10.

The void key word :

The void key word allows us to create methods which do not return a value. Here, is the following example we are considering a **void** method rank points. This method is a void method, which does not return any value call to void method must be a statement i.e. method Rankpoint (189.6); It is a java statememnt which ends with semicolon as show in the following example.

```
Public class. Example void
    {
        Public static void main (string [ ] args)
        {
            If points > 189.6)
        }
        Public static void method Rank Points ( "Rank:
        A"); Else
        If (points >= 150.5)
        {
        System.out.println("Rank.A2");
        Else
        System.out.println("Rank.A3");
        }
        }
        }
        Output : Rank A.
```

Passing parameters by value :

While working under calling process, an argument is to be passed. These should be is the same order as their respective parameters in the method specification parameters can be passed by value by reference. Passing parameters by value means calling a method with a parameter. Through this, the argument value is passed to the parameters.

Example : The following program shows an example of passing parameters by value. The value of the arguments remains the same even after the method invocation.

```
Public class swapping
Public static void main (String args[])
{
Int a = 20
Int b = 35
System.out.println("Before swapping a = "+a" and "b = "
+b); Swapping (a,b);
```

```
        System.out.println ("\n before and after swapping values will be same
        here"); System.out.println("After swapping a = "+a", and b = "+b");
Public static void swapping (int a, int b)
        {
        System.out.println("before swapping (inside), a=" +a "and b=",
                +b); Int c = a;
                        A = b;
                        B = c;
System.out.pintln("After swapping (inside), a = " +a or "b = " + b);
        }
        }
```

Output

Before swapping a = 20 and b = 35
Before swapping (inside) a = 30, b = 35
After swapping (inside) a = 35 b = 20
After swapping a = 20, b = 35

Method Overloading :

When a class has two or more methods by the same name but different parameters. It is known as method overloading. It is different from overloading in overriding, a method has the same method name, type, number of parameters etc. let's consider the example discussed earlier for finding minimum numbers of double type. Then the concept of overloading will be introduced to create two or more methods with the same name but different parameters.

Example :

```
        Public class example overloading
        {
        Public static void main (string [ ] args ())
        {
        Int a = 15 ;
        Int b = 12 ;
        Double c = 9.7;
        Double d = 14.2;
        Int result = min Function (a,b); Double
result 2 = min Function (c,d); System.out.println
("Minimum value = "+ result 1);
System.out.println ("Minimum value ="+ result2);
Public static int min Function (int N1, int N2)
        {
                Int min ;
                If (N1 > N2)
                Min = N2;
                Else
                Min = N1;
                Return min;
```

```
}

Public static double min Function (double N1, double N2)
    {
        Double min
        If (N1 > N2)
        Min = N2;
        Else
        Min = N1;
        Return min;
        }
        }
    Minimum value 15
    Minimum value 9.7
```

Overloading methods makes program readable. Here two methods are given by the same name but with different parameters. The minimum number from integer and double types in the result.

Static Members : Normally for every instance variable and methods in a class are instantiated for every creation of object a new copy of each of these is created. These are accessed using the objects.

When ever we want to define a member this is common to the objects and accessed without using a particular object. That is the member belongs to class a whole return then the objects created from the class. Such member can be defined as follows.

```
Static in I;
Static int max (int x, int y);
```

These are called static members. Since these members are associated with class itself rather than individual objects, the static variables and static variables and static methods are often referred as class variables and class methods.

Example :

```
Float x = math.sqrt (25.0)
```

Then method sqrt is a class method defined in math class. Class math operation.

```
Math class. Class math operation.
{
Static float mul (float x , float y)
{
Return x * y :
}
Static float divide (float x, float y);
{
Return x/y;
}
}
```

this : "this" key word refers to the object of the class where it is used. In other words "this" refers to the object to the object of the present class. "This" refers all the members of a class. When object is created, a default reference is also created internally to the object. "This" default reference is nothing but "This".

Abstract classes and methods:

Any class will have all concrete methods implemented to perform a particular task. But are same situations it's not possible to provide full functionality to the methods in the class. At that time we will be declare that method as abstract.

An abstract method doesn't contain any body. It contains only the method header. So we can it is an incomplete method. An abstract class is that generally contains same. Abstract methods. Both the abstract class and to abstract methods should be declared by using the key word "abstract". So it's not possible to estimate the total memory required to create the object so JVM cannot create object to an abstract class. We should create sub classes and all the abstract methods should be implemented in the sub classes. Then it is possible to create objects to the sub classes since they are complete classes.

Syntax :

```
Abstract classes shape
{
        ---
        ---
        Abstract void
        Show ();
        ---
        ---
}
```

- An abstract class is a class that contains 0 or more abstract methods.
- An abstract class can contain instance variable and concrete methods in additions to abstract methods.
- Abstract methods or classes should be declaring with the key word 'abstract'.
- We cannot create an object to the abstract class. Other class must inherit this class.
- The reference of the abstract class can be used to refer to objects of its sub classes.
- The references of the abstract class cannot be used to refer to individual methods of its subclasses.
- We cannot write a class as both abstract and final.

Method Over loading:

In java it is possible to create methods that have the same name, but different parameter lists and different definitions. This is called method overloading. It is used when objects are required perform similar. Java matchers up the method name first and then number and type of parameters to decide which one of the definitions to execute. This process is known as polymorphism.

Rules for overloading:

Then may be a difference in the number of parameter passed to the methods.
There may be difference is the data types of parameters.
There may be difference in the sequence of the parameters.

Example :

```
Class room
{
Float length;
Float breadth;
Room (float x, float y)
{
Length = x;
Breadth = y;
}
Room (float y)
{
Length = breadth = y;
}
Int area()
{
Return (length, breadth)
}
}
```

Here, we an overloading the constructor method room (). Room r1 = new room(25,20) on the hand, if the room is square then we may create the corresponding object as Room r2 = new room (40).

Short answer type questions

1. Define Variable.
2. Define Constants.
3. Types of Constants.
4. What are literals?
5. What are Identifiers?
6. What is Mixed mode?
7. What is Casting values
8. Write about Access specifiers
9. What is constructor?
10. Define method.
11. Define Class.
12. Define Object.
13. Write about Void.
14. What is Instance variable?
15. Write about this.

Long Answer Type Questions

1. Explain Accessing method of a java class.
2. Explain the access specifiers in java.
3. Explain operators in java.
4. Explain logical operators in java.
5. Explain logical operators in java.

UNIT 3

Control statements and Array

The conditional statements

- If statement
- If – else statement
- nested – if statement
- if – else if lader
- switch statement

Java Iteration statements

- While statement
- Do – while statement
- For statement

Break and continue statements

- Break
- Continue

Array

- Definition of array
- Declaration of array
- One dimensional array
- Initialization of array
- Declaration of string
- String array
- Variable size array
- String
- String methods
- String buffer class
- Difference between string buffer and string builder class
- Difference between string and string buffer classes.

Conditional Statement in Java :

Simple if statement:-

Syntax:-if (test _ condition)

```
{
        statement_ block;
}
        statement _ X;
```

In the above simple if syntax, the statement - block may be a single or group of statements. If the test _ condition is true. The statement – block will be executed, otherwise the statement will be skipped and execution will jump to the statement _ X. Remember, When the condition is true, both the statement_ block and statement _ X will be executed.

Flow-chart of simple if:-

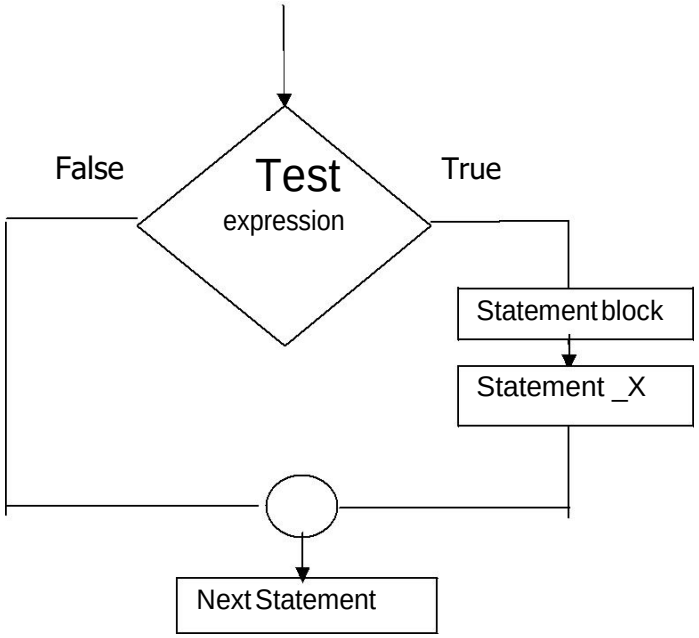


Fig : Flow chart of simple if

Example:- if (total marks >= 400)

```
{
        total_ marks + bonus _ marks;
}
        System.out.println("%d", total marks);
```

In the above example the total marks of the student is >=400, then additional Bonus marks are added to his marks before they are printed total _ marks. Otherwise if total marks < 400 the bonus- marks are not added. The actual given total_marks only printed.

If else statement:

The if – else statement is used to execute true or false.

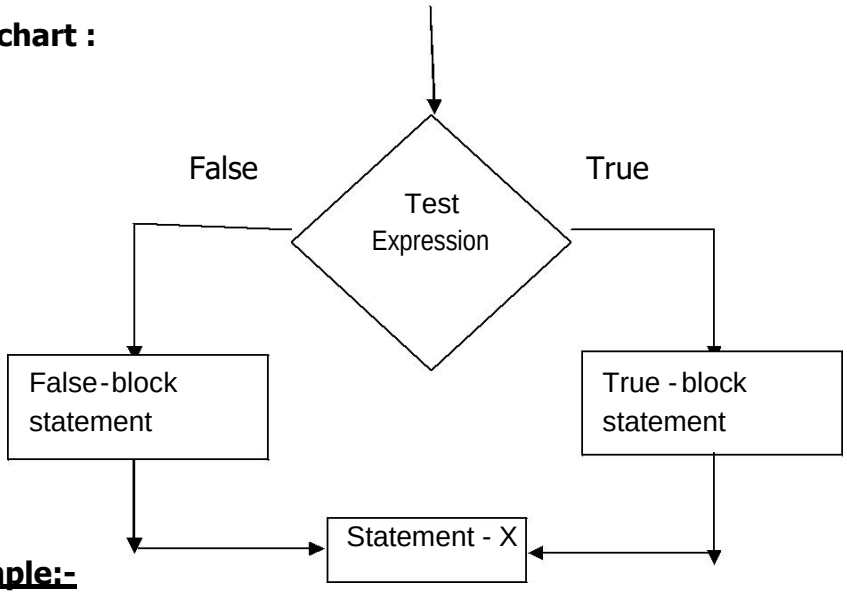
Syntax:-

```
if (test _ expression)
{
 true _ block statements;
}
else
{
false_ block statement;
}

statement _ X;
```

If the test_expression is **true** , then the true_block statements are executed true_ block may have one or more statements if the test_expression is false (else) the false_block statements are executed. In either case, either true_block or false_block will be executed, not both.

flow chart :



Example:-

In a class we want count the student of boys and girls. We use code for 'B' for boys and 'G' for girls.

```
if( student _ code == B)
{
Boys = boys + 1;
}
else
{
Girls = girls + 1;
}
System.out.println ("%d %d", boys, girls);
```

In the above example the test_condition is **true**. The number of boys is incremented by one. If the test_ condition false the girls are increment by one. Then the control reaches the statement _X.

3.1.3.Nested if statement:-

When a series of decision are involved, we may have to use more than one if else statement in nested form.

Syntax:-

```
if ( test _ condition _ 1)
{
        if (test _condition _2)
        {
                statement _ 1;
        }
else
        {
                statement _2;
        }
else
        {
                statement _ 3;
        }
                statement _ n;
```

The execution of nested if is, if the condition _1 is false statement _3 will be executed, otherwise it continued and perform the test condition _ 2 if it also true the statement will be executed. If means to the statement _1 will be executed only when condition _1 and condition _2 is not true i.e false, it mean the 1^{st} condition _1 is true but test_ condition _2 is false. Then the statement _2 will be executed, and then the control is transferred to the statement _ X.

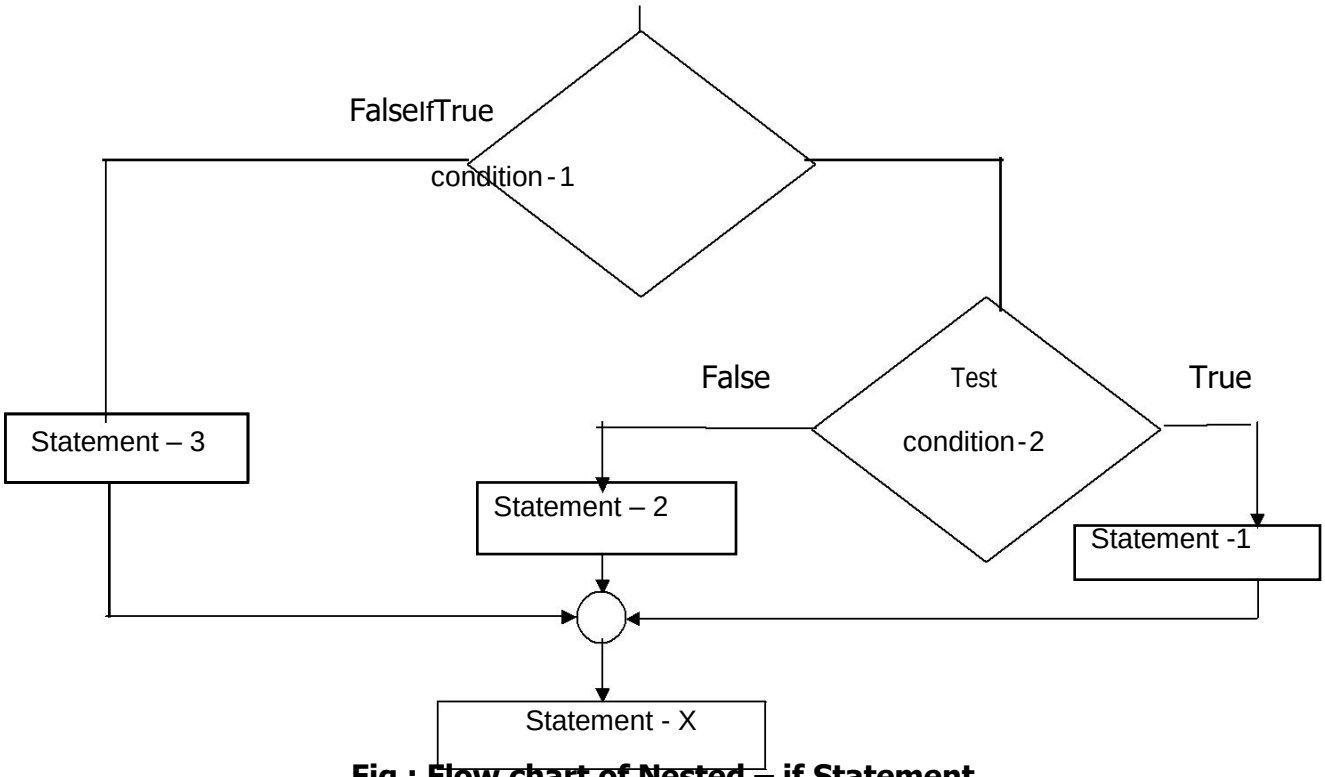


Fig : Flow chart of Nested – if Statement

Example:-

A bank has introduced continue interest to give giving bonus interest all senior citizen holders who deposit more than 50,000. The bonus interest is 5%. If the deposit holder is not to senior citizen, the bonus interest is 0%.

```
if ( customer _ code = senior citizen )
{
        if ( deposit _ amount > 50000)
{interest = interest rate + 0.05
}
        else
        interest = interest + 0.02;
}
        else
        interest = interest + 0.00;
}
        amount = interest + bonus interest.
```

In the above example, if the condition _code == Senior citizen and the deposit amount is >= 50,000 the interest is calculated as actual interest plus 0.05 Bonus interest. i.e. interest = interest rate + 0.05.

If the customer code = senior citizen and the deposit amount is < 50,000 the interest is calculated as actual interest plus 0.02 Bonus interest. i.e. interest = interest rate + 0.02

If the customer code is not = senior citizen and the actual interest is paid only. There is no Bonus interest. i.e. interest = interest rate + 0.00

2.1.4. ELSE……IF ladder:

Another way of using if's together when multipath are involved. A multipath decision is a chain of **ifs** in which the statement associated with each else is an if.

Syntax:-

```
if (condition_1)
 statement _1;
else if ( condition _2)
     statement _ 2;
else if ( condition _3)
     statement _ 3;
          ---
          ----
else
          default _statement;
          statement_ X;
```

This construction is known as else if ladder. The conditions are evaluated from top to bottom. As soon as a true condition is found the statement associated with it executed the control is transferred to statement _ X. It is simple skipped the rest of the statement when all conditions becomes false, then the final else containing the default – statement will be executed.

Flow – chart – to else if ladder statement:

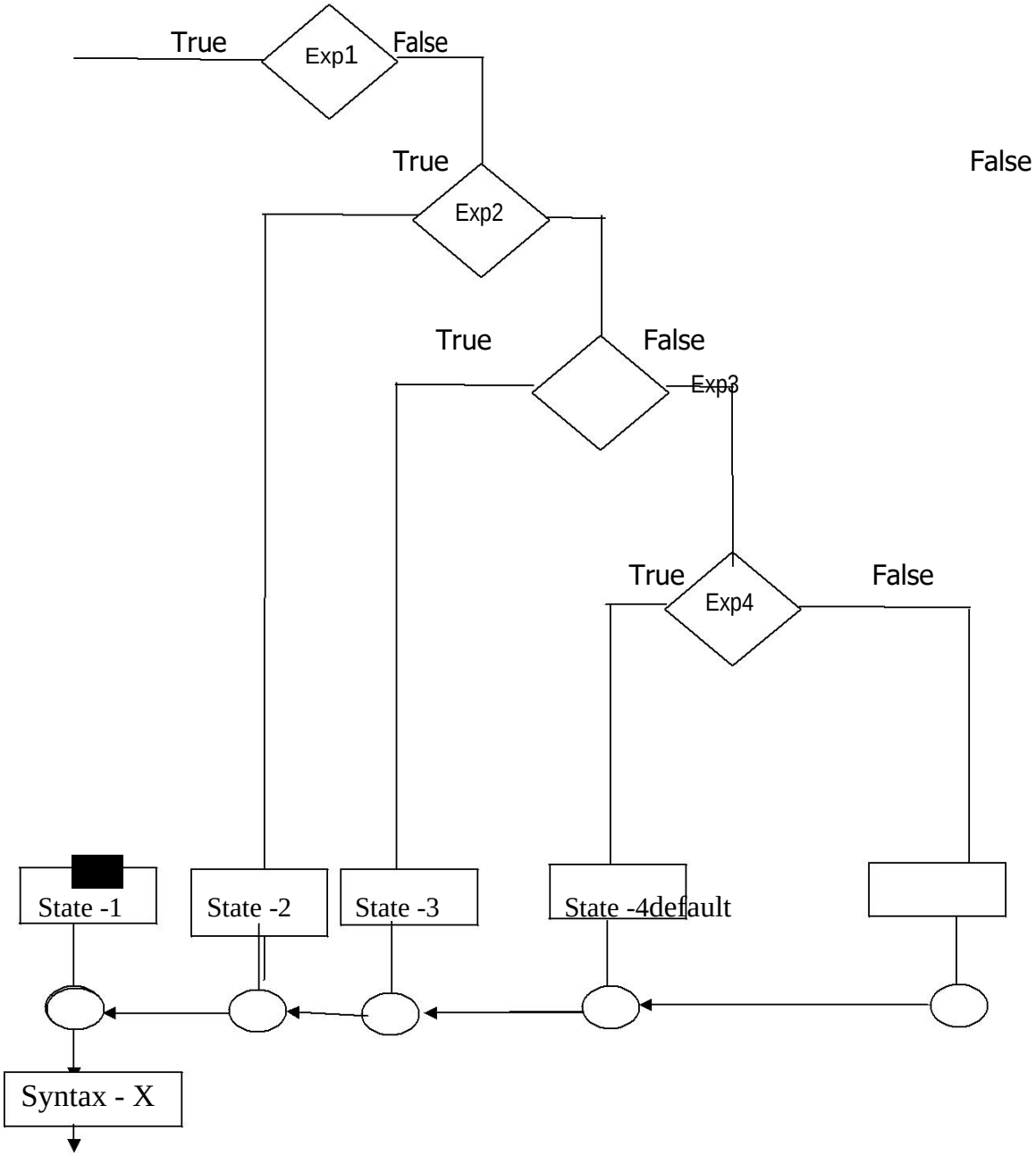


Fig : Flowchart - Else if ladder Statement

Example:- If you give grading the students in your college. The grading is done according to the following marks out of 100.

Marks	Grade
80 – 100	A
60 – 79	B
50 – 59	C
40 – 49	D
Below - 40	Fail

```
If ( marks > 79 )
        Grade = "A"
elseif (marks > 59)
        Grade = "B"
elseif ( marks = 49)
        Grade = "C"
else
        Grade = "Fail"

System.out.Println ("%S", grade);
```

3.1.5 SWITCH STATEMENT:-

C has a built - in – multiway decisions statement known as a switch. The switch statement tests to value of value of a given variable against a list of **case values** and when a match is found, a block of statements associated with that **case** is executed.

Syntax:

```
switch (Expression)
{
        Case Constant-value _1:
                Statement - block _1;
                Break;
              Case Constant-value _2:
                Statement block _2;
                Break;

        Case Constant-value _3:
                Statement block _3;
                Break;
```

```
        Case Constant-value _4:
                Statement block _4;
                Break;

         Case Constant-value _5;
                Statement block _5;
                Break;
        Default:
                default – statements block;
                break;
 }
        Statements – X;
```

The above syntax the expression is an integer expression or character. Constant – values are constants or constants expressions and are known as case labels. Each of these values should be unique within the switch statement. The statement blocks may contains zero or more statements. There no need to put braces around these statement blocks. Note that case label end with : colon. When switch is executed, the value of the expression is successively compared against the constant value 1, constant value_2. …….
Constant value_n. If a case found whose value matches with the value of the expression, then the block of statements that follows the case are executed. The break statement at the end of each block signals the end of a particular case and causes an exit from the switch statement, transferring the control to the statement_x following the switch . The default is an optional case, When present, it will be executed if the value of the expression does not match with any of the case values. If not present, no action takes place if all matches fail and the control goes to the statement_x.

Example:-

If you want declare a grade to the student in class see maximum total marks is 100. With the following conditions.

```
Total marks    80 to 100    grade    = A+
Total marks    70 -  60     grade  =  A
Total marks          >50 - <59   grade    = B
Total marks          >40 - <49   grade    = C
 Total marks         <40         grade    ="Fail".

switch (Total marks)
{
  Case 100:
  Case   90:
  Case   80:
        Grade = "A+";
        Break;

  Case   70:
```

```
                Case 60:
                        Grade = "A";
                        Break;
                Case 50:
                        Grade = "B";
                        Break;
                Case 40:
                        Grade = "C";
                        Break;
               Default:
                        Grade = "Fail";
               }
System.out.Println ("%s", grade) ;
```

LOOPS IN JAVA

WHILE STATEMENT:

Syntax:- while (condition)

```
{

Body of the loop;

}
```

The while statement in an entry controlled loop statement . The condition is evaluated first and if condition is true, then the body of the loop is executed. After execution of the body the test condition is once again tested. Evaluated and I it is true, the body is executed once again. Evaluated once again. This process of repeated execution of the body continues until the condition finally becomes false and control is transferred out of the loop. One on exit, the program continues until the with the statement immediately after the body of the loop. The body of the loop may have one or more statements. The braces ({ }) are needed only if the body contains to or more statements. However it is a good practice to use the braces even if the body has only one has only one statement.

Example:-

```
        s = 0;
        i = 0;
        while (i ≤ 10)
        {
        s = s+i;
        i= i+1;
        }
System.out.Println ("sum = 5d \n", s);
```

The body of the loop is executed 10 times from 1 to 10 each time increment by 1 value of i in inside loop. The condition may also be written I <11, the result would be the same.

Program:- **To calculate sum of 1 to n natural numbers.**

```
    Import java.util.*:
Class sumdigits
{
  public static void main (string args [])
{
  Scanner sc = new scanner (system.in);
System.out println ("enter a value :");
int n=sc.nextInt();
int d, s=0;while (n!=0)
{
  d=n%10;
s=s+d;
n=n/10;}
System.outprintln ("the sum is "'+s);}}
```

DO STATEMENT:-

The do statement is an exit controlled loop statement. The do statement the program proceeds to evaluate the body of loop first. At the end of the loop, the condition is in the while statement if the condition is **true** the program continues to evaluate the body of the loop once again. This process continues as long as the condition is true. When the condition becomes **false**, the loop will be terminated and the control goes to the statement that appears immediately after the while statement.

Syntax:- do

```
{
        Body of the loop;
}
while (condition);
```

Example:-

```
i=0
s=0
do
{ s=s+i;
  i=i+1;
}
while (i ≤ 10);
```

In the above example it reads the 1 to 10 natural numbers the loop to be executed as long as the number keyboard is lies between 0 to 10.

FOR STATEMENT:

The **for** is another entry controlled loop statement that provides a more concise loop control structure.

The syntax of the for loop is:

```
for (initial_value; final_value; increment value )
        { body of the loop;
        }
```

Example:-

```
        S=0;
System.out.println("enter the value");
Scanner in =new scanner (system.in);
N=in nextInt();
{
```

```
  for(i=0;I,<=n;i++)

{

s=s+i:

}

System.out.println("sum="+s);
```

The execution of the control variables is done first in the above example **I** is the control variable. The value of control variable is tested using the condition (final, value). The condition is a relational expression such as <, >, ≤, ≥, !, etc. i≤ 10 that determines when the loop will exit. If the condition is true, the body of the loop is executed, otherwise, the loop is terminated from body of the loop will exit and execution continue with the statement that immediately follows the loop.

When the body of the loop is executed, the control is transferred back to the for statement after evaluated the last statement in the loop. Now, the control variable in incremented using an assignment statement. Such as i=i+1 and the new value of the control variable is again tested to see whether it satisfies the loop condition. If condition is satisfied the body of the loop is again executed. This process continues till the value of the control fails to satisfy the condition.

Example:-
```
for (i=0, i<=10, i++)
        {
        System.out.Println ("%d\n",i);
        }
```

The for loop is executed 10 times and print 10 digits 0 to 9. The three sections enclosed within the parenthesis must be separated by semicolons. Note that there is no semicolon at the end of the loop.

The for statement allow for negative increments i.e decrement the control variable.

Example:-
```
for (i = 9; i>=0;i--)
        System.out.Println ("%d", i);
```

This loop is also executed 10 times, but the output would be from 9 to 0 instead of 0 to Science the conditional test is always performed at the beginning of the loop the body of the loop may not be executed at all. If the condition fails at the start.

BREAK AND CONTINUE STATEMENTS :-

BREAK STATEMENT: The break statement is used to quit from loop. It can be used with in if, switch(), for (), while () and do- while (). When break statement is encountered in loop, automatically the control will come out of the loop.

Syntax :-break;
Example :-

```
Public static void main (string args());
{
int i;
for (i=0; i<=10; i++)
{
if (i==5)
break
System.out.println(i)
}
}
```

CONTINUE STATEMENT :- The continue statement tells the compiler "SKIP THE FOLLOWING STATEMENTS AND CONTINUE WITH NEXT ITERATION". It will work just opposite to the break statement.

Syntax :-continue;

If a continue statement is encountered in a loop then the control will passed to the beginning of the loop. In while and do-while the control will passed to the conditional part where as in for loop the control will passed in increment or decrement part.

Example :-

```
Public static void main (string args [])
{
int i
for (i=0; i<=10; i++)
{
if (i==5)
continue;
System.out.println ("" +i);
}
}
```

3.3 Arrays :-

Array : Sometimes user may have to work with more number of variables with different values in same data – type in a program. It is difficult to specify different names to them. Remembering of all those names are also difficult. To avoid this we use the arrays.

Definition of array :-

" An array is defined as a set of homogeneous (same datatype) data items. They can have a common name and stoned in continues locations of the memory". The individual values of array are called as elements. Array can be any data type item.

Declaration of Array :-

Syntax :- Data type Array – name [size]

In this data type is any one of the java supported data types. Which indicates the type of elements of array. Array – name is valid identifier size indicates the number elements of the array. It should be positive integer. Every array must have a subscript.

Arrays can be classified in to :

- Single dimensional array or Linear array
- Double dimensional array, (Non- Linear array)
- Multi dimensional array

Single dimensional array :-

It is a list of items whose elements are specified by one subscript. It is also called as one dimensional array.

Syntax :- data – type array – name [size];

Example ;

```
int a[10]
int a[] = new int [5] Float
f[ ] = new float[10];
```

Individual data items of the array are accessed by specifying their subscript number. The subscript number of the array always starts zero.

Initialization of array :-

The elements of an array can be assigned we initial values in the array definition with a list initial values enclosed in braces and separated by commas. Ex:- int a[5] = {41, 52, 65, 74,91};

In the above example we declared an array with name **"a"** and size **"[5]".** The 5 integer values are initialized as a [0]=41, a[1] = 52, a[2]=65, a[4]=74, a[5]=91. It is not necessary to specify the size of an array, in case we provide the initialized elements. Example :- int a[] = {"20,35,45,52"}

Array size :

In java, all arrays store the allocated size in variable named size we can obtain the size of the array a using a. size.

Int a size = a. length

Double – or(two) dimensional arrays :-

Arrays whose elements are specified by two subscripts are called two – dimensional arrays. It is also called as double subscripted arrays. Each subscript must contain a positive integer and should always start with 0.

Syntax :-

Data – type array – name [R- size][c – size]; Example : int a[5] [5];

Data –type array name [ ] [ ] = new data type [row-size] [col size];

Example a = new int [3] [4];

int a[ ] = new int [3] [4];

A two dimensional array is a table of items which had a common name. The number of elements in a two dimensional array = number of rows x number of columns.

In the above example – Number of elements = 3 x 4 . Therefore we can store 12 elements.

Two dimensional arrays are also initialized like one dimensional array. They can be initialized with a list of values separated by commas and enclosed in braces

Example :- int a[2][2] = {5,6,8,9};

Or

Int a[2][2] = {{5,6},{8,9}}

In the above example we declared an array with **"a"** and size 2 x 2, the 4 integer values are initialized as

A[0][0]=5, a[0][1] =6, a[1][1]=?

Two dimensional array can also be initialized like Int a[ ] [ ] = {{5,6}, {8,9}}

In the above the size will be interpreted as int a[2][2].

Int a[2][2] = {2, 2};

In the above example, to first element of each row will be initialized as **"2"**, the remaining elements are set to zero.

Variable size arrays (or) Multidimensional array

Java treats multidimensional array as "arrays of Arrays". It is possible to declare a two-dimensional array as follows

int a [ ] [ ] = new int [3] [ ];

A[0] = new int [2];

A[1] = new int [4];

A[2] = new int [5];

These statements create a two-dimensional array as having different lengths for each row.

Strings :- "String represents a sequence of characters".

In java we can handle these strings by using character array as :

char a [ ] = new char [6];

A[0] = 'M', a[1]='U', a[2]='R' a[3]='T' a[4]='H' a[5]='Y' i.e. "MURTHY"

But, we have some disadvantages by using the character array to handle strings. For example if you want to copy one string into another string, we have to traverse each character by using loops.

In java , strings are class objects and implanted using two classes.

string buffer. Java strings are, as compared to "C" strings one or more reliable and predicated. A java string is not a character array and is not null terminated.

Declaration of string :-
String stringname;
String name = new string ("string");

Example :-
String str;
str = new string ("murthy");
These two statements may be combined as allows.
String str = new string ("murthy");
It is possible to get the length of the string using the length of the string class. Int N = str.length();

String array :-
We can also create and use arrays that contains strings.
That is String str [ ] = new string [6];
The str of size 6 to hold three string constants. We can assign the strings to the str elements by elements using 6 different statements or more efficiently using a for loop. String methods :-
The string class defines a number of methods that allows us to accomplish a variety of string manipulation task.

Method call	Task performed
1) S2 = S1.to lower case ();	Convert the string S1 to all lower case.
2)S2=S1. Upper case();	Convert the string S1 to all upper case.
3) S2=S1. Replace ('a','b')	Replace all appearances of a with b
4) S2=S1. Equals ignore case (s2)	Returns true if s1 is equal to s2, ignoring the case of characters.
5) S1.equals (S2)	Returns a true if S1 is true to S2
6) S2=S1 trim()	Removes white spaces at the beginning and end of the string S1.
7) S1.length ()	Gives the length of S1
8)S1.character	Gives the n^{th} character of S1
9)S1. Comparator (S2)	Returns negative if (S1<S2), positive if (S1>S2) and zero if S1 is equal to S2.
10)S1.concat (S2)	Concatenates S1 and S2

11) S1.substring (n)	Gives substring starting from n^{th} character
12) S1.substring (n, m)	Gives substring starting from nth character up to in the character (not including m^{th})
13) String.value of (P)	Creates a string object of the parameter P(simple) type of object
14) P. to string ()	Creates a string representing of object 'P'
15) SI. Index of ("A")	Gives the first occurrence of 'A' in the string SI
16) S1.index of ('X,n)	Gives the position of 'X' that occurs after nth position in the string S1.
17) String value of (variable0	Converts the parameter value to string representation

Strings buffer class :-

It is a pure class of string, while string creates strings of fixed- length, string buffer creates strings of flexible length that can be modified in terms of flexible length that can be modified in terms of both length and content. We can insert characters and substrings in the middle of string, or append another string to the end. Some of the methods frequently used in the string buffer class.

String Buffer append (val) :Val may be Boolean int, short, long etc. it will be added to string

buffer object

String buffer sb=new string buffer("SREE");

Sd.append ("RAMA")

String Buffer insert (int I , val)

Val may be any data type, that will be inserted in to the string buffer at the specified position

by i.

*String Buffer delete (int I, int j);

It removes the characters from i^{th} to $J\text{-}1^{th}$ position in the string buffer.

*String Buffer reverse () : It reverse the character sequence in the string buffer

*int length () : It returns the no of characters in the string buffer object.

*String to string () : This is used to convert string buffer to string object

*int idex of (string str) : It returns the first occurrence of 'str' in string buffer Object

*String Buffer set char at (n,x) : It returns the modified the n^{th} character to X

*Int set length (n) : sets the length of the string S1 to n. if n<S1 length (). S1 is truncated if n>S1. Length() zeros one added to S1.

String Buffer class :-

It is also similar to string buffer, it has been added in Jdk 1.5. This is also generates mutable string object like string Buffer.

Methods :-

String builder append (val)
String builder insert (int I, val)
String builder delete (int I, int j)
String builder reverse ()
String to string ()
Int length ()
Int index of (string str)
String builder replace (int I, int j, string str)
String substring (int I)
String substring(int I, int j)

-

What is the difference between string Buffer and string builder classes?

String class is synchronized and string builder is not synchronized. When the programmer wants to use several threads. It should be string Buffer as it gives reliable result. It only one thread is used string builder is preferred, as it improves the execution time.
The difference between string and string Buffer classes?
String class object one immutable and hence their contents cannot modified string Buffer class objects are mutable so they can be modified. Some methods are available in string buffer to manipulate the data.

Short answer Type questions

1. **Write about if statement.**
2. **Write about Break statement.**
3. **Write about Continue statement.**
4. **Define an array.**
5. **Write different types of arrays.**
6. **What is a string array**
7. **Write about Buffer classes.**
8. **Write about String method.**
9. **Write about Nested loops.**

Long answer type questions

1. **Explain the conditional statement in java.**
2. **Explain switch statement with example.**
3. **Explain the loops in java.**
4. **What are the string method explain**
5. **Explain the string Buffer class in Java.**
6. **What are the differences between string buffer and string builder classes.**
7. **Write a java program to print 1 to n.**
8. **Write a java program addition of two matrices.**
9. **Write a java program to sort "N" numbers in an array.**
10. **Write a java program multiplication of two matrices.**
11. **Write a java program to find the factorial of given number.**
12. **Write a java program to find whether the given number is perfect or not.**
13. **Write a java program to transpose of given matrices.**

UNIT 4

Implementing oops in Java

Java Methods

4.1.0. What is method.
4.1.1.Types of Java methods.
4.1.2.How to call Java method.
Java methods with arguments and return value.
Advantages of methods.
4.2.0.Inheritance in Java with example.
4.2.1.Types of Inheritance
4.2.2. Constructor and inheritance
4.2.3.Method overloading
4.2.4. Over loading.
4.2.5.Operator Overloading.
4.2.6. Difference between overloading and overriding.
4.2.7.Final variables and methods.

IV. IMPLEMENTING OOPS IN JAVA
JAVA METHODS

4.0. What is a Method ?

In mathematics, you might have studied about functions for example, f(x)= x^2 is a function that returns squared value of x

If x=2, then f(2) = 4

If x=3, f(3) = 9 and so on.

Similarly in programming, a function is a block of code that performs a specific task.

In object – oriented programming, method is a jargon used for function. Methods are bound to a class and they define the behavior of a class.

TYPES OF JAVA METHODS:-

Depending on whether a method is defined by the user, or available in standard library, there are two types of methods

(a) Standard library methods
(b) User – defined methods

(a) Standard Library Methods :-

The standard library methods are built in methods in java that are readily available for use, the standard libraries come along with the java class library (JCL) in a java archive (*jar) file with JVM and JRE.

For example :-

Print () is a method of java.io.Print steam. The print ("…..") prints the string inside quotation.

Sqrt() is a method of math class. It returns square root of a number.

Here's an working example :

```
Public class number {
        Public static void main(string []args){ System.out.print("Square
                root of 4 I :"+ math.sqrt(4));
        }
}
```

When you run the program , the output will be square root of 4 Is 2.0

(b) User-defined Method :-

You can also define methods inside a class as per your wish. Such methods are called user defined methods.

How to create a user-defined method?

Before you can use (call a method) you need to define it. Here is how you define methods in java.

```
Public static void Methods() {
System.out.println("My function called");
}
```

Here a method named my Method () is defined

You can see three key words public, static and void before the function name.

The public key word makes my method() method public. Public members can be accessed outside of the class.

The static keyword denotes that the method can be accessed from outside of the class.

The void keyword signifies that the method doesn't return any value.
You will learn about returning value from the method later in this article.

In the above program, our method doesn't accept any arguments. Hence the empty parenthesis (). You will learn about passing arguments to a method later in this article.

The complete syntax for defining a java method is

```
Modifier static return type name of method (Parameter List) {
        // method body
}
```

Here,

Modifier ⟶ defines access type whether the method is public, private, protect and so on.

Static ⟶ if you use static keyword in a method then it becomes a static method.

Static methods can be called without creating an instance of class.

For example, the sqrt() method of standard math class is static. Hence we can directly call math. Sqrt() without creating an instance of math class.

Return type :- A method can return a value.

It can return native data types (int, float, double etc.,), native objects (string, Map, List etc), or any other built in and user defined objects.

If the method does not return a value, its return type is void.

Name of method :-The name of method is an identifier

You can give any name to a method. However it is more conventional to name it after the tasks it performs. For example , calculate Interest, calculate area and so on.

Parameters (arguments) :- Parameters are the values passed to a method. You can pass any number of arguments to a method.

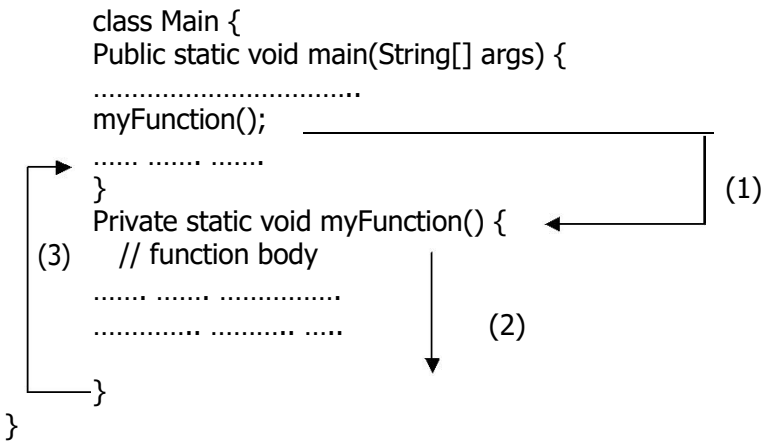
Method body : - It defines what the method actually does, how the parameters are manipulated with programming statements and what values are returned. The codes inside only braces { } is the body of the method.

How to call a Java Method ?

Now you defined a method you need to use it for that, you have to call the method Here's how :

myMethod ();

This statement calls the myMethod() method that was declared earlier.



While java is executing the program code. It encounters myMethod (); in the code.

The execution then branches to the my function() method, and executes code inside the body of the method.

After the codes execution inside the method body is completed, the program returns to the original state and executes the next statement.

Example :- Complete program of Java Method

Let's see a Java method in action by defining a Java class

```
Class main {
Public static void main(String [] args){
System.out.println ("About to encounter a method");
        method call
      myMethod;
System.out.println("Method was executed
successfully"); }

// method definition
```

```
private static void myMethod() {
System.out.println ("Printing from inside myMethod()!");
        }
}
```

When you run the program, the output will be:

```
About to encounter a method
Printing from inside myMethod()
Method was executed successfully
```

The Method myMethod() in the above program doesn't any arguments. Also, the method doesn't return any value (return type is void).

Note that, we called the method without creating object of the class. It was possible because myMethod() is static.

Here's another example. In this example, our method is non-static and is inside another class.

```
class main()
 Public static void main (String [] args) {
Output obj = new Output ();
System.out.println ("About to encounter a method");
//calling myMethod() of output class
obj.myMethod();

System.out.println("Method was executed successfully!");
}
}

Class output {

//public: this Method can be called from outside the class

Public void myMethod() {
System.out.println ("printing from inside myMethod()")
}
}
```

When you run the program, the output will be:

```
About to encounter a method.
Printing from inside myMethod()

Method was executed successful!
```

Note that, we first created instance of output class, then the method was called using obj object. This is because myMethod() is a non-static method.

Java Methods with Arguments and Return Value

A Java Method can have zero or more parameters. And, they may return a value.

Ex : Return Value from Method

Let's take an example of Method returning a value.

```
Class square Main {
Public static void main (String [] args)
{
Int result;
Result = square ();
System.out.println("squared value of 10 is :"+ result);

}

Public static int square () {
  return statement
  Return 10 * 10 ;
}
}
```

When you run the program, the output will

be Squared value of 10 is : 100

In the above code snippet the method square () does not accept any arguments and always returns the value of 10 squared.

Notice, the return type of square()method is int meaning, the method returns an integer value.

```
Class Square main {
    Public static void main(String [] args) {
            ---- ----
    100 result = square ();
            ---- -----
    }
    Private static int square ( ) {
              return statement
            Return 10 * 10;
    }
}
```

As you can see, the scope of this method is limited as it always returns the same value Now let's modify the above code snippet so that instead of always returning the squared value of 10, it returns the squared value of any integer passed to the method

Example :- Method Accepting Arguments and Returning Value

```
public static void main (String [] args) {
        int result, n;
        n=3;
        result = square (n);
        System.out.println ("Square of 3 is :" + result);
        n=4;
        result = square (n);
        System.out.println ("Square of 4 is : "+ result);
}

        Static int square (int i)  {
        return I * I;
        }
}
```

When you run the program, the output will be :
Squared value of 3 is : 9
Squared value of 4 is : 16
Now, the square () method returns the squared value of whatever integer value passed to it.

```
class Squaremain  {
        public static void main (String [] args) {
        ---   ---    ---
        N = 3;          (3)
        (9) result = square (n);
        -----          ----
        }
        Private static    square (int i)  {
                // return
                        Return i * i;
        }                          (9)
}
```

Java is a strongly – typed language. If you pass any other data type except int (in the above example), compiler will throw an error.

The argument passed n to the get square () method during the method call is called actual agreement.

result = get square (n);

The parameter i accepts the passed arguments in the method definition get square
(int i). This is called formal argument (parameter). The type of the formal argument must be explicitly typed.

You can pass more than one argument to the java method by using commas. For example ,

```
public class Arithmatic main {
        public static int get Integer sum (int i, int j) {
                return i+j;
        }
        public static int multiply Integer (int i, int j) {
                return x * y;

        }
        Public static void main (String [] args) {
        System.out.println (10 + 20 =”+ get integer sum (10,20));
        System.out.println (20 + 40 =”+ multiply integer (20,40));
        }
}
```

When you run the program, the output will be:

```
 10 + 20 = 30
20 X 40 = 800
```

The data type of actual and formal arguments should match i.e.,. The data type of first formal argument. Similarly the type of second actual argument must match the type of second formal argument and so on.

Ex : Get Squared value of Numbers from 1 to 5

```
Public class I methods{
        // method defined
        private static int get square ( int x) {
        return x * x;
}
        public static void main (String [] args) {
        for (int i=1; i<=5; i++)
                // method call
                result = get square (i)
                System.out.println(Square of “+I” is :”+ result);
        }
}
```

When you run the program, the output will be:

Square of 1 is : 1

Square of 2 is : 4

Square of 3 is : 9

Square of 4 is : 16

Square of 5 is : 25
In the above code snippet the method get square()takes int as a parameter.
Based on the argument passed, the method returns the squared value of it.
Here, argument I of type int is passed to the get square () method during method call Result = get square(i);
The parameter x accepts the passed argument [in the function definition get square (int x)].
Return i*i; is the return statement. The code returns a value to the calling method and terminates the function.
Did you notice, we reused the get square method 5 times?

What are the advantages of using methods :

The main advantages is code reusability. You can write a method once, and use it multiple times. You do not have to rewrite the entire code each time. Think of it as " Write once, reuse multiple times."

Methods make code more readable and easier to debug :

Ex: Get salary Information() method is so readable , that we can know what this method will be doing than actually reading the lines of code that make this method.

Polymorphism in java with example

Polymorphism is one of the oop's feature that allows us to perform a single action in different ways. For example, we have a class animal that has a method sound (). Since this is a generic class so we can't give it a implementation like rower, meow, rink etc. we had to give a general message

```
Public class animal
{
Public void sound()
{
System.out.println("animal is making a sound");
}
}
```

Now lets say we two subclasses of animal class :

Horse and cat that extends (see inheritance) animal class. We can provide the implementation to the same method like this Public class horse extends animal

```
{
public void sound()
{
System.out.println("Neeigh");
}
}
```

And

```
public class cat extends animal
{
public void sound()
{
System.out.println("Meow");
}
}
```

As you can see that although we had the common action for all subclasses sound() but there were different ways to do the some action. This is perfect example of polymorphism. It would not make sense to just call the generic sound()method as each animal has a different sound. This we can say that the action in method performs is based on the type of object.

Inheritance in java with example :

Process by which one class acquires the properties (data member) and functionalities (methods) of another class is called inheritance. The aim of **inheritance** is to provide the reusability of code so that a class has to write only the unique feature of and reset of the common properties and functionalities can be extended from the another class.

Child class :- The class that extends the features of another class is known as child class. Sub class or derived class.

Parent class :- The class whose properties and functionalities are used by another class is known as parent class, super class or base class.

Inheritance is process of defining a new class based on an existing class by extending its common data numbers and methods. Inheritance allows us to reuse of code. It improve reusability in your java application. The advantages of inheritance is that the code that is already present in base class need not be rewritten in the child class. This means that the data members (instance variable) and methods of the parent class can be used in the child class.

Syntax : - To inherits a class we use "extends" keyword. Here class ABC child class and class DEF is parent class. The class ABC is inheriting the properties and methods of DEF class

```
class DEF extends ABC
{
  }
```

Example :-

In this, we have a base class lecturer and sub class computer lecturer. Since class computer lecturer extends the designation and college properties and work() method from base class, we need not to declare these properties and method in subclass. Here we have college name, designation and work() method which are common to all the lecturers so we have declared them in the bas class, this way the child classes like maths lecturer, physical- director, etc., does not need to write this code aand can be used directly from base class.

```
class lecturer
{
String designation = "Lecturer";
String college name = "GJC";
Void does ()
{
System.out.println("Teaching");
}
}
public class Mathslecturer extends lecturer
{
        String main subject = " Maths"
        public static void main (String args[])
        {
                Mathslecturer obj = new maths lecturer();
                System.out.println (obj. collegename);
                System.out.println (obj. designation);
                System.out.println (obj. main subject);
                obj.does();
                }
}
```

Output:-

```
GJC
Lecturer
Maths
Teaching
```

Based on above example we can say that maths laecturer is – A lecturer. This means that child class IS – A relationship with the parent class. This is inheritance is known as Is- A relation ship between child and parent class. The derived class inherits all the members and methods that are declared a public or protected. If the members or methods of super class are declared as private then the derived class cannot use them directly. The private members can be accessed only in its own class. Such private members can only be accessed using public or protected getter and setter methods of super class as shown in the example below.

```
class lecturer
{
private string designation = "Lecturer";
private string college name = "GJC";
public string get designation ()
{
return designation;
}
protected void set designation (string designation)
{
this. Designation= designation;
}
protected string get college name ()
{
return college name;
}
return college name;
}
protected void set – college name (string college name)
{
this. college name= college name
}
Void does()
{
System.out.println ("teaching")
}
public class java example extends teacher
{
String main subject = "maths";
public static void main (String args[])
{
Java example ojt = new java example();
System.out.printl n(obj .get college name())
System.out.printl n(obj .get designation())
System.out.printl n(obj .get main subject())
obj.does();
}
}
```

Output:-
GJC
Lecturer
Maths
Teaching

The improvement point to note in the above example is that the child class is able to access the private members of parent class through protected method of parent class. When we make a instance variable or method protected. This means that they are accessible only in the class itself and in child class. These public and protected private etc all access specifies.

Multiple inheritance :

Multiple inheritance is the ability of a single class to inherit from multiple classes. Java does not have this capability in case of class . but it is supported in case of interface because there in no ambiguity as implementation is provided by the implementation class.

Syntax:-

```
interface A
{
----
----
-----.
}
interface B
{
----
----
-----.
}
Class C implemented A, B
{
----
----
-----.
}
```

The java programming support multiple inheritance of type, which is the ability of a class to implement more than one interface. An object can have multiple types, the type of its own class and the types of all the interfaces that the class implements. These means that if a variable is declared to be the type of an interface, then its value can reference any object that is instantiated from any class that implements the interface.

$$\text{Speed} = \frac{\text{Distance}}{\text{time}}$$

Distance = speed *time

$$\text{Time} = \frac{\text{Distance}}{\text{Speed}}$$

```
Interface auto
{
int speed = 50
void distance ();
}
interface car
{
int time = 60;
void speed();
}
interface bus
{
int distance = 100;
void time();
}
Call multi implementation auto, car, bus
{
public void distance();
{
System.out.println("the distance is :" + (speed * time));
}
public void speed()
{
System.out.println("the speed is :" + (distance/ time*100));
}
public void time ()
{
System.out.println("The time is:"+(distance * (speed/60));
}
}
class multidemo
{
public static void main (string args[])
{
Multi obj = new multi();
obj.distance();
obj.speed();
obj.time();
}
}
```

Inheritance :

The java classes can be reused in several ways. Creating new classes, reusing the proper. It is of existing one, basically does this, the mechanism of deriving a new class from and old one is called inheritance. The old class is known as the base class is sub class or derive class or child class. The inheritance allows sub classes to inherit all the variables and methods of their parent classes. Inheritance may take different forms.

single inheritance (only one super class)
multiple inheritance (several super classes)
hierarchical inheritance (one super class many sub classes)
multilevel inheritance (derived from a derived class)

but java doesn't support directly to implement multiple inheritance. But this concept is implemented using interfaces.

Advantages of inheritance :-

Reduce code redundancy
Provide code reusability
Provides code source size and improves code readability
Codes easy to manage and derived into parent and child class
Supports code extensibility by overloading to base class functionally child class

Disadvantages of inheritance :-

In heritance base class and child classes are tightly coupled
Hence if you change the code of parent class it will get effects to the all to chaild classes.
In class hierarchy many data members remain unused.
The memory allocated to then is not utilized.
Hence affect performance of your program if you have not implemented inheritance correctly.

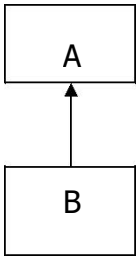
Types of inheritance :-

Types of inheritance are

a. Single inheritance
b. Multilevel inheritance
c. Hierarchical inheritance
d. Multiple inheritance
e. Hybrid inheritance

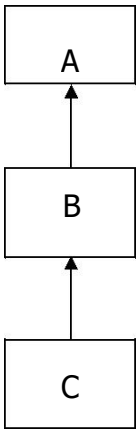
1. Single Inheritance :-

It refers to a child and parents class relationship where a class extends the another class.



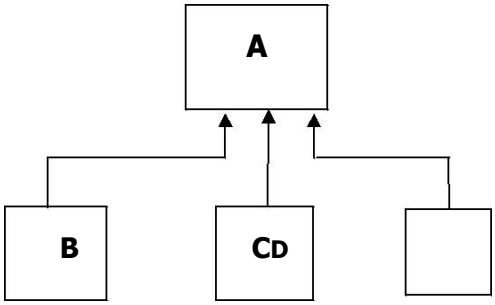
2. Multilevel inheritance :-

It refers to a child and parent class relation ship where a class extends the child class for example class C extends class B and class B exit ends class A



3. Hierarchical Inheritance :-

It refers to a child and parent class relation ship where more than one classes extends the same class. For example, classes B,C and D extends the same class A



4. Multiple Inheritance :-

It refers to the concept of are extending more than one classes. Which means a child class had two parent classes. For example class C extends both classes A and B. But java doesn't support multiple inheritance read more about it here.

Multiple inheritance is the ability of a single class to inherit from multiple classes. Java does not have this capability in case of class . but it is supported in case of interface because there in no ambiguity as implementation is provided by the implementation class.

Syntax:-

```
interface A
{
----
----
-----.
}
interface B
{
----
----
-----.
}
class C implemented A, B
{
----
----
-----.
}
```

The java programming support multiple inheritance of type, which is the ability of a class to implement more than one interface. An object can have multiple types, the type of its own class and the types of all the interfaces that the class implements. These means that if a variable is declared to be the type of an interface, then its value can reference any object that is instantiated from any class that implements the interface.

$$\text{Speed} = \frac{\text{Distance}}{\text{time}}$$

$$\text{Distance} = \text{speed} * \text{time}$$

$$\text{Time} = \frac{\text{Distance}}{\text{Speed}}$$

```
interface auto
{
int speed = 50
void distance ();
}
interface car
{
int time = 60;
void speed();
}
interface bus
{
int distance = 100;
void time();
}
Call multi implementation auto, car, bus
{
public void distance();
{
System.out.println("the distance is :" + (speed * time));
}
public void speed()
{
System.out.println("the speed is :" + (distance/ time*100));
}
public void time ()
{
System.out.println("The time is:"+(distance * (speed/60));
}
}
class multidemo
{
public static void main (string args[])
{
Multi obj = new multi();
obj.distance();
obj.speed();
obj.time();
}
}
```

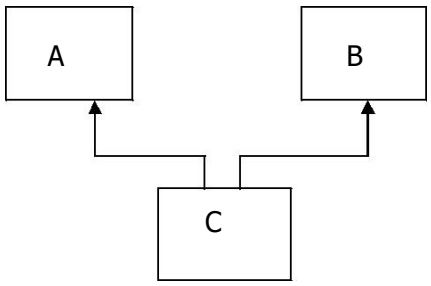


Fig : Multiple Inheritance

5. Hybrid Inheritance :-

Combination of more than one type of inheritance is a single program. For example A & B extends class C and another class D extends class A then this is a Hybrid inheritance example because it is a combination of single and hierarchical inheritance.

Constructors and inheritance :-

Constructor of sub class is invoked when we create the object of subclasses, by default invokes the default constructor of super class. Hence, in inheritance the objects are constructed top -down. The super class constructor can be called explicitly using the super keyword, but it should be first statement in a constructor. The super keyword refers to the super class, immediately above of the calling class is the hierarchy. The use of multiple super keyword to access an ancestor class other than the direct parent is not permitted.

Example :-

```
class Parentclass
{
Parentclass ()
{
System.out.println("constructor of parent");
}
}
class Javaexample extends Parentclass
{
Javaexample()
{
System.out.println (" constructor of child");
}
public static void main (String args[]);
new Javaexample();
}
}
```

Output :-

Constructor of parent
Constructor of child

Method overloading :-

When we declare the same method is child class which is already present in the parent class this is called method overloading. In this case when we call the method from child class object, the child class version of the method is called. However we can call the parent class method using super keyword.

Example :-

```
Class parent-class
{
Parent class();
{
System.out.println("construct of parent");
}
void disp()
{
System.out.println ("parent method");
}
}
class java example extends parentclass
{
Javaexample ()
{
System.out.println ("Constructor of child");
}
void disp()
{
System.out.println ("child method");
Super disp();
}
Public static void main(string args[])
{
Java example obj = new java example ();
Obj.disp();
}
}
```

Out put :-

Constructor of parent
Constructor of child
Child method
Parent method

Overloading :-

Overloading allow the different methods to have same name, but different signatures where signature can differ by number of input parameters or type of input parameter or both. Overloading is related to compile time polymorphism.

Definition:- overloading refers to the ability of use a single identifier to define multiple methods of class that differ in their input and output parameters. Overloading methods are generally used when they conceptually execute the same task but with slightly different set of parameters.

Overloading concept used to avoid redundant code where the same method name is used multiple times but with a different set of parameters. The actual method that gets called during runtime is resolved at compile time, thus avoiding runtime errors. Overloading provides code clarity eliminates complexity and enhance runtime performance. Overloading is used in programming language that enforces type checking in function calls during compilation. When a method is overloaded, the method chosen will be selected at compile time. This is not the same as virtual function where the method is defined at runtime.

Example :-

```
Public class Sum
{
public int sum (int a, int b)
{
return (a+b);
}
public int sum (int a, int b, int c)
{
return (a+b+c);
}
public double sum(double a, double b, double c);
{
return (a+b+c);
}
public static void main (String args[])
{
Sum s = new sum();
System.out.println (s. sum (20, 30));
System.out.println (s. sum (50,60,80));
System.out.println (s. sum (20.5, 50.5, 30.5));
}}
```

Output :-

```
50
190
101.5
```

Operator overloading :-

Java doesn't support user defined operators over loading. The only aspect of java which comes close to "custom" operator overloading is the handling of **"+"** for string, which either result in compile – time concalination of constants or execution – time concatenation using string builder or string buffers which act in the same way through. In addition to all the people pointing out that + is overloaded for string **" -"** over loaded for both floating point and integer operators as are * and /. % is also overloaded for floating point which can be a bit of surprise for those with a C or C++ back ground.

When you add a non string operand such as an integer or char to a string, the non-string operand is converted to a string and string concatenation happens. If both operands are char literals the + operator perform addition either than connecting by promoting each of the char – valued operands to int values through widening primitive conversion.

Example :-

```
Public static void main (String args[])
{
Final string first = " length: 25;
Final string second = "length + first. Length();
System.out.println ("first and second are equal :" + first == second );
}
```

It prints false and nothing else

The + operator, whether used for addition or string concatenation has more precedence than the == operator. Therefore the parameter of the println method is evaluated like.

System.out.println ("first and second are equal: "+ first == second);

Example 2 : String concatenation with short hand operator

Find output :

```
String str = "500"
Str += 50;
System.out.print (str);
```

We can expand str += 50 as str = str+50. When you add a non string operand such as an integer or char to a string the non string operand is converted to a string and string concatenation happens. Therefore "50" is concatenated to 500 to the 500 giving "50050".

Final Variables and Methods :

All methods and variables can be over ridden by default in subclass. If we want to prevent the subclass from overriding the members of super class. We can declare them as final using the keyword final as a modifier. Methods which are declared as "final" are called final method. Final method cannot be overridden, because they are not avaialabel to the sub classes. Therefore only method overloading is possible with the final methods.

Ex : final int size = 100;

Final void show staturs () { …..}

The value of a final variables can never be changer. Final variable, behave like class variables and they do not take any space on individual object of the class.

Final classes : A final is a class which is declared as final find keyword before a class prevents inheritance. This means subclasses cannot created to a final class.

Final class A{ ….}

Final class B extends some class {.....}

Any attempt to inherit these will cause an error at compilation time.

Final Methods :

We know constructor method is used to initialize an object when it is created. This process is known as initialization. Similarly java support a concept called finalization. Which is just possible to initialization. In java garbage collected is used free the object in the memory, but sometimes object may hold other non-object resources like descriptions or window system fonts. The garbage collector can not these resources, we must use a finalize method.

SHORT ANSWER TYPE QUESTIONS.

1. What is a method.
2. How to call method.
3. What are the advantages of using methods.
4. What is Constructor?
5. What is Over loading?
6. What is Inheritance?
7. What is Polymorphism?
8. What is Overriding?

Long Answer Type Questions.

1. Explain the Java methods with arguments and return value.
2. Explain the final variables and methods.
3. Explain the inheritance with example in java.
4. Write about Types of inheritance in Java.
5. Explain the method overloading with example in Java.
6. Explain the operator overloading in Java.
7. Explain the polymorphism with example.

UNIT 5

PACKAGES AND ITERFACES

Introduction
Defining packages
Advantages of packages
5.3Types of packages
5.4 Creating package
5.5Accessing a package
Adding a class to package
Interface
How to provide the implementation to the interface by implementation a class
Abstract class vrs. Interface
Defining a sub – class

Introduction In java we need to used classes from the programs without physically copying them into the program this can be accomplished in java by using what is known as **packages**. In this way we can achieve the reusability in java.

Packages is a concept "that grouping a variety of classes and / or interfaces together". The packages act as "containers" for classes.

Defining of package :

A package is collection of classes grouped to gather for achieving responsibility under a common name packages are a way of grouping a variety of class together. The grouping is done according to their functionality simple packages is container of class.

Syntax : package <package name >

Ex : package student

Ex :- Packages begin with lowercase letters

double y = java.math.squart(x);

long . package name

Math – class name

sqrt – method name

Advantages of packages :

The classes contained in the package of other program can be easily reused

Two classes in two different packages can have same name

It provides a way to "hide "classes thus preventing other programs.

Its also provides a way of separating "design" and "coding"

Types of packages :- Packages are two types. Java

API packages

Java user defined packages.

Java API packages :-

They provides a large number of classed grouped in to different packages according to functionality like long. Util , out. Applet etc. each package provides different functionality to our java program. The different packages are :

Java Long :- It includes class primitive data types, string math functions.

Java .util :- Languages, utility classes such as vector, hash tables, random numbers, date etc.

Java . Applet : - It includes classes for windows buttons list menus and abstract window tool kit.

Java. IO :- Input output support classes.

Java . net :- Net stands for network. Its mainly in client – sever programming.

Java.text :- this package is useful to format the numeric values (Date format class for dates) [number format class for values]

Java.sql:- Sql stands for structured query language. This package helps to connect to database like or Sybase, retrieve data from there and used in the java application.

User_ defined packages :-

While declaring package first declare the name of the package using the packages keyword followed by package name must be the first line in java

Syntax :- package < package name>;
Example:- package student;

Using one class from other package. The following sample program will be class from other package

Example :-

```
package x;
public class y
{
        public void display ()
        {
         System.out.println ("class y");
        }
}
```

The above example has a package name x and containing a simple class y. this should be name y.java stored in the subsidiary "x". if we compile this java file, resultant "y" class with stored in the same sub- directory.

Creating package :-

First we declare name of the package using package keyword followed by the package name. This must be first statement in a java source file. Then we define classes for that package.

Ex :- package first package;

```
Public class first-class
{
        --------
        --------
        --------
}
```

Hence the package name **first package.** The class **first** is now considered a part **class** of directory and named **first package**. When the same compiled it will create a file is **class.** File and store it in the same directory.

Class file must be located in a directory that has the same name as the package and this directory should be subdirectory of the directory where classes that import the packages are located.

Step to create own package

Create a subdirectory with the same as package name under directory where the main source files are stored.

Open the notepad editor and type the following

Declare the package at the beginning of a file **"package package name "**

Define the class that is going to place in the package, declare its as

public Example:-

```
public class classA
{
    public void display ()
    {
     System.out.println("class A");
    }
}
```

Save the java file as the class name. java (i.e class A.java) file in the subdirectory created (package name). Package name java class A.java

Compile the file, this creates class file in the sub directory.

also group related classes into a large package.

Accessing a package :-

We use the import statement to import the classes packages. The import statement can be used to list of packages for a particular class. The general form of the import statement is

Syntax :- import package 1 [.package 2] [.package2] [.package3]. class name

Hence package1 is the name of the top-level package. Package2 is the name of the package that is inside the package1, and so on. Finally, the explicit class name is specified.

The import statement ends with semicolon(;). This should appear before ant class definition in a source file. Multiple import statements are allowed. Example :

Import first package. Second package. My class another

approach is Import package name.*

"*" indicates that the compiler should search the entire package hierarchy. It implies that we can access all classes contained in the above package directly.

Using a package :-

This below example shows a package named package 1 containing a simple class class A

```
package package1
public class classA
```

```
{
  public void display ()
        {
          System.out.println ("class A");
        }
}
```

Source file should be named classA. Java and stored in the any package1. Now compile this java file. The result is 1. Class will be stored in the same subdirectory.

```
Import package1. Class A
class package test 1
{
  public static void main (string args[])
{
  class obj = new class A();
        Obj.display();
        }
}
```

Here we are importing the classA from the package 1. The source file should be saved as **"packagetest1.java "** and then compiled. The source file and compiled and compiled file would be saved in the directory of which package1 was a subdirectory. Now we can run the program.

During the compilation of package1 test. Java the compiler checks for the file classA. Class is the package 1 directory, but it doesn't include the code from class A. class in the file package test 1. Class, while running the program java looks for the file package test1. Class and loops it something called class loader.

Adding a class to package :

It is to add a class to an exciting package. Consider the following package :

```
package p1; Public class A
{
Body of A;
}
```

The package p1 contains one public class by name class A suppose we want to add another class B to this package. This can be done as follows.

Define the class and make it public

Place the package statement

```
Package p1;

//before the definition as follows
Public class B
{
   Body of B;
}
```

Store this as B java file under the directory P1
Compile B java file. This will create a B class and class B. to import all classes from the p1 package p1 will contain both the classes class A and class B import all classes from the p1 package then we have import the p1 package like :

Import p1.*;

Note :- Remember that, since a java source file can have one class declared as public, we can not put two or more public classes together in a **java** file. This is because of the restriction that the file name should be same as the name of the public class with **java** extension.

Interface :-

Normally java doesn't support multiple inheritances, but we can achieve by using the multiple inheritance, it means java can not have more than one super class.

```
class A extends B extends C
{
  This is not permitted in java
}
```

But java provides an alternate approach known as **interface** to support the concept of multiple inheritances. Although a java class cannot be subclass of more than one super class, it can implement more than one **interface**.

Definition of interface :-

An inheritance is basically a kind of class. It contains methods and variables. But the major differences is that interfaces define only abstract methods and final fields. This means that interfaces do not specify any code to implement those methods and data fields contain constants. Therefore, it is the responsibility of the class that implements an interface to define the code for implementation of these methods.

Syntax :-

```
interface interface name
{
Variable declaration;
Methods declaration;
}
```

Here, interface is a key word and interface name is an valid java variable. Declaration of variable is :

Static final type variable name = value

Methods declarations will contain only a list of methods without any body statements. Example :- return – type method name 1 (parameter list);

Example of an interface definition :

```
Interface item
{
Static final code = 11001;
```

Static final string name = "tube light";
Void display()
}

Note :- Code for the method is not included in the interface. An interface contains only abstract methods, which is incomplete methods, so it is not possible to create to an interface. Hence we create separate classes where can implement all the methods of the interface. These are called implementation classes. This implementation class contain all the bodies of the corresponding interfaces. Interfaces provide good flexibility, it means**, it provide necessary flexible services to other classes.**

Example :
Accessing database by different customers with different applications to connect in different ways. Interfaces are declared by using **interface** key in front of the interface name.

How to provide the implementation to the interface by the implementation a class
In java, we are using the i**mplements** keyword to implement the methods, which are declared in the corresponding interface
That is :
class< class name > implement <interface name >
{
Body of the class name
}

An interface is a specification of method prototypes. All the methods of the interface are public and abstract.
Interface methods are public since they should be available to third party vendors to provide implementation. They are abstract because their implementation is left third party.
An interface can have variables, which are public static and final by default. This means all the variables of the interface are constants.
None of methods in interface can be private, protected or static.
All the methods of interface should be implemented in its implementation classes .if any method is not implemented then ,implementation class should be declared as "abstract".
Interface reference can refer to the objects of its implementation classes.
When an interface in written, any third party vendor can provide implementation classes to it.
An interface can extend another interface.
An interface can not implement another interface.
A class can implements (not extends).multiple interfaces.

Abstract class vs interface:-

Abstract class	Interface
1.an abstract class is written when there are some common feature shared by all the objects.	1.an interface is written when all the features are implemented differently in different objects.
2.when an abstract class is written ,it is duty of the programme to provide sub classes to it.	2.an interface is written when the programme wants to leave the implementation to the third party vendors .
3.it contains some abstract method and some can create methods.	3.it contains only abstract methods.
4.it also contains instance variables	4.it contains only constants
5.all the abstract methods of the abstract class should be implemented is its sub classes.	5.all the methods of the interface should be implemented in its implementation classes.
6.it is declared by using the keyboard abstract	6.it is declared using the keyboard interface.

Defining sub classes :-

```
class subclass name extends superclass name
{
Variables declaration;
Methods declaration;
}
```

The key word extends signifies that the properties of the super class name are extended to the sub class name. the sub class will now contain its own variables and methods as well those of the super class name. this kind of situation occurs when we want to add some more properties to an existing class with out actually modifying it.

Short Answer Type Questions:-

1. Define package.
2. Create a package.
3. Accessing a package
4. What is an Interface?
5. What is Abstract class?
6. Define a sub- class?

Long Answers Type Questions :-

1. Explain the types of packages.
2. Explain to create own package with example.
3. How to provide implementation to the interface by the implementation a class?
4. What are the differences between abstract class and interface?
5. What are the advantages and disadvantages of interface?

UNIT 6

EXCEPTION HANDLING

Debugging

Errors in Java Program

Exceptions

How to handle the exceptions.

Types of exceptions.

6.5.Built in exceptions.

6.6.User defined exceptions.

6.7.Differences between error and exceptions.

Exception Handling

6.0 Debugging :-

A software engineer may also commit several errors while designing the project are developing the code .The errors are the wrong that can make a program go wrong. These errors are also called "bugs" and the process of removing them is called debugging.

Errors in java program: there are basically 3 types of errors in the java program.

Compile Errors
Runtime Errors
Logical Errors

Compile time errors:- These are systematical errors in the java program compilation. Nothing but writing a statement without proper syntaxes .these errors will be detected by the java compile and displays the list of errors with the line number only with their description.

Example:-

Missing semicolons.
Missing brackets in classes and methods.
Misspelling of identifiers and keywords.
Missing double quotes in strings.
Use of undeclared variables
Incompatible types in assignments/initialization
Bad references to objects
Use of =in place of == operators

Run time errors:-These run time errors represent insufficiency of the computer system to execute a particular statement. For example, insufficient memory to store something but these are not detected by the java compiler. They are detected by the JVM only at run time.

*Dividing an integer by zero.

*Accessing an element that is out of the bound of an array.

*Trying to store to value into an array of an in compatible class or type.

*Trying to cast of an instance of a class to one of it's sub class

*Passing a parameter that is not in a valid range or value for a method.

*Trying to illegally change the state of a thread.

*Attempting to use a negative size for an array.

*Converting invalid string to a number.

3. Logical errors:-These errors depict flows in the logic of the program it means programmer might use wrong formula for doing the task in the program. These errors are not detected either by java compiler or JVM. Only the programmer in the responsible to handling those errors.

Exceptions:-An exception is a runtime error. It means all exceptions occur only at runtime but some exceptions are detected at compile time and some at runtime .the exceptions that are checked at compilation time by the java compile are called "**checked exception**" while the exceptions that are checked by the JVM are called "un checked exceptions".

Unchecked exceptions, the programmer should handle them or through without handing . he con not simple ignore them ,as java compile will remind him something to avoid any harm caused by the rise of an exceptions . All exceptions are declosed as classes in java here classes also represent even errors. all these classes are descended flow a super class " throwable ".

Exception handling:- Exception occurs at run time in this case JVM displays exception details and then terminates the program abnormally. So the subsequent statements are not executed .this means the program is terminated abnormally .so it is compulsory to design the program in such a way that even if there is an exception ,all the clean up operations an performed and only then the program should be terminated .this is called "exception handling".

In java exception handling is done with the help of five statements , namely ,try ,catch ,finally throws and throw . when you have a solution for an exception then you use throw statement .in case of generating you own exceptions to use throw statements.

How to handle the exceptions:

We have to follow three steps. They are: Step-1:-the programmer should observe the statements in his programmer where there may be a possibility of exceptions .such statements should be written inside a **try** block so by using this, if any exception occurs then program will not be terminated and that exception in understand by the JVM and stores the exception details in an exception stock and then jumps into a catch block.

Syntax:-

try

{

Statements

}

Step-2: the programmer should write the "catch "block were it should display the exception details to the user .The try block will immediately follow this block

.it helps to the user about the error of the program . Here we can develop multiple catch blocks to a corresponding try block to handle different try block to handle different type of exceptions .But we have to remember only catch block will be executed among then base on the exception raised by the try block

. This means, even if there is scope of multiple exceptions . Only one exception at a time will occur.

Syntax:

catch(exception.obj)

{

Statements;

}

Step-3:the programmer should perform clap up operations like closing the files and terminating the threads . the programmer should write this code in the finally block . the statements in finally block are executed irrespective of whether there is an exception or not .so the program is terminated in the safe-mode.

Syntax:-

finally

{

Statements;

}

Performing above 3 types is called exception handling .

Note:- by using exceptions handling user con not prevent the exception . but the programmer is avoiding any damage that may happen to use data.

An exception can be handled using try, catch and finally blocks.

It is possible to handle multiple exceptions using multiple catch blocks.

Even though there is possibility for several exceptions in try block ,at a time only one exception will be raised.

Several catch blocks can follow a single try block .

We can not write catch blocks without a try block but we can write try block without catch block.

We can not write catch blocks without a try but we can write try block without catch block

It is not possible to insert some statements between try and catch block.

It is possible to write a try block with in another try . they are called nested try blocks.

Throws clause :- the programmer may or may not handle the runtime exceptions , but the rule is that the programmer should handle checked exceptions. In case the programmer does not want to handle ,the checked

exceptions , it should throw it out using the ”throws” clause . otherwise there will be an error will be flagged by java compiler .

Here the java compiler excepts the programmer to handle the exception using try and catch block ;

Else it should throw out the exceptions without handling it by using “throws” clause to the generally the throws clause is placed after the method header .

Example :-

Public static void main (String args [])throws IO Exception {}

This indicate that ,there is an IO exception will be occurring . if then this is exceptions is throwing to the outside person who is using this object. In this case. This exception is throwing to the JVM .

Throw clause:-

This statement is used to throw an exception explicitly from the try block to the corresponding catch block . simply by using throw class , we can throw the exception object to the corresponding catch block . so throw statements are commonly placed in the try blocks only . Example :-

Throw new IO Exception (“simple”);

This throw clause is used for.

Throw clause is used In software testing to test whether a program is handling all the exceptions as claimed by the programmer.

Throw clause can be used to throw our own exceptions also . this means throw clause can also throws user-defined exceptions .

Example:-

```
class sample
{
static void demo();
{
try
{
System. Out. println(“Inside demo());
Throw new null Point Exception(“exception data”);
}
```

```
catch (Null Pointer Exception ne);
{
System .out.println (ne);
}
}
}
class throwdemo
{
public static main(String args[])
{
Sample .demo();
}
}
C:\>javac throwdemo.java
C:\>java throwdemo
Inside Demo()
Java . lang nullPoint Exception Date.
```

Types of Exceptions :-

They are 2 types of exception are available in java .they are:

1.Built- in -exceptions

2.user defined exceptions

Built-in-exceptions:- these are exceptions ,which are already available in java .they are suitable to explain certain error situations.

The following are some built – in – exceptions :

Exception class	Meaning
Arithmetic Exception	Throw when an exceptional conditions has occurred in an arithmetic operations.
Array Index Out of Bound Exception	Thrown to indicate that an array has been accessed with an illegal index.
Class Not Found Exception	This exception is raised when we try to access a class whose definition is not found.

File Not Found Exception	Raised when a file is not accessible failed or interrupted .
IO Exception	Thrown when an I/O operation failed or interrupted.
Interrupted Exception	Thrown when a thread waiting, sleeping or doing some processing, and it is interrupted.
No Such File Exception	Thrown when a class does not contain the failed (or variable)specified.
No Such Method Exception	Thrown when accessing a method which is not found.
Null Pointer Exception	Raised when reffering to the member of a null object Null represents nothing.
Number Format Exception	Raised when a method could not convert a string into a numeric format
Run Time Exception	This represents an exception which occurs during run time.
String Index Out of Buffer Exception	Thrown by string class methods to indicates and index is either negative(-)or greater than size of the string.

User define exception:-

Sometimes, the built-in functions in java or not able to describe a certain situation .In such cases, the user can also create his own exception, which one called “user defined exceptions”. We have to fallow the following steps to create user define exceptions.

Step1:- The user should create his own exception class as a sub class to exception class.

Step2:- The user can write his own exception class. If the user does not want to create an empty to his exception class, he can eliminate writing to default constructor.

Step3:- The user can create a parameterized constructor with a string as a parameter .He can use this to store exception details he can call super class(exception) constructor from this and send the string there.

Example:-

```
class MyException extends Exception
{
MyException ( ) { }
MyException (String str)
{
super (str);
}
}
```

Step4:- When the user wants to raise his own exception he should create an object to his exception class and throw it using “throw” class as.

Example;-

```
My Exception obj =new My Exception (“sample Exception”);
```

Ex:-

User defined exception to throws when ever balance amount is below Rs.1000.

```
class MyException extends Exception
{
private static int accno[]={1001,1002,1003,1004,1005};
private static string name [](“ pavani”, ”sravani”, ”athreya”, “murthy “, “sailaja”);
private static double balance []={10.000,5000,3000,6000,8000};
MyException(){}
{
super(str)
}
}
class ex1
```

```
{
public static void main(String args[])
{
try
{
System.Out.Println(“A/cNo\tcustomer\tbalance”);
for(int i =0;i<5;i++)
{
System.out.println(accno[i]+”\t”+name[i]+bal[i]);
if(bal[i]<1000)
{
MyException me = new MyException (“Balance amount is less”);
throw me;
}
}
}
```

C:\>Javac Ex1.java

C:\>Java Ex1

Account no	customer	Balance
1001	pavani	10,000
1002	sravani	5,000
1003	athreya	3,000
1004	murthy	6,000
1005	sailaja	8,000

MyException: Balance amount is less

At MyException .main (My Exception .Java .40);

Difference between Error and Exception:

Exception is the super class of the all exceptions in Java .An exception is an error, which can be handled .it means when we exception happens, the programmer can do something to avoid any harm. But an error, which cannot be handled. It happened the programmer cannot do anything.

Short answer type questions

1. What is Debugging?
2. Write the Types of errors.
3. Define exception.
4. Write about throws clause.
5. What is throw clause.
6. write difference between error and exception
7. Write about Unchecked exceptions

Long answer type questions.

1. Explain the types of errors in java.
2. How to handle the exception in java.
3. Explain types of exception in java.
4. What are the built – in exceptions in Java.
5. Explain user define exceptions.
6. What are the differences between error and exceptions?

UNIT 7

Threads

Concepts of Threads

Introduction of threads

Multi threading

Multi Tasking

Creating of threads

Extending thread class

Stopping a thread

Blocking thread

The thread classes and the runnable interface

How to manipulating threads.

The life cycle of a thread

Differences between multi threading and multi tasking

Concepts of threads :-

Those who an familiar with the modern operating system such as window and window-7 may recognize that they can execute several programs simultaneously i.e editing a document in MS word and listening says from a system etc. This ability is known as multitasking. In system terminology it is called multithreading.
In modern operating systems they can execute several programs simultaneously, this facility is known as multitasking.
A thread is similar to program that has a single flow of control multithreading refers to two or more tasks executing concurrently with in a single program. A thread is an independent path of execution with in a program. Many threads can run concurrently with in a program. Every thread in java is created and controlled by the **java.long.thread.class**. A java program can have many threads and these threads can run concurrently, either asynchronously or synchronously.

Multithreading :

In the modern operating systems they can execute several programs simultaneously, this facility is known as multitasking. Multithreading is a conceptual programming model where a program is divided into two or more subprograms (process), which can be executed at the same time in parallel. For example, one subprogram can display a video on the screen at the same time as another play the audio.

A thread is similar to a program that has a single flow of control. Multithreading refers to two or more tasks executing concurrently within a single program. A thread is an independent path of execution with in a program. Many threads can run concurrently with in a program. Every thread in Java is created and controlled by the **java.lang.Thread Class.** A Java program can have many threads, and these threads can run concurrently, either asynchronously or synchronously.

Multi Tasking :

It is an operating system concept in which multiple tasks are performed simultaneously. It support execution of multiple programs simultaneously. The processor has to switch between different programs are processes. It is less efficient in comparison to multi threading. A program or process is the smallest unit in a multi tasking environment. It helps in development efficient operating system. It is expensive in case of context switching

Creating threads :-

Creating threads in java is simple. These are implemented in the form of objects that contain of the method called run(). The run method is the heart of the thread.thread. It keeps the total body of a thread.

```
public void run()
{
---
---
---
 Statements for implemented thread
---
---
}
```

The run () method should be invoked by an object of the related thread. This can be achieved by creating the thread and initiating it with the help of another thread method called start (). A new thread can be created in two ways :

By creating a thread class
By converting a class to a thread

By creating a thread class :-

Define a class that extends thread class and override its run() method with the code required by the thread.

2) By converting a class to a thread :-

Define a class implemented runnable interface. The raunnable interface has only one method, run(), that is to be defined with the code to be executed by the thread.

Extending the thread class :-

We can make our class as thread by extending the class java.long.thread. It includes the following steps

declare the class as extending thread

```
{
-----
-----
-----
}
```

Now we have a new type of thread mythread.

Implement the run() method that is responsible for executing the sequence of code that the thread will execute.

```
public void run()
{
----
----
----
}
```

When we start the new thread, java calls the threads run() method.

create a thread object and call the start() method to initiate the thread execution. Mythread T1= new mythread();
T1. Start ();

Example :-

```
class A extends thread
{
public void run()
{
for (int i=1; i<=5; ++i)
{
System.out.println("threads A: i="+i);
}
}
}
class B extends thread
{
public void run()
{
for (int j=1; j<=5;++j)
{
System.out.println("thread B:j="+j)
}
}
}
class c extends thread
{
public void run()
{
for (int k=1; k<=5;++k)
{
System.out.println("thread c:k="+k)
}
}
```

```
}
class thread test
{
public static void main (string args[])
{
new A().start();
new B().start();
new C().start();
}
}
```

Stopping thread:-

When ever we want to stop thread from running, we use it stop() method,

like T1. Stop();

This statement cause the thread to move to the dead state.

Blocking a thread :-

A thread can also be temporarily suspended or blocked by using following methods. Sleep () // blocked for a specified time
Suspend () // blocked until further order
Wait() // blocked until certain condition occur.
These methods cause the thread to go in to the **not – runnable** state. The thread will return to the runnable state when the specified time is elapsed in the case of **sleep(),** the **resume()** method is invoked in the case of **suspend()**and the **notify()** method is called in the case of **wait()**

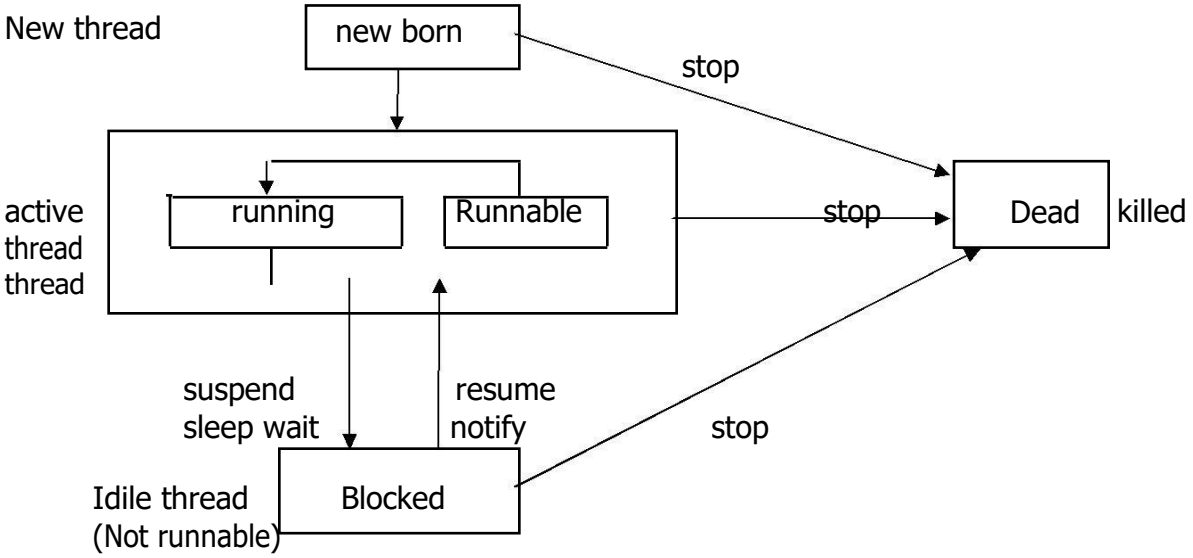
The thread classes and the runnable interface:-

Method	Description
Get name ()	Returns the name of the thread
Get priority ()	Returns the priority of the thread
Isalive	Determains whether the thread is running
Run()	The entry point the thread suspends a thread. This method you to specify the period.
Sleep()	Suspends a thread. His method you to specify the period
Suspend()	The thread is suspended
Yield()	Cause the currently running thread to yield to any other threads of the same priority that are waiting to be scheduled.
Start()	Starts the thread

The life cycle of the thread :-

During the life time of a thread, a thread is always is one of five states. It can move from one state to another

New born state
Runnable state
Running state
Blocked state
Dead state



1) The new born state :-

When a thread's called it is in the newborn state, that is, when it has been crated and is not yet running. In other words a start() method has not been invoked on this thread. In this state system resources are not yet allocated to the thread. When a thread is in new born state, calling any method other than starts method cases Illegal Thread State Exception.

1. The runnable state :-

A thread is the runnable state is ready for execution but is not being executed currently. Once a thread is in the runnable state, it gets all the resources if the system and moves on to the running state. All runnable threads are in a queue and wait for CPU access. When the start() method is called on the new born thread, it will be runnable state.

2. Running state :-

After the runnable state, if the thread gets CPU acces, it moves in to the runnable state. The thread will be the running state unless one of the following things occur.

It dies
It gets blocked to the input/output
It calls sleep()
It calls wait()
It calls yield()

3. The blocked state :-

A thread can enter the blocked state when one of the following 5 conditions occurs.

When sleep() is called
When suspend () is called
When wait is called
The thread is waiting for monitor

The dead state :-

A thread goes into the dead state in 2 ways

If it run() method exits. When it finished execution
A stop() method is invoked. Naturally an uncaught exception. The stop() method kills the thread. A thread in the dead state can not be executed further.

Manipulating threads :-

In addition to the methods there are some which need to be includes they are sleep(), suspend(), resume(), wait(), notify(), notify all() and yield () methods.

Sleep():-

A thread being executed can invoke sleep() to block the thread for some time and the to CPU.

Suspend() :-

The thread class has a method suspend() to stop the thread temporarily and a method resume() to restart it at the point at which it is halfed. The resume() method must be called by some thread other than the suspend one.

Wait(), notify() and notify all() :-

When a running thread calls waits(), it enters into a waiting state for the particular object on which wait was called. A thread in the waiting state for the particular object on which wait was called. A thread in the waiting state for an object becomes ready on a call to notify() assued by another thread associated with that object.

Yields() :-

Some CPU intensive operations of one thread may prevent other threads from being executed. To prevent this from happening, first one can allow other threads to execute by invoking the yield() method.

Difference between multi threading and multi tasking.

Multi threading	Multi tasking
It is a programming concept which a program or a process is divided in to two or more sub programs or threads that are executed at the same time in parallel.	It is an operating system concept in which multiple tasks are performed simultaneously.
It support execution of multiple parts of a single program simultaneously	It support execution of multiple programs simultaneously
The processor has to switch between different parts or threads of a program	The processor has to switch between different programs are processes.
It is highly efficient	It is less efficient in comparison to multi threading
A thread is a small unit it multi threading	A program or process is the smallest unit in a multi tasking environment.
It helps in developing efficient programs.	It helps in development efficient operating system.
It is cost – effective in case of context switching	It is expensive in case of context switching

Short answer type Questions :-

1. What is a thread?
2. What is multitasking?
3. What is multithreading?
4. Write about Stopping a thread
5. Write about Blocking a thread

Long answer type questions**:-**

1. Explain creating a thread with example.
2. Explain extending thread with example.
3. Explain the life cycle of the thread.
4. What are the differences between multithreading and multitasking?
5. How to manipulating the threads?

UNIT 8

THE JAVA APPLET

8.0 Introduction to java applets

8.1. Creating Applet.

8.2 Basic methods of the applet class

Designing the Web Page.

Attributes in Java

A Java Program create an Applet with some back ground color and

fore ground color.

8.6 Passing Parameters to Applet.

More about a HTML tags.

Getting the input from the user with example.

8.0. Introduction to Applets:

When a HTML page wants to communicate with the user on internet it can use a special java program, called applet to communicate and respond to the user. The user can interact by typing some details or by checking to results. The main() methods is not available in case of applets.

We can understand an applet as a java byte code embedded in a HTML page, generally for the purpose of achieving communication with the user.

Applet = java byte code + HTML page

8.1. Creating applet :-

To create an applet we need to write a java program and compile in to get byte code . Then we should embed (include) it into HTML page on a particular location when we want it to be displayed. This page is then sends the HTML page contains the applet. The page is then transmitted to the client.

To create an applet, we have applet class of java. Applet package and J applet class of java x. saving package. These classes use the following methods, when are automatically run by any applet program.

Public void int ():

This method is the first method to be called by the browser and it is executed only one. So, the programmer can use this method to initialize any variables creating components and creating threads etc.

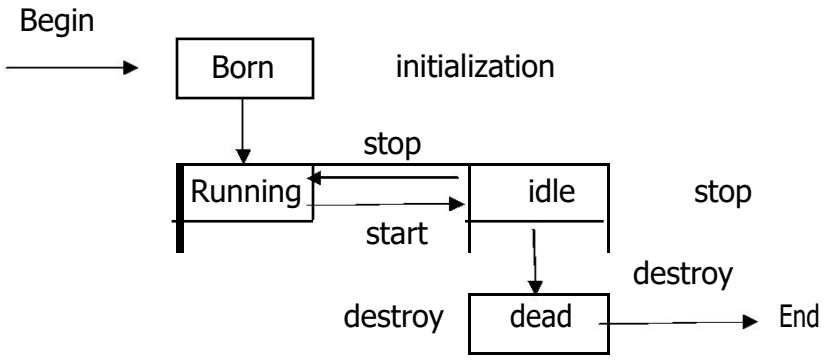
Public void start () :-

This method is called aften int()method and each time the applet is refresh by the user. For example, the user has minimized the web page. That contains the applet and moved to another page then this method's execution is stopped. When the user comes back to view the web page again start() method execution will resume.

Public void stop() :-

This method is called by the browser when the applet is to be stopped public void destroy(). This method is called when the applet is being terminated from memory. An applet is a born with int () method and to terminate the applet completely from memory to destroy() method is called. Once to the applet terminated, we should to executing the methods is called applet life cycle. Java inherits a set of default behaviors from the applet class. As a result, when an applet is loaded, made a series of changes in its state.

Basic Methods of Applet Class :



Exit of the bowser

The applet states include :-

Born or initialization state
Running state
Idle state
Deed or destroyed state

Initialization state :-

An applet enters the initialization state it is first loaded. This is achieved by calling the int () method of applet class. The applet is born. At this stage, we may do the following it required.

Create objects needed by the applet
Setup initial values
Load images of fonts
Setup colors

The initialization occurs only once in the applet life cycle. To provide any of the behaviors mentioned above, must over rode to main() method. Public void int ()

```
{
    ---
    ---
    ---
}
```

Running state :-

Applet enters to running state when the system calls the start() method of applet class. This occur automatically after to applet is initialized .starting can also occur if the applet is alredy in "stopped state".

```
Public void start ()
{
}
```

3)**Stopped state:-**

An applet becomes idle when it is stopped from running ,stopping occurs automatically .when we leave the page containing to currently running applet . we can also do so by calling to stop () method terminate the thread .we can achieve this by overloading the stop () method .

```
Public void stop ()
{
}
```

Destroy state :-

An applet is said to dead when it is removed from memory . this occurs automatically by invoking the destroy () method when we quit by browser .like initialization ,destroying state occurs only once in the applet's life cycle. If the applet has created any resources, like methods we may override the destroy (). Public void destroy () {

```
}
```

Paint state:-

Applet moves to the display after whenever it has to perform, some output operations on the screen. This happens immediately after the applet enters in to the running state. The paint method is called to accomplish this task. Public void paint (Graphics.g)

```
{
      ---
      ---
      ---
      ---
}
```

Designing a web page :-

A web page is basically made up of text and HTML tags can be understand by a web browser or an applet viewer. Like java source code, it can be prepared using any ASCII text editor. A web page is also known as HTML page or HTML document. We pages are stored using a file extension**".html"** such as name.html. HTML files should be stored in the some directory as the compiled code of the applets.
Web pages contains both text as well as HTML tags. A web page is marked by an opening HTML tags <HTML> and a closing HTML tag </HTML> and is divided in to the following three major sections.

Comment section (it purly option)
Head section (it is also option)
Body section

A web page outline containing these three sections and opening and closing HTML tags.

Example :-

```
<HTML>
<!, - - - - - -
- - -
- - -
- - -
>
<HEAD>
Title tag
</HEAD>
<BODY>
Applet tag
</BODY>
</HTML>
```

Comment section :-

This section contains about the web page. It is important to include comments that tells us what is going on in the web page. A comment line begins with a <!: and ends with a Web browsers will ignore to text enclosed between them. Note that comments are optional and can be included any where in the web page.

Head section :-

The head section is defined with starting <HEAD> tag and a closing </HEAD> tag. This section usually contains a title for the web page as shown below.

<head>
<title > Gjc, Chirala </title>
</head>

3) Body section :-

After the head section comes the body section. We call as body section because this section contains information about the web page and its behavior. We can set up many options to indicate how our page must appear on the screen like color, location, sound etc.,

<BODY>
<CENTER>
<HI> <Govt. Jr. College chirala/ Hi>
</CENTER>
</BODY>

<APPLET> tag

The body section of the web page contains <APPLET> tag is useful to include an applet into an HTML page. This tag supplies the name of the applet to be 10 added and tells the browser how much space the applet requires. It has to following form
<APPLET CODE = "name of the applet class file" CODE BASE= "path of the applet class file"
HEIGHT = maximum height of applet in pixels
WIDTH = maximum width of applet in pixels
ALIGN = alignment (LEFT,RIGHT, TOP, BOTTOM, MIDDLE)
ALT =alternate text to be display
NAME =applet = instance_name
VSPACE = pixels HSPACE = PIXEL
<PARAM NAME= parameter name 1 VALUE = its value> <PARAM NAME = parameter name 2 VALUE = its value> </APPLET>

The <PARAM> tag is useful to define a variable <parameter> and its value inside the HTML page which can be passed to the applet. The applet can access the parameter value using get parameter method() method as :
String value = get parameter("parameter name");

Attributes in Java.

Attribute	Meaning
CODE = Applet file name class	It specifies the name of the applet to be loaded that is, the name already compiled. Class file in which to executable java byte code for the applet is stored.
CODE BASE = code base – URL	It specifies the URL of the directory in which the applet is stored (optional) if the .html and .class files are takes place in the same directory, this attribute may be omitted.
HEIGHT = pixels WIDTH= pixels	These attributes specify the width and height of the space on the HTML page.
ALLIGN= alignment	This optional attribute specifies where on the page the applet will appear such as top, bottom, left, right, middle
ALT = alternate-text	Non- java browsers will display this text.
NAME = applet-instance – name	A name of the applet may optinally be specified so that other applets on the page may refer to this page
VSPACE = pixels	It is used only when some vertical alignment is specified with the ALIGN attribute. This specifies the amount of vertical blank space to browser should leave surrounding to applet.
MSPACE=pixels	It is used only ALLIGN is set to LEFT or RIGHT. This specifies the amount of horizontal blank space.

Example :

Writing code to meet simple applet

<Applet. CODE= "my app. Class" HEIGHT= 300 WIDTH = 400>

This HTML code tells to browser to load the compiled java applet myapp. Class, which is the same directory as the HTML file and also specifies to display area for the applet output as 300 pixels height and 400 pixels width. The following step, involved in developing and testing in applet are

```
import java. awt.*
import java. applet.*
public class myapp Extends Applet
{
public void in it ()
{
Setbackground (color, yellow);
public void main (graphics g)
{
g. drawString ("Govt Jr Colleg" 50, 100);
}
}
```

C:\> javac myapp java

Now, myapp. Class is created. This byte code should be embedded in to a HTML page using

<APPLET> tag is

```
<!myapp . html that embeds myapp applet>
<HTML>
<APPLET CODE = "myapp.class" HEIGHT = 300 WIDTH =
400> </APPLET>
</HTML>
```

Save the above code with name myapps.html. This HTML page contains applet which can be opened in the browser, or an applet viewer supplied by he sun micro system. Inc, it can be used to test the applet, for this purpose, open any browser and in the browser's address bar

type . html file name along with directory path. The applet opens in the browser or give the command at system prompt as

C:/> applet viewer myapps.html

Write a java program create an applet with some back ground color and foreground color with message

```
import. Java. awt.*
import .java.applet.*;
public class app1 extends applet
{
string msg = "   ";
public void int ()
{
```

```
setbackground (color. Yellow);
setforeground (color. Red);
Font f = new Font ("Arial", font. BOLD, 20);
setfont (f);
msg"="init";
}
public void start()
{
msg += "start";
}
public void point (Graphics g)
{
g. draw string (msg, 50,100);
}
public void stop();
{
msg+="stop";
}
public void destroy()
{
msg += "destroy";
}
}
```

C:\>javac app1. Java

Now, appl. Class is created. This byte code should be embedded into a HTML page using <APPLET> tag, as

```
<!APP1. html that embeds myapp applet>
<HTML>
<APPLET CODE = "Appl. Class" HEIGHT = 300
WIDTH= 400> </APPLETT>
</HTML> open the above appl. Html file in the applet viewer as c:/> applet viewer app1.html
C:\> applet viewer app1.html
}
```

Passing parameters to applets:-

Along with applet tag, we can use <PARAM> tag to pass information to applet. For example to pass name and salary of an employee to an applet from HTML page, we can write the applet and param tags as :

```
<APPLET CODE = 'TAX CLASS" WIDTH = 300 HEIGHT = 300>
```

<PARAM NAME = "E1" VALUE = "MURTHY">
<PARAM EMPLOYEE = "E2" VALUE = "JL IN
CS&E"> </APPLET>
<PARAM> tag has two attributes "NAME" and VALUE NAME represents the name of the parameter. E1 and E2 and the review VALUE are "MURTHY" and "JL in CS&E" respectively.

To receive the values of the parameters an applet uses get parameter () method. This method takes to parameters name and return its value as a string.

```
Name = get parameter ("E1");
Str = get parameter ("E2");
```

Where name and str are string type variable set parameter () method should be used inside the init () method of the applet.
Passing parameters to an applet code using <PARAM> tag is something similar to passing parameters to an applet code using <PARAM> tag is something similar to passing parameters to the main () method using command line arguments.

Ex:-

```
import java.awt.*
import java.applet.*
public class parademo extends applet
{
String str;
public void init()
{
str= get parameter ("color")
if (str. Equals("red")
setbackground (color. Red);
else if (str. Equals ("blue")
setbackground (color. Blue);
else
setbackground (color. pink);
}
public void paint (graphics g)
{
g. draw string (str, 100, 100);
}
}
javac parademo. Java
```

<! Create a app.html file>
<HTML>
<Applet code = "paramdemo".class" Height = 300 width
= 400> <param.name = "color VALUE="green>

</APPLET>
</HTML>
Applet viewer app.html
Now change the <PARAM> tag VALUE attribute values and run the applet. That will give a different color appearance.

More about HTML tags :-

HTML support a low number of tags that can be used to control the style and format of the display of web pages.

TAG	FUNCTION
<HTML> </HTML>	Signifies the beginning and end of a HTML file
<HEAD. </HEAD>	This tag may include about the web page ,usually tag with in it
<TITLE> </TITLE>	The text contained in it will appear in the title bar of the browser.
<BODY> </BODY>	This tag contains the main text of the web page. It is the place where the <APPLET> tag is declared.
<H1> … </H1>	Header tags used display
<H6>….</H6>	Headings <H1> creates the largest font header
<CENTER> … </CENTER>	Places the text contain in it at the centre of the page
<APPLET>…</APPLET>	It declares the applet details as it attributes.
<PARAM>	This tag is placed inside the <APPLET> tag and it hold the user defined parameter.
<B>...</B>	Text between these tags will be displayed in bold type.

<U>…</U>	Text between these tags will be displayed in the underline style.
<I>…</I>	Text between these tags will be displayed in italic type.
 	Like break tag these will skip a line. Does not have an end tag.
<P>…</P>	Paragraph tag.
</MG>	This tag declares attributes of an image displayed.
<HR>	Draws horizontal line.
<A>…</A>	Anchor tag used to add hyperlinks.
<FONT>…</FONT>	We can change the colour and size of the text that lines in between these tags using attributes.

PROGRAM:- a program for example animated import java.awt. *;
import java. applet.*;
public class animate extends applet
{
public void paint(graphics g)
{
Image img=get Image (get document base(),"lab gif);
for (int x=0;x<800;++x)
{
g.drawImage (img,x,50,null); try
{
Thread. Sleep(20)
}
catch (Interrupted Execution){} Java animate.java
<html>
<Applet code = "animate.class" height = 600 width =
800> </Applet>
</Html>
Applet viewer app.html
}
}
}

Getting input from the user:-

Abstract window toolkit (AWT) represents a class library to develop applications using GUI. The java.awt package got classes and interfaces to develop GUI and let the user interact in a more friendly way with the application.

A component is a graphical representation of an object on the screen. For example push buttons, radio buttons, menus buttons etc are components objects are used to accept the user input.

Example:- A program to take two numbers flow user and calculate sum.

```
import java. awt.*;
import java. applet.*;
import java. awt. Event.*;
public class user input extends applet implement ActionListener {
TextField N1, N2;
public void int ();
{
N1= new TextField (10);
N2= new TextField (10);
add (N1);
add (N2);
N1. Add ActionListener (this);
N2. Add ActionListener (this);
N1. setText (“0”);
N2. setText (“0”);
}
public void paint (Graphics g)
{
int x=0, y=0, z=0;
String s1, s2, s3;
g. drawstring (“Input a number in each box”15.40); try
{
s1=N1.getText ();
x= Integer.parseInt (s1);
s2=N2.get Text ();
y=Integer.parseInt (s2);
}
catch (Exception ex){} Z=x+y;
s3= String. valueof (z);
```

```
g.drawString ("the sum is :"10,20");
g.drawString (s3,15,25);
}
public void action performed (action event e)
{
repaint ();
}
}
>javac user input. Java
<HTML>
<APPLET CODE="userinput. Class" HEIGHT =200 WIDTH=300>
</APPLET>
</HTML>
>applet viewer app.html.
```

Short answer type questions

1. What is applet?
2. How can we Create an applet?
3. Write about Applet initialization and termination?
4. Write any 4 HTML applet tags?

Long answer type questions

1. Explain applet life cycle?
2. Explain the HTML applet tags in java?
3. Explain writing code and creating simple applet?
4. Write the procedure to passing parameters to applet?
5. Write about the attributes in java?
6. Write a java program to create applet with some background color and foreground color with message.
7. Write about How to passing parameters to applet.
8. Write about the html tags in java.

.

----"HARD WORK IS SECRETE OF SUCCESS"----

COMPUTER SCIENCE & ENGINEERING

Paper – II

Relational Database Management System

INDEX

UNIT 1

CONCEPT OF DBMS

Structure

Introduction
Purpose Of Data Base Systems
Data Abstraction
Data Models
Instances, Schemes
Data Independence
Data Integrity
DDL, DML, DCL
Responsibilities Of Database Manager
Data Base Administrator
Data Base Users

Learning Objectives:

Database System.
Comparison between Traditional File Processing System and DBMS. Data Abstraction, Instance, Schema and Sub – schema .
Understanding data models, Database Languages and Database Administrator. Understanding the functions and responsibilities of Database Manager.
Understanding the functions and responsibilities of Database Administrator

1.1 Introduction

A database can be defined as a collection of inter related relevant data stored together to serve multiple applications.

A DBMS is a collection of inter related data and a set of programs to access those data. The collection of data usually called as database. It contains the data or information about one particular enterprise or an organization.

A Database Management System (DBMS) can be defined as a collection of programs or software packages those are used to define, manipulate, control and processing the database.

A DBMS is essentially a tool to build a database structure, store and operate on the data contained in it. The main objective of designing any database management system is to provide its users an appreciably higher level frame work (as compared to files access facilitates provided in high-level programming languages) such that the relevant interrelated data can be collected and stored together to serve multiple application. A DBMS creates and maintains a database, loads and manipulates data within a database.

PURPOSE OF DATA BASE SYSTEMS

One way to keep the information on a computer is to store it in permanent system files. To allow users to manipulate the stored information, the system has a number of application programs that manipulate the organized files..

*A program to debit or credit an account
*A program to add a new account
* A program to find balance of an account

These application programs are written by system programmers in response to the needs of the typical bank organizations using *file_processing system*. New application programs are added to the system as the need arises. Thus, as the time goes more files and more application programs are added to the system. A typical file processing system described above is the system used to store information before the advent of DBMS.

Keeping the information of an organization in a file processing system has a number of disadvantages, namely

Data Redundancy and Inconsistency : Since the files and applications programs are created by different programmers over a long period, the various files are likely to have different formats and the programs may be written in several programming languages. Moreover, the same information may be duplicated in several places. This redundancy leads too higher storage and access cost. In addition, it may lead to data inconsistency.

Difficulty in Accessing Data : Another important problem in traditional file environment system is difficulty of getting quick results because it needs more adhoc queries and more programming for new reports. So, the file processing system does not allow needed data to be retrieved in a convenient and efficient manner.

Data Isolation :I a file processing system, as the data are scattered in various files, and files may be in different formats. It is very difficult to write new application programs to retrieve the appropriate data.

Integrity problems : The data values stored in the database must satisfy certain types of consistency Constraints (Conditions). For example, the minimum balance in a bank account may never fall below an amount of Rs. 500.Developers enforce these constraints in the system by adding appropriate code in the application programs. However, when new constraints are added, it is difficult to change the application programs to enforce them.

Security problems : Not every user of the database system should be able to access all the data. Even the simplest data-retrieval task requires extensive programming in the file processing systems and as these application programs are added to the system in an adhoc manner, it is difficult to enforce security.

Concurrent_Access Anomolies: If multiple users are updating the same data simultaneously it will result in inconsistent data state. In file processing system it is very difficult to handle this using program code. This results in concurrent access anomalies.

To overcome the problems in keeping the data in Traditional File Processing System, the DBMS was introduced. Keeping the information of an organization in a Database Management system has a number of advantages, namely

The data base approach offers a number of potential advantages compared to traditional file processing system. The primary advantages are:

Program data independence
Minimal data Redundancy
Improved data consistency
Improved data sharing
Increased productivity of application development.
Enforcement of standards
Improved data Quality
Improved data accessibility and responsibility.
Reduced program maintenance.

Program data Independence: The separation of data description of metadata from the application programs that use for data is called data independence.

Minimal data redundancy: The data base approach does not eliminate redundancy entity, but it allows designers to carefully control the amount of redundancy.

Improved data consistency: By eliminating data redundancy we can greatly reduce the for data inconsistency.

Improved data sharing: The data is designed as a shared corporate resource authorized internal and external users and granted the permission to use data base and each user is provided one or more user view to facilitate this use.

Increased productivity of application development: The major advantages of the data base approach are that greatly reduces the coast and time for developing new business application.

Enforcement standards: These standards will include naming convention, data quality standards, a number of uniform processors for accessing, updating protecting the data.

Improved data quality: The database approach provides a number of tools and processes to improve data quality. Two of the more important are constraints and cleanup.

Improved data accessibility and responsibility: With relational database the users can experience without programming knowledge to retrieve and display the data using SQL.

Reduced Program Maintenance: Stored data must be changed frequently for a variety of reasons new data item types are added, data formats are changed and so on.

Simply DBMS provides the following advantages over File Processing System.

Provides for mass storage of relevant data.
Make easy access of the data to user.
Allows for the modification of data in a consistent manner.
Allows multiple users to be active at a time
Eliminate or reduce the redundant data.
Provide prompt response to the users request for data.
Supports Backup and recovery of data.
Protect data from physical hardware failure and unauthorized access.
Constraints can be set to database to maintain data integrity.

DATA ABSTRACTION

ADBMS is a collection of interrelated files and set of programs which allows the users to access and modify these files. A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

Levels of Abstraction: basically, Abstraction can be divided in to 3 levels. **Those are**

Physical Level : The lowest level of data abstraction describes how the data are actually stored in the database. At the physical level, complex low-level data structures are described in detail.

Logical Level (Conceptual Level) : This is the next higher level of abstraction. It describes what data are stored in the database, and what relationship exist among those data. This level of abstraction is used by Database Administrator (DBA), Who decides what information is to be kept in the database.

View Level: This is the Highest level of data abstraction. It describes only a part of the entire database. The use of simpler structures at the logical level, some complexity remains, because of the large databases. Many users of the database system will not be concerned with all this information. Such users need to access only a Part of the database. So that their interaction with the system is simplified, the view level of abstraction is defined. The system may provide views for the same database.

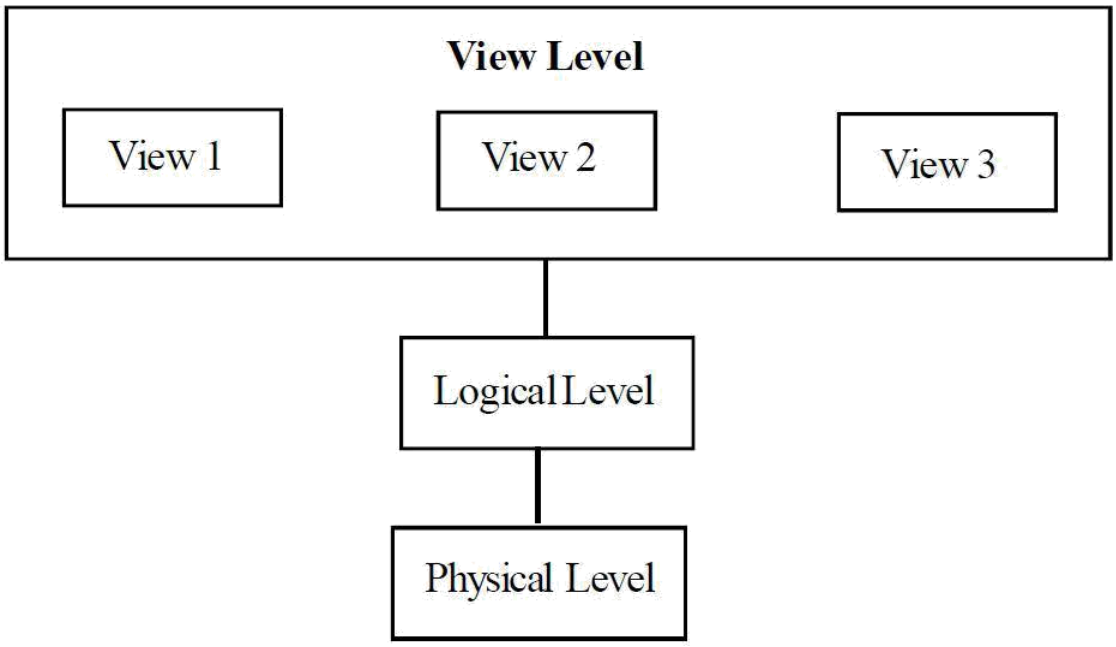


Fig 1.3

DATA MODELS

Simple meaning of Data modeling is describing Data.So we need Data models to describe dataand itplays an important role in database design. The physical or logical structure of a database is spelt out by the data model. A data model is a collection of conceptual tools used for describing data, data relationships, data semantics and data constraints.

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.

Data Model is a logical structure of Database. It describes the design of database to reflect entities, attributes, relationship among data, constrains etc.

The very first data model could be flat data-models, where all the data used are to be kept in the same plane. Earlier data models were not so scientific, hence they were prone to introduce lots of duplication and update anomalies.

In order to depict the overall organization of a database, a data model is used. The model should enable the designer to incorporated a major portion of semantics of the database in the schema. The data models have been broadly classified in to :

Object based data models
Record – based data models
Physical data models.

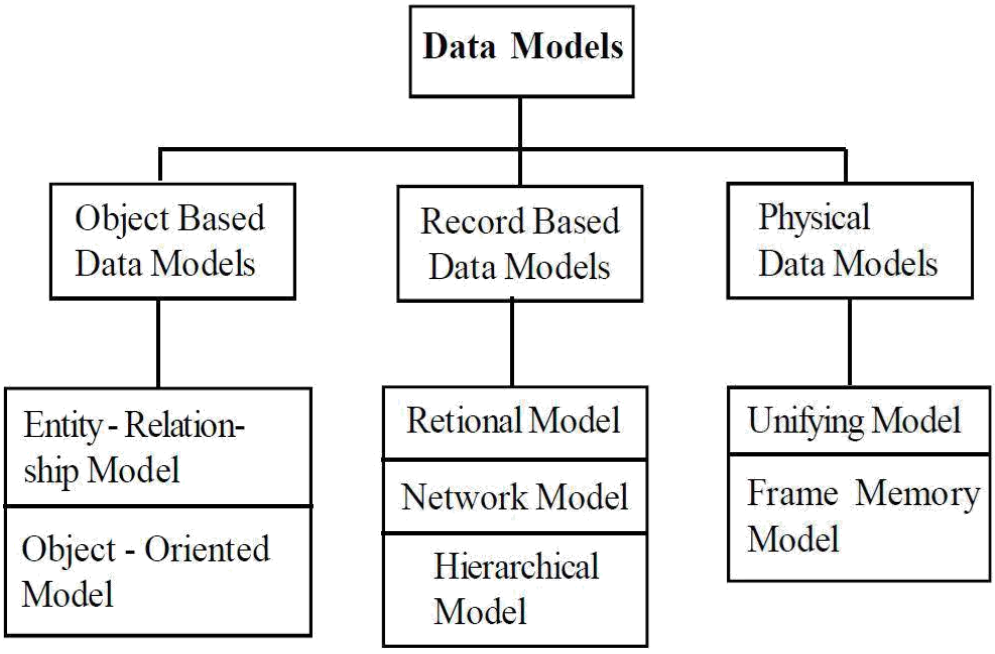


Fig 1.4

Object based data models

Object – based data models are used in describing data and data – relationships in accordance with concept. In general, the object – based models are gaining wide acceptance for their flexible structure capabilities Various data integrity constraints can be specified explicitly by using the object – based models. A number of object – based models have been proposed later.

The Entity – Relationship model which is an object – based model is widely used in the practice as an appropriate database design The entity-relationship model and the Object Oriented model as representative of the class of Object-based logical models.

The Entity-relationship Model : This data model is based on a perception of real world that consists of a collection of basic objects, Entities, Entity sets, relationship and relationships sets. The overall logical structure of a database can be expressed graphically by an E-R-Diagram, which is made up of components. Some of they are

Rectangles : Which represent entity sets.
Ellipses : Which represent attributes
Diamonds : Which represent relationship sets
Lines : Which link attributes to entity sets and entity sets to relationship sets.

The Object-Oriented Model : This model is based on a collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called Methods**.**

Objects that contain the same types of values and the same methods are grouped together into classes**.** A class may be viewed as a type definition for objects. This combination of data and methods combining a type definition is similar to a programming language (OOPS) abstract data type .

The only way in which one object can access the data of another object is by invoking a method of that other object. This action is called sending a message to the object.

For Example, let us consider an object representing a Bank account. Such an object contains instance variables 'account-number' and 'Balance'. It contains a method 'pay-interest', which adds interest to the balance.

Record – Based data Models

These models are used to specify the overall logical structure of the database. With some models a higher level description of the implementation of the structure of the database can also be specified explicitly. The data integrity constraints cannot be specified with these models. The three widely accepted record – based data models are

- Relational model
- Hierarchical model.
- Network model

In record based data models, the database is structured in fixed formats records of several types. Each record defines fixed number of fields (attributes) and each field is fixed length. These models are used to specify the overall logical structure of the database and are used in describing the database at conceptual level.

Relational model:

The relational model represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

Relational data has usually consists many relations, that are related in various ways. A Relational Data Base schema S is a set of relational schemas S = { R1, R2, ……, Rm } and a set of integrity constraints i.e. A relational data base state DB of S is a set of relation states DB = { r 1, r 2, r 3, ….. ,r m}such that " r i " is a state of " R i " and such that the " r i " relation states satisfy the integrity constraints specified in IC.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as

tables. However, the physical storage of the data is independent of the way the data are logically organized.

Relational model uses a collection of tables to represent both data and the relational ships among those data. Each table has multiple columns and each column has a unique name.

Relational Model Concepts

Attribute: Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, NAME,etc.

Tables – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.

Tuple – It is nothing but a single row of a table, which contains a single record.

Relation Schema: A relation schema represents the name of the relation with its attributes.

Degree: The total number of attributes which in the relation is called the degree of the relation.

Cardinality: Total number of rows present in the Table.

Column: The column represents the set of values for a specific attribute.

Relation instance – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.

Relation key - Every row has one, two or multiple attributes, which is called relation key.

Attribute domain – Every attribute has some pre-defined value and scope which is known as attribute domain

Advantages of using Relational model

Simplicity: A relational data model is simpler than the hierarchical and network model.

Structural Independence: The relational database is only concerned with data and not with a structure. This can improve the performance of the model.

Easy to use: The relational model is easy as tables consisting of rows and columns is quite natural and simple to understand

Query capability: It makes possible for a high-level query language like SQL to avoid complex database navigation.

Data independence: The structure of a database can be changed without having to change any application.

Scalable: Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.

Disadvantages of using Relational model

Few relational databases have limits on field lengths which can't be exceeded.

Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.

Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.

Hierarchical Data Model:

This model is introduced in the information management system developed by the IBM in 1968. This relates records by the **Parent / Child** or **Supervisor / Sub-ordinate** relationship. This model like a hierarchical tree structure which is used to contract a hierarchy of records in the form of branches and nodes as the model its name.

Information which is closely related in a Parent – Child structure is stored together n the form of logical unit. Here a parent unit may have many child units. But a child unit can have only one parent unit. This model a permit to basic types of relationship namely **One – to – One** and**One – to – Many** relationships. The relation is Irreflexive , Anti symmetric and Transitive.

This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and many students.

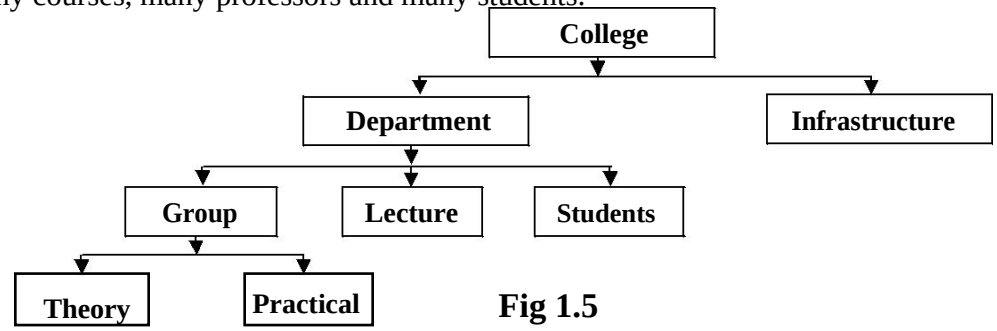


Fig 1.5

Advantages

The hierarchical model, allows one-to-one-and one-to-many relationships.
The model has got the ability to handle large amount of data.

Disadvantages

The model involves with complicated querying.
As duplication of data takes place, there is wastage of storage space.
During updating of data inconsistency exists.
The model does not allow many-to-many relationships.

Network model:

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

It is quite similar to hierarchical data model. But it has slite improvement here we can have multiple parent – child relationships i.e. many – to – many relationships can be represent when designing this the model one has to been establish relation between records at the time of creation of database. This model helps in rapid any easy access to data has we have multiple access paths to the records.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast.

This was the most widely used database model, before Relational Model was introduced.

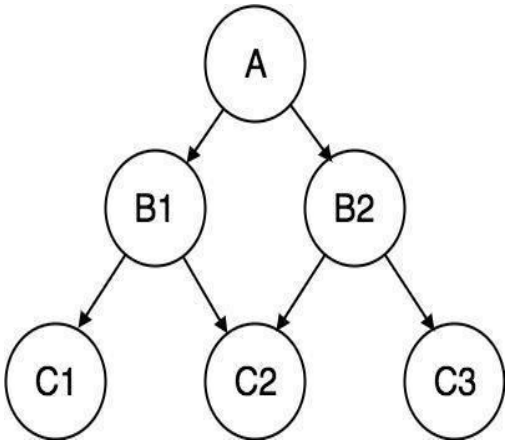


Fig 1.6

Advantages

It can be used to represent many-to-many relationships
It offers integration of data
The storage space is reduced considerably due to less redundancy
It provides faster access of data.

1.4.3 Physical Data models

A physical data model defines all of the logical database components and services that are required to build a database or can be the layout of an existing database.

A physical data model consists of the table's structure, column names and values, foreign and primary keys and the relationships among the tables.

These models are used to have higher level description of the storage structure of the database and their access mechanism. With the physical models it is possible to implement the database at the system level. A very few physical data models have been proposed so far. TWO of these well known models are the unifying model and the frame memory model.

INSTANCES, SCHEMAS

A **database instance** is a set of memory structures that manage **database** files.
A **database** is a set of physical files on disk created by the CREATE **DATABASE** statement.
The **instance** manages it's associated data and serves the users of the **database**.

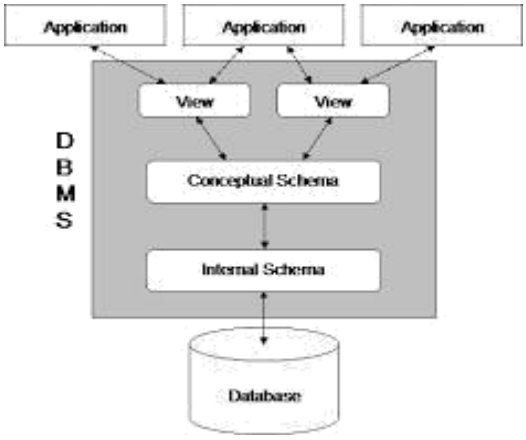
Database change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an 'Instance' of the database.

The overall design of the database is called the database 'schema'. Schemas are changed frequently.

A database scheme corresponds to the programming language type definition. A variable of a given type has a particular value at a given instant, Thus, the value of a variable in programming languages corresponds to an instance of a database schema.

Database systems have several schemas, portioned according to the level of abstraction. At the lowest level is Physical scheme, at the intermediate level is logical schema and at the higher level is a subschema.

In general, database system supports one physical schema, one logical schema and several subschema's.



DATA INDEPENDENCE

The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called data Independence. There are two levels of data independence.

They are

Physical Data Independence
Logical Data Independence

Physical Data Independence: The ability to modify the physical schema without causing application programs to be rewritten. The modifications at this level are occasionally necessary to improve performance.

Logical Data Independence: The ability to modify the logical schema without causing application programs to be rewritten. The modifications at this level are necessary whenever the logical structure of the database is altered.

Logical data independence is more difficult to achieve than is physical data independence, since application programs are more dependent on the logical structure of data that they access.

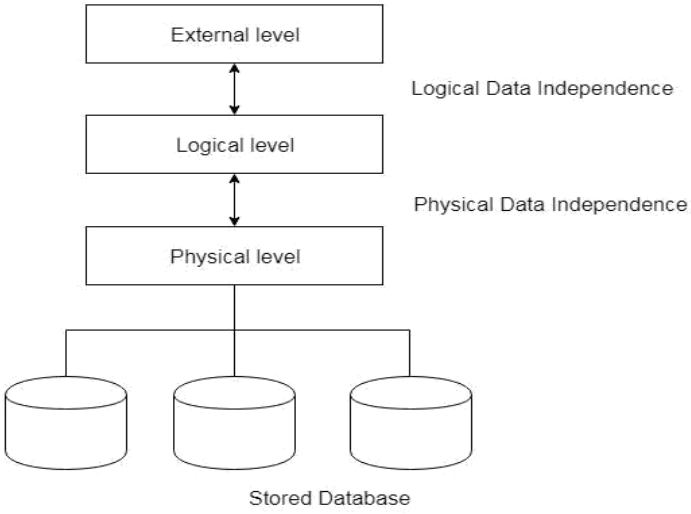


Fig 1.7

DATA INTEGRITY

Data integrity is the maintenance of, and the assurance of the accuracy and **consistency** of, **data** over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves **data**.

Data integrity is a fundamental component of information security. In its broadest use, "data integrity" refers to the accuracy and consistency of data stored in a database, data warehouse, data mart or other construct.

Data integrity is usually imposed during the database design phase through the use of standard procedures and rules. Data integrity can be maintained through the use of various error-checking methods and validation procedures.

DDL, DML, DCL

A DBMS must provide appropriate languages and interfaces for each category of users to express database queries and updates.. Database languages are used to read, update, store and maintain data in a database on computer. There are several such languages that can be used for this purpose; one of them is SQL (Structured Query Language).

A database system provides three different types of Languages, one will specify the schema, and other will express database queries, updates and maintenance.
Those are

Data-Definition Languages (DDL)
Data-Manipulation Language (DML)
Data – Control Language (DCL)

Data-Definition Languages (DDL):

DDL is used for specifying the database schema. It is a language that allows the users to define data and their relationship to other types of data. It is also used to specify the structure of each table, set of associated values with each attribute, integrity constraints, security and authorization information for each table and physical storage structure of each table on disk.

The operations that we can perform on database using DDL:

To create the database instance
To alter the structure of database
To drop database instances

The result of compilation of DDL statements is a set of tables that is stored in a special file called 'Data dictionary' or "data directory.

A data dictionary is a file that contains metadata, i.e. Data about data. This file is consulted before actual data are read or modified in the database system.

The storage structure and access methods used by the database system are specified by a set of definitions in a special type of DDL called a 'data storage and data definition language'. The result of consultation of these definitions is a set instruction to specify the implementation details of the database schemas. Which are usually hidden form the users.

The DDL commands are
Create, Alter, Drop.

Data-Manipulation Language (DML)

A data manipulation language (DML) is a language permit the users to access and manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

The operations that we can perform on database using DML:

- Retrieving data from a database
- Inserting data into a table
- Updating existing data within a table
- Delete all records from a database table
- UPSERT operation (insert or update)
- call a PL/SQL or Java subprogram
- interpretation of the data access path
- concurrency Control

It includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

A DML is language which enables users to access or manipulate data. There are basically two types.

Procedural DML : This requires a user to specify what data are needed and how to get those data from existing database.

Non procedural DML : Which require a user to specify what data are needed 'without' specifying how to get those data.

Nonprocedural DMLs are usually easier to learn and use than procedural DMLs. A user does not have to specify how to the data, these languages may generate code that is not as that produced by Procedural DML. Hence we can make remedy this difficulty by various optimization techniques.

A Query is a statement, a request for retrieval information. The portion of a DML, that involves information retrieval is called a 'Query Language'.

The DML commands are
SELECT, INSERT, UPDATE, DELETE

Data – Control Language (DCL)& Transaction Control Language(TCL)

A data Control language (DCL) is a language is used for granting and revoking user access on a database. The changes in the database that we made using DML commands are either performed or rollbacked using TCL.

To grant access to user.
To revoke access from user.
To persist the changes made by DML commands in database.
To rollback the changes made to the database.

The DCL commands are
GRANT, REVOKE, COMMIT, ROLLBACK

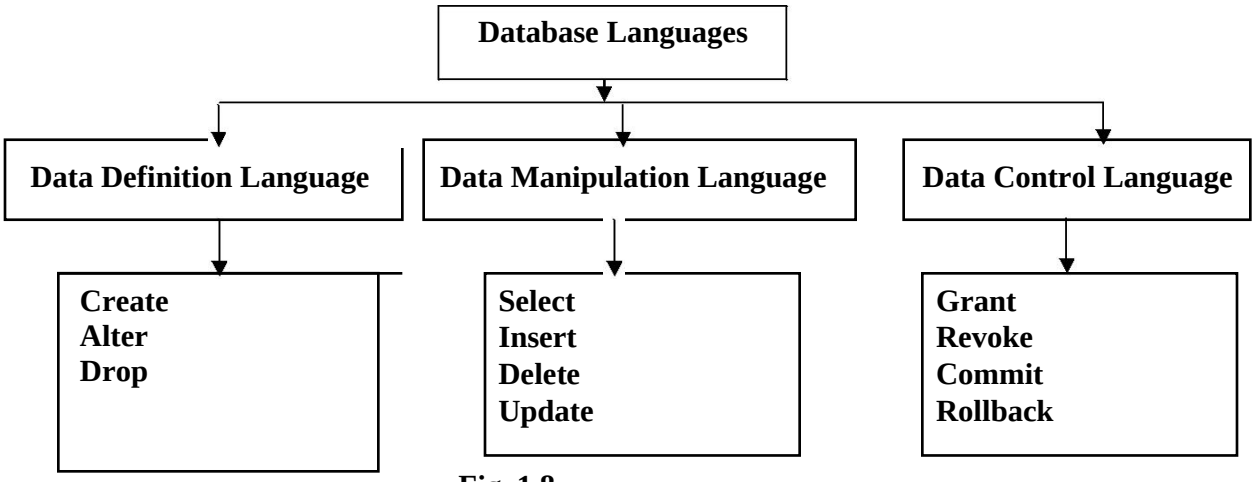


Fig. 1.8

RESPONSIBILITY OF DATABASE MANAGER

A Database Manager is a program module which provides the interfacebetween the low level data stored in the database and the application programand queries submitted to the system.

Responsibilities of Database Manager

Interaction with File Manager : The row data is stored on the disk using the file system which is usually provided by conventional operating system.

Integrity Enforcement : The data values stored in the database must satisfy certain types of consistency constraints.

Security Enforcement : Not every user of the database needs to have access to the entire content of the database.

Backup and Recovery : It is the responsibility of database manager to detect such failures and restore the database to a state that existed prior the occurrence of the failure this is usually accomplished through the backup and recovery processor.

Concurrency Control : It is necessary for the system to control the interaction among the concurrent users, and achieving such control is one of the responsibilities of database manager.

Authorization Control : This module checks that the user has necessary authorization to carry out the required function.

Command Processor : Once the system has checked that the user has authority to carry out the operation control satisfies all necessary integrity constraints such as key constraints.

Integrity Checker : For an operation that changes the database the integrity checker checks that the requested operation satisfies all necessary integrity constraints such as key constraints.

Query Optimizer : This module determines on optional strategy for the query execution.

Scheduler : This module is responsible for ensuring that concurrent operations or transactions on the database proceed without conflicting with one another.

Recovery Manager : This module ensures that the database remains in a consistent state in the presence of failures. It is responsible for transaction commit and abort that is success or failure of transaction.

Buffer Manager : It is responsible for the transfer of data between main memory and secondary storage such as disk and tape. The recovery manager and buffer manager are sometimes collectively referred to as data manager. The buffer manager is sometimes known as cache manager.

Query Processor : This is responsible for receiving query language statements and changing them form the English like syntax of the query language to from the DBMS can understand. The query language processor usually consists of two parts i.e. 1. The purser 2. Query Optimizer.

DATA BASE ADMINISTRATOR

Database Administrator is a person who has central control over the system is called data base administrator. **DBA** is the final authority on deciding the structure accessing strategies, storage criteria, user preventing data base from frauds. The functions or responsibilities of DBA includes

Schema Definition.
Storage Structure and access method definition.
Schema physical Organization and Modification.
Granting of authorization for data access .
Routine maintenance.

Schema Definition: The DBA creates the original database schema by executing a set of definition statements in the DDL.

Storage structure and access Method Definition: DBA will decide the actual storage structure and different access methodologies for the database.

Schema Physical Organization and Modification: The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, are to alter the physical organization to improve performance.

Granting of Authorization for data access: By granting different types of authorization, the database administrator can regulate which of the database various users can access.

Routine Maintenance: DBA is the final authority to regulate the daily activities.

As a whole , the DBA jobs are

Creating primary database storage structures
Modifying the structure of the database
Monitoring database performance and efficiently
Transferring data between the database and external file
Monitoring and reestablishing database consistency
Controlling and monitoring user access to the database
Manipulating the physical location of the database.

Data Base Users

Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

Application Programmers - They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, JAVA, etc. These queries are converted into object code to communicate with the database. For example, writing a JAVA program to generate the report of

employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the JAVA Program.These users implement specific application programs to access the stored data. They must be familiar with the DBMSs to accomplish their task.

Sophisticated Users - They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement. In short, we can say this category includes designers and developers of DBMS and SQL.

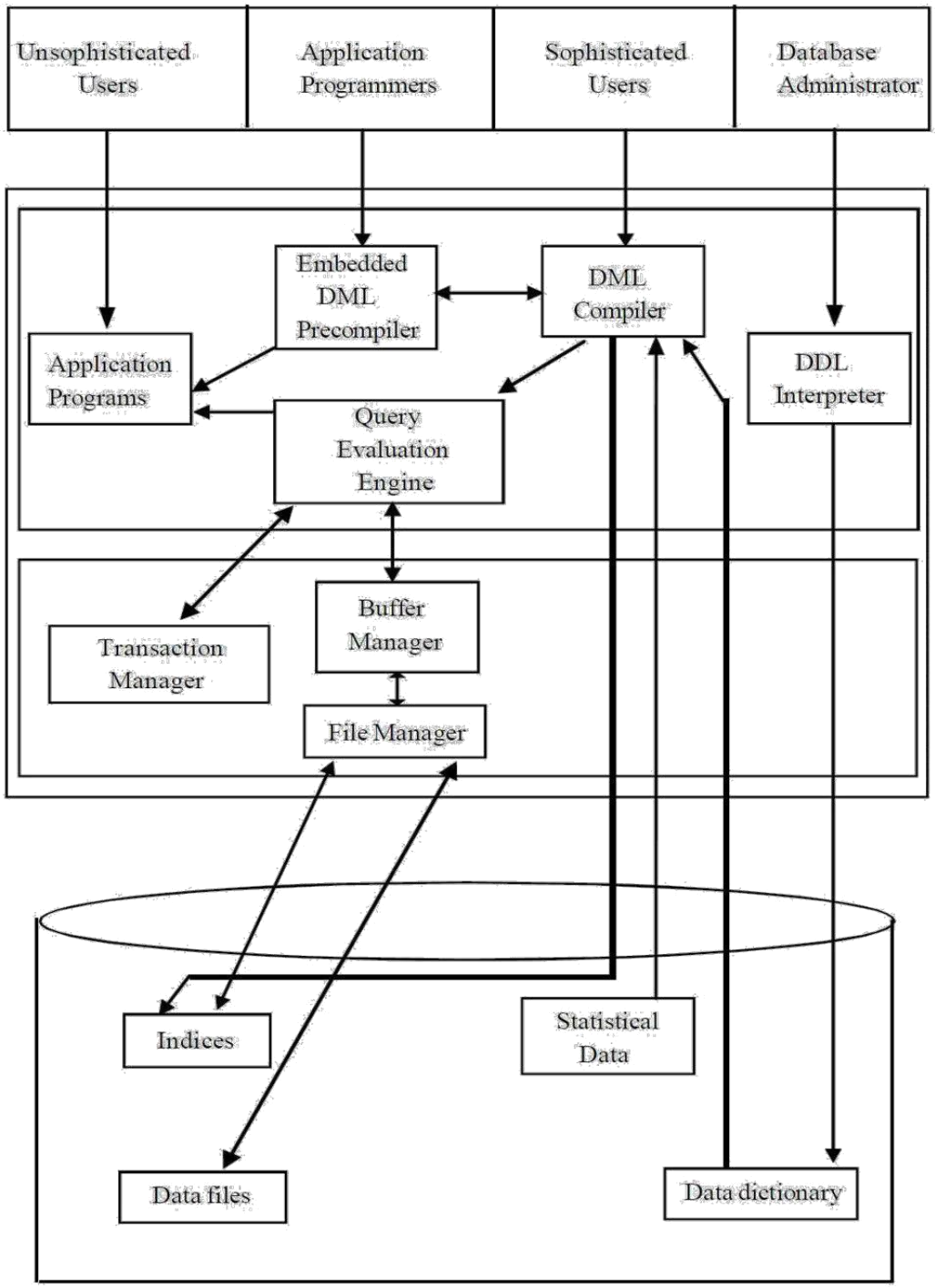
Sophisticated users are those who have their own way of accessing the database. This means they do not use the application program provided in the system. Instead, they might define their own application or describe their need directly by using query languages. These specialized users maintain their personal databases by using ready-made program packages that provide easy-to-use menu driven commands, such as MS Access.

Specialized Users - These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.

Stand-alone Users - These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.

Native Users - these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

Database Administrators (DBA): This may be one person or a group of people in an organization responsible for authorizing access to the database, monitoring its use and managing all of the resources to support the use of the entire database system.


Unsophisticated Users
Application Programmers
Sophisticated Users
Database Administrator
Embedded DML Precompiler
DML Compiler
Application Programs
DDL Interpreter
Query Evaluation Engine
Buffer Manager
Transaction Manager
File Manager
Indices
Statistical Data
Data files
Data dictionary

Quick Reference

A database can be defined as a collection of inter related relevant data stored together to serve multiple applications.

A DBMS is a collection of inter related data and a set of programs to access that data. The collection of data usually called as database. It contains the data or information about one particular enterprise or an organization.

The system hides certain details of how the data are stored and maintained is Data Abstraction.

Data Model is a logical structure of Database. It describes the design of database to reflect entities, attributes, relationship among data, constrains etc.

The overall logical structure of a database can be expressed graphically by an E-R-Diagram.

The collection of information stored in the database at a particular moment is an

'Instance' of the database.

The collection of information stored in the database at a particular moment is an

'Instance' of the database.

The accuracy and consistency of data stored in a database, data warehouse, data mart or other construct is data integrity.

The DDL commands are Create, Alter, Drop.

The DML commands are SELECT, INSERT, UPDATE, DELETE.

The DCL commands are GRANT, REVOKE, COMMIT, ROLLBACK

Database Administrator is a person who has central control over the system is called data base administrator. **DBA** is the final authority on deciding the structure accessing strategies, storage criteria, user preventing data base from frauds.

application programmer: user who implements specific application programs to access the stored data

application user: accesses an existing application program to perform daily tasks.

database administrator (DBA): responsible for authorizing access to the database, monitoring its use and managing all the resources to support the use of the entire database system

end user: people whose jobs require access to a database for querying, updating and generating reports

sophisticated user: those who use other methods, other than the application program, to access the database

Short Answer Type Questions

What is database ?
What is Data Processing ?
What is a DBMS ?
What are components of DBMS ?
Define Instance.
Define Schema and Sub-Schema.
What is data independence ?
What is difference between physical and logical data Independence?
Who will be called as DBA.
What is Meta data or Data Dictionary ?
What the three levels of data abstraction ?
Write different types of database users.
Expand the terms DDL, DML, DCL?
Write the commands of DDL, DML, DCL

Long Answers Type Questions

What are the advantages of DBMS over File Processing system ?
Explain about different data models.
What is data abstraction? Explain in detail ?
Explain DDL, DML and DCL commands with examples.
What are the responsibilities of Database Manager ? Explain
What are the functions of DBA ?
Discuss briefly about different types database users.

UNIT 2

ENTITY AND RELATION SHIP

Entity and Entity sets
Attributes
Relationship and Relationship sets
Mapping Constraints
E - R Representation Symbols
Drawing E - R Drawings

Learning Objectives:

Understanding and identifying Entities and Entity Sets.
Understanding and identifying the properties of different types of Attributes.
Understanding and identifying Relationships and Relationship sets.
Understand different mapping constraints
Drawing entity relationship diagrams

Entity and Entity sets

The **Entity – Relationship**(E - R) data model was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical by structure of a data base. The E-R model lies in its representation of the several semantic data models; the semantic aspect of the model lies in its pre presentation of the meaning of the data. The E-R model is very useful in mapping the meanings and interactions of real – world enterprises into a conceptual schema. Because of this usefulness, many database design tools draw on concepts from the E – R Model. The E – R data model employs three basic notions: **Entity, Relationship sets and Attributes**.

2.1.a Entity

An **Entity** is an "**object**" that exists and is distinguishable from other objects. The distinction is accomplished by associating with is an entity set of attributes which describe the object. An Entity is represented by a set of attributes. These attributes are the descriptive properties possessed by each entity. For example, Roll Number of a Student, is an entity, it is uniquely identifies a student in a class. i.e. the Roll Number of a student is distinguishable from one other .

Entity Sets

The **Entity Set** is a set of entities of the same type, that share the same properties or Attributes. Ex : Persons having an account at bank. or Each student having Student id **or** Admission Number.

Std_id Or Admission No.	Std_name	Address
1001	Purhoothika	Vijayawada
1002	Amar	Amarawathi
1003	Sankari	Srilanka
1004	Venkatesh	Tirupati
1005	Shiva	Varanasi
1006	Ganesh	Rajamahendravaram
1007	Shanmukh	Tiruchandur
1008	Vaaraahi	Thanjasvur

The set of all students in a class., can be defined as the entity-set. Who are students of an university, for example, can be defined as the entity set student. Similarly, the entity set Admission might represent the set of all admission awarded by a particular university. The individual entities that constitute a set are said to be the extension of the entity set. Thus, all the individual students are the extension of the entity set student.

Entity sets are is of two types
- Strong Entity Set
- Weak Entity Set

Strong Entity Set: An Entity set which is having sufficient attribute to form aprimary key and does not depending on other entity set to form a primary key iscalled strong entity set.

Weak Entity Set: If existence of an entity set of depends on other entities then it is called weak entity set. A weak entity set does not have sufficient of attributes to form a primary key.

Attributes : Characteristics or properties of an entity is called Attribute. The values for these characteristics are called attribute value.For instance the employee entity might include the following attributes, Employee ID, Employee name etc. For each attribute, there is a set of permitted values, called the **domain,** or **value set**, of that attribute. An attribute of an entity set is a function that maps from the entity set into a domain. Since an entity set may have several attributes, each entity can be described by a set of (attribute, data value) pairs, one pair for each attribute of the entity set.

The attributes can be classified in to
- Simple Attribute or Atomic Attribute
- Complex/ composite attributes
- Single – valued attributes
- Multi - valued attributes
- Derived attribute
- Null Attribute
- Key Attribute

An attribute, as used in the E – R model, can be characterized by the following attribute types.

Simple attributes or Atomic Attributes : These attributes have been simple; that is, they have not been divided into subparts.

Example: Assume Student is an **entity** and its **attributes** are Admission No. , Name, Age, Address and Phone no. Here the Admission No. (attribute) of student (entity) cannot further divide. In this example Admission No. is an **atomic attribute**..

Composite attributes : The attributes, which can be sub divided in to sub parts. An attribute that can be divided into smaller independent attribute is known as **composite attribute.**

Example: assume Student is an **entity** and its **attributes** are Name, Age, Address and Phone no. Here the address (attribute) of student (entity) can be further divide into House no, city and so on Student Name, Which can be divided in parts like First name, Middle name and Last name. Note also that a composite attribute may appear as a hierarchy. In the composite attribute address, its component attribute street can be further divided into street_number, street_name, and Door _ number etc.

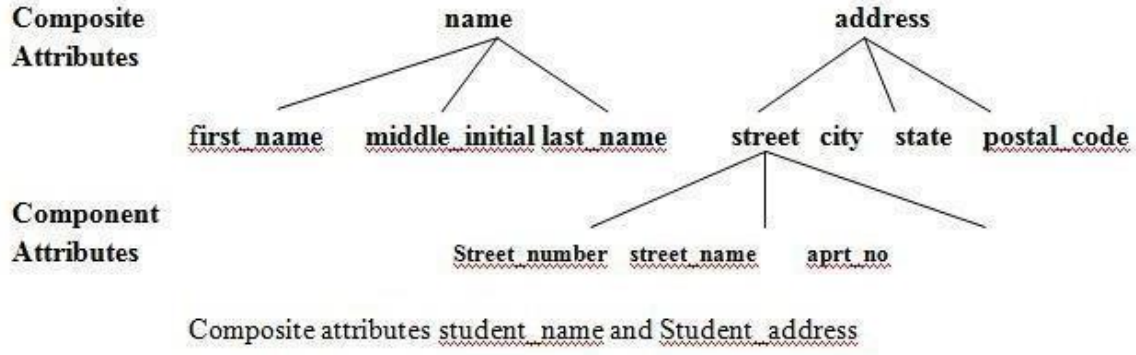


Fig 2.1.a

Single valued attributes: An attribute that has only single value for an entity is known as **single valued attribute.**.

Example: assume Student is an **entity** and its **attributes** are Name, Age, Sex, Address and Phone no. Here the Sex (attribute) of student (entity) can have only one value. Here, Sex is **single valued attribute**. i.e. Male or Female similarly, if marital status, Married or Unmarried.

Multi valued attributes: An attribute that can have multiple values for an entity is known as **multi valued attribute.**

Example: assume Student is an **entity** and its **attributes** are Name, Age, Address and Phone no. Here the Phone no (attribute) of student (entity) can have multiple value because a student may have many phone numbers. Here, Phone no is **multi valued attribute**.

Derived attribute: An attribute that can be derived from another attribute is known as **derived attribute.** The value of this type of attribute can be derived from the values of other related attributes or entities. The value of a derived attribute is not stored but is computed when required.

Null Attributes: An attribute takes a null value when an entity does not have a value for it. The null value may indicate "not applicable" – that is, that the value does not exist for the entity. An attribute, which has not any value for an entity is known as null valued attribute.

Key Attribute: An attribute that has unique value of each entity is known as ***key attribute.*** **E**xample, every student has unique roll no. Here roll no is **key attribute**.

Relationship and Relationship sets

A **Relationship** is an association among several entities. For example, we can define a relationship between student entity and bank account entity. The student named Hasini having savings bank account in SBI ,Kothapeta branch with an account number SB-101.

A **Relationship set** is a set of relationships of the same type. Formally, it is a mathematical relation on $n \geq 2$ (possibly nondistinct) entity sets. If *E1 ,E2, ……, En*are entity sets, then a relationship set *R* is a subset of { (e_1, e_2, ……,e_n) |$e_1 \in E_1$, $e_2 \in E_2$,.....e_n E_n } Where (e_1, e_2,.....,e_n) is a relationship.

The association between entity sets is referred to as participation; that is, the entity sets $E_1, E_2, \ldots\ldots, E_n$ **participate** in relationship set *R*. A **relationship instance** in an E – R schema represents an association between the named entities in the real – world enterprise that is being modeled.

The function that an entity plays in a relationship is called that entity's **role**. Since entity sets participating in a relationship set are generally distinct, roles are implicit and are not usually specified. However, they are useful when the meaning of a relationship needs clarification. Such is the case when the entity sets of a relationship set are not distinct; that is, the same entity set participates in a relationship set more **cursive** relationship set, explicit role name are necessary to specify how an entity participates in a relationship instance.

A relationship may also have attributes called **descriptive attributes.** A relationship instance in a given relationship set must be uniquely identifiable from its participating entities, without using the descriptive attributes.

Mapping Constraints

An E – R enterprise schema may define certain constraints to which the contents of a database must conform. In this section, we examine mapping cardinalities, key constraints, and participation constraints.

Cardinalities : Mapping Cardinalities**,** or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set.

Mapping Cardinalities are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets. In this section, we shall concentrate on only binary relationship sets.

For a binary relationship set R between entity sets A and B, the mapping cardinality must be one of the following. There are 4 types of mapping cardinalities.

ONE – to – ONE relationship
MANY – to – MANY relationship
ONE – to – MANY relationship
MANY – to – MANY relationship

ONE – to – ONE relationship: An entity in A is associated with atmost one entity in B is also associated with atmost one entity in A.

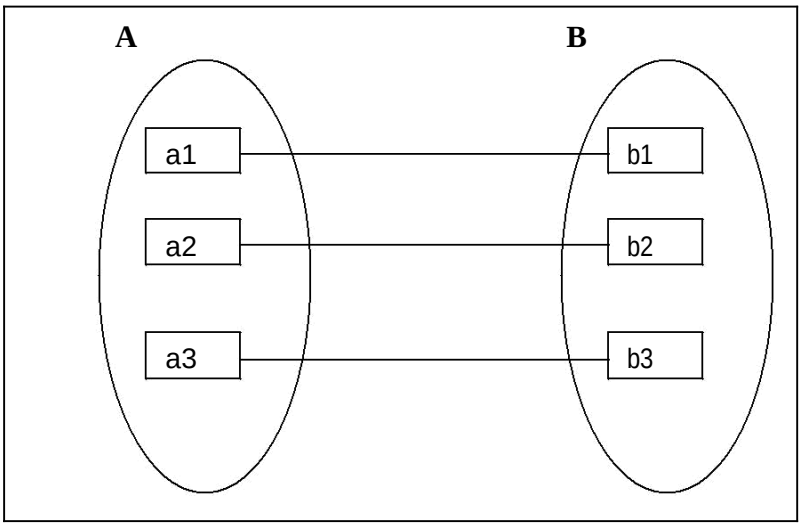


Fig 2.4.1

Example : Relationship between the entities principal and college. i.e., Principals can lead a single college and a principal can have only one college.

Many – to – One relationship: An entity set in A is associated with atmost one entity in B, An entity in B however can be associated with any number of entities in A.

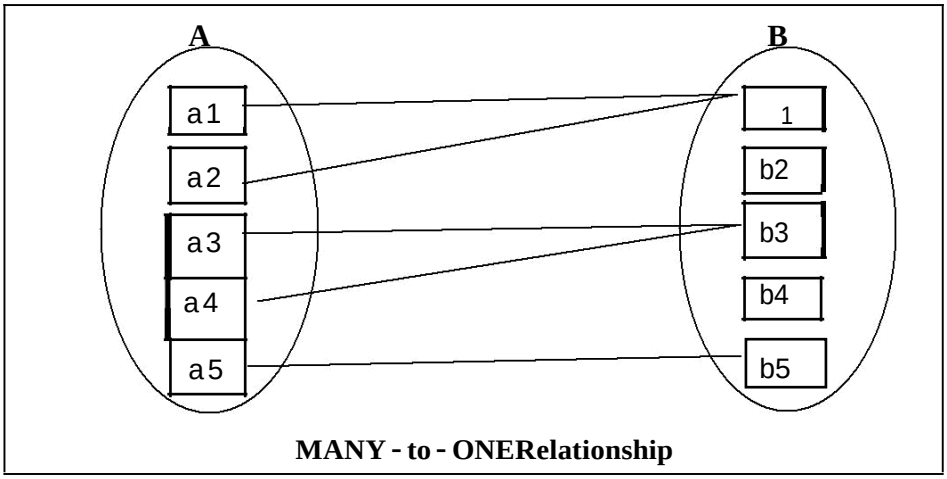


Fig 2.4.2

Example : Relationship between the entities Districts and state .i.e. many districts belong to a single state but many states cannot belong to single district.

ONE – to - MANY relationship: An entity set A is associated with any number of entities in B. An entity in B, however can be associated with atmost one entity in A.

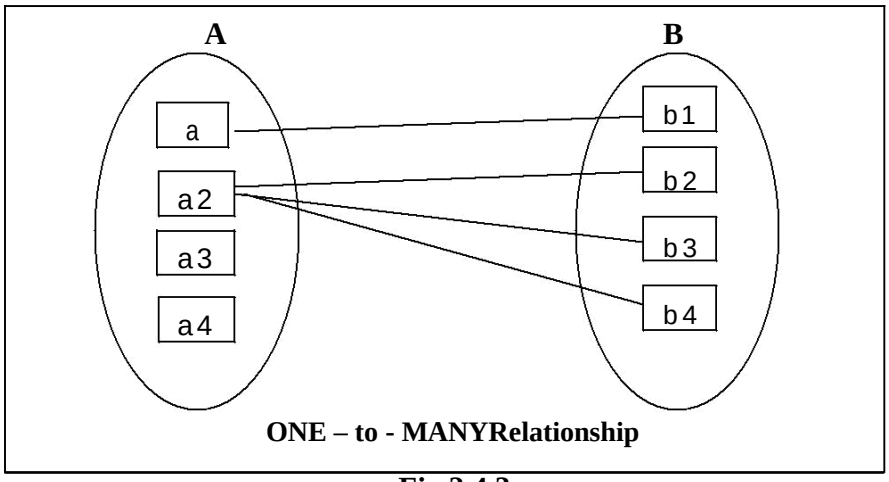


Fig 2.4.3

Example : Relationship between the entities class and student i.e., a class can have many students but a student cannot be in more than one class at a time**.**

MANY – to – MANY relationship: An entity set in A is associated with any number of entities in B and an entity set in B is associated with any number of entities in A.

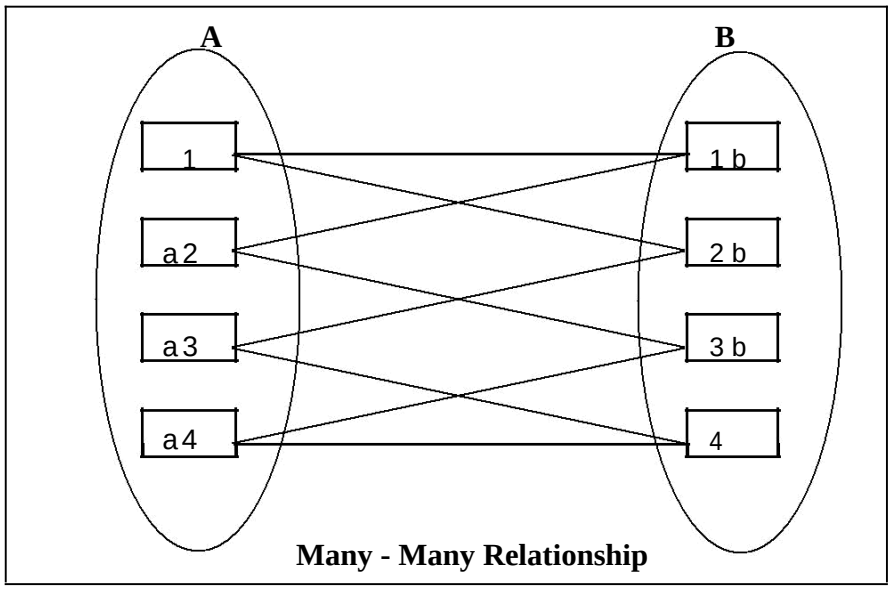


Fig 2.4.4

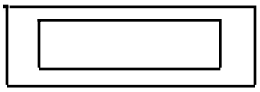
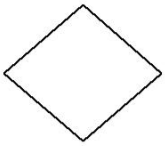
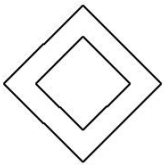
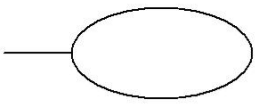
Example : Relationship between the Entities College and course .i.e. a college can have many courses and course can be offered by many colleges.

The appropriate mapping cardinality for a particular relationship set obviously depends on the real – world situation that the relationship set is modeling.

E - R Representation Symbols

An **E-R diagram** can express the overall logical structure of a database graphically. E-R diagrams are simple and clear – qualities that may well account in large part for the widespread use of the E-R model. Such diagram consists of the following major components.

Rectangles, which represent entity sets.
Ellipses, which represent attributes
Diamonds, which represent relationship sets
Lines, which link attributes to entity sets and entity sets to relationship sets
Double ellipses, which represents multivalued attributes
Dashed ellipses, which denote derived attributes.
Double Lines, which indicate total participation of an entity in a relationship set.
Double Rectangles, which represent weak entity sets

Symbol	Represented ERD Property
	**Entity**
	**Weak Entity**
	**Relation Ship**
	**Identifying Relationship**
	**Attribute**

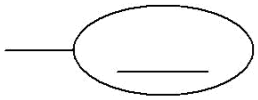
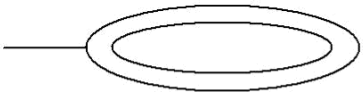
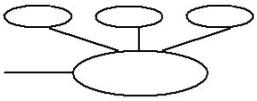
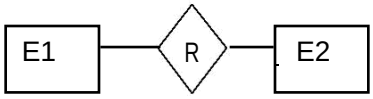
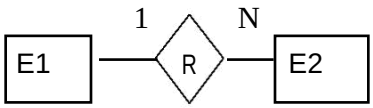
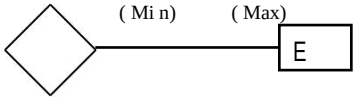
	**Key Attribute**
	**Multivalued Attribute**
	**Composite Attribute**
	**Derived Attribute**
	**Total Participation of E 1 in R**
	**Cardinality Ratio 1: N for E 1, E2 in R**
	**Structural constraint (Min, Max) on participation of E in R**

Fig 2.5

1.6 Drawing E - R Drawings

ER diagram: An Entity Relationship Diagram (ERD) is a visual representation of **different entities within a system and how they relate to each other**. It describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between **entities**. An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

For example, the elements Author, Book, and a consumer may be described using ER diagrams the following way:

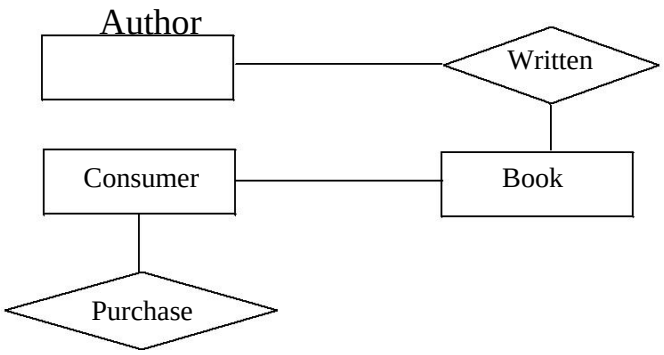


Fig 2.6.1

An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one building.

History of ER Diagrams:

Although data modeling has become a necessity around 1970's there was no standard way to model databases or business processes. Although many solutions were proposed and discussed none were widely adopted.

Entity–relationship modeling was developed for database design by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously. Some ER models show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontologies.

Peter Chen is credited with introducing the widely adopted ER model in his paper "The Entity Relationship Model-Toward a Unified View of Data". The focus was on entities and relationships and he introduced a diagramming representation for database design as well.

His model was inspired by the data structure diagrams introduced by Charles Bachman. One of the early forms of ER diagrams, Bachman diagrams are named after him.

ER Diagrams Usage

What are the uses of ER diagrams? Where are they used? Although they can be used to model almost any system they are primarily used in the following areas.

They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it's commonly used for database troubleshooting as well.

ER Models in Database Design

They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to

visualize database tables and their relationships it's commonly used for database troubleshooting as well.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.There is a tradition for ER/data models to be built at two or three levels of abstraction.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is,

at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design."

Entities can be thought of as **nouns** such as car, bank, student, or product.

In an ER Diagram, entities are the most important parts. To proceed, we will be creating a conceptual ER diagram of a simple system in which a student registers for a course that is taught by a professor. In this example, the three entities are "Student," "Course," and "Professor."

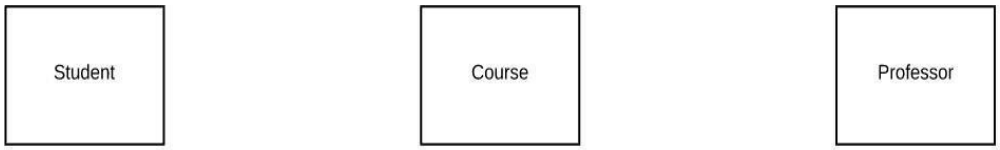


Fig 2.6.2

Examples: a computer, an employee, a song, a mathematical theorem, etc.

A **relationship** captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. The relationships "Registers for" and "Teaches" effectively explain the interactions between the three entities for the above example.

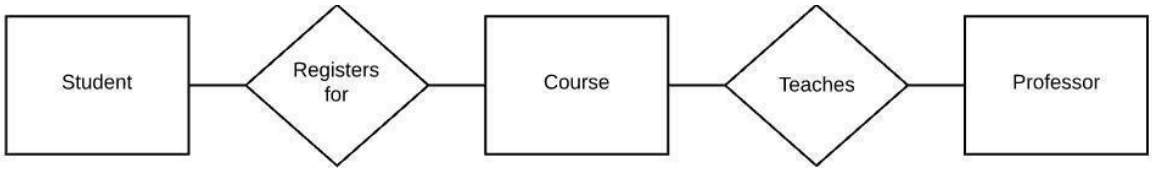


Fig 2.6.3

Examples:an owns relationship between a company and a computer,
a supervises relationship between an employee and a department, a performs relationship between an artist and a song, a proves relationship between a mathematician and a conjecture, etc.

Entities and relationships can both have attributes.

Attributes are necessary to model what characteristics will be included with each entity. Attributes such as "IDNumber," "Name," and "SKU" are common attributes.

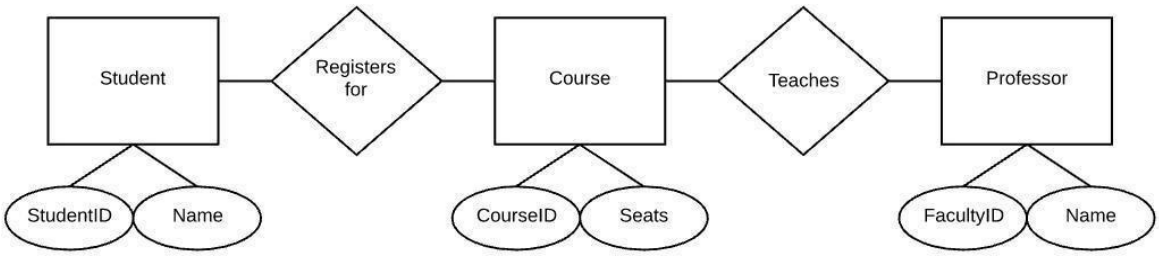


Fig 2.6.4

Examples: an employee entity might have a Social Security Number (SSN) attribute, while a proved relationship may have a date attribute.

Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes, which is called the entity's primary key.

A weak entitydoes not have a minimal set of uniquely identifying attributes to form primary key.

Weak Entity Fig 2.6.5

Entity–relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets (all entities of the same entity type) and relationship sets (all relationships of the same relationship type).

Examples: a particular songis an entity; the collection of all songs in a database is an entity set; the eaten relationship between a child and her lunch is a single relationship; the set

of all such child-lunch relationships in a database is a relationship set. In other words, a relationship set corresponds to a relation in mathematics, while a relationship corresponds to a member of the relation.

Certain cardinality constraints on relationship sets may be indicated as well.

Chen proposed the following rules for mapping natural language descriptions into ER diagrams:

English grammar structure	ER structure
Common noun	Entity type
Proper noun	Entity
Transitive verb	Relationship type
Intransitive verb	Attribute type
Adjective	Attribute for entity
Adverb	Attribute for relationship

Relationships, roles and cardinalities

A relationship describes how entities interact.The example of a relationship and its rolesDescribes a relationship "marriage" and its two roles "husband" and "wife".

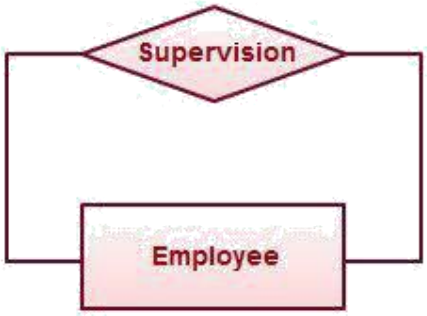
A person plays the role of husband in a marriage (relationship) and another person plays the role of wife in the (same) marriage. These words are nouns.

The entity “Carpenter” may be related to the entity “table” by the relationship “builds” or “makes”. Relationships are represented by diamond shapes and are labeled using verbs.

Relationship Fig 2.6.6

Recursive Relationship

If the same entity participates more than once in a relationship it is known as a recursive relationship. In the below example an employee can be a supervisor and be supervised, so there is a recursive relationship.



Recursive Relationship Fig 2.6.7

Cardinality and Ordinality

These two further defines relationships between entities by placing the relationship in the context of numbers. In an email system, for example, one account can have multiple contacts. The relationship, in this case, follows a "one to many" model. There are a number of notations used to present cardinality in ER diagrams. Chen, UML, Crow's foot, Bachman are some of the popular notations. Creately supports Chen, UML and Crow's foot notations.The following example uses UML to show cardinality.

Cardinality and Ordinality Fig 2.6.8

How to Draw ER Diagrams

Below points show how to go about creating an ER diagram.

Identify all the entities in the system. An entity should appear only once in a particular diagram. Create rectangles for all entities and name them properly.

Identify relationships between entities. Connect them using a line and add a diamond in the middle describing the relationship.

Add attributes for entities. Give meaningful attribute names so they can be understood easily.

Fig 2.6.9

In a complex system, it can be a difficult to identify the relationships. The following points should be kept mind while drawing the complex ER Diagrams.

Points to remember while drawing ER Diagrams.

Provide a precise and appropriate name for each entity, attribute, and relationship in the diagram.
Terms that are simple and familiar always beats indistinct, technical-sounding words.
For naming entities, remember to use singular nouns.
Adjectives may be used to distinguish entities belonging to the same class (part-time employee and full-time employee, for example).
Attribute names must be meaningful, unique, system-independent, and easily understandable.
Remove indistinct, redundant or unnecessary relationships between entities.
Never connect a relationship to another relationship.
Make effective use of colors. You can use colors to classify similar entities or to highlight key areas in your diagrams.

Benefits of ER diagrams

ER diagrams constitute a very useful framework for creating and manipulating databases. First, ER diagrams are easy to understand and do not require a person to undergo extensive training to be able to work with it efficiently and accurately. This means that designers can use ER diagrams to easily communicate with developers, customers, and end users, regardless of their IT proficiency. Second, ER diagrams are readily translatable into relational tables which can be used to quickly build databases. In addition, ER diagrams can directly be used by database developers as the blueprint for implementing data in specific software applications. Lastly, ER diagrams may be applied in other contexts such as describing the different relationships and operations within an organization.

ER Diagram for college:

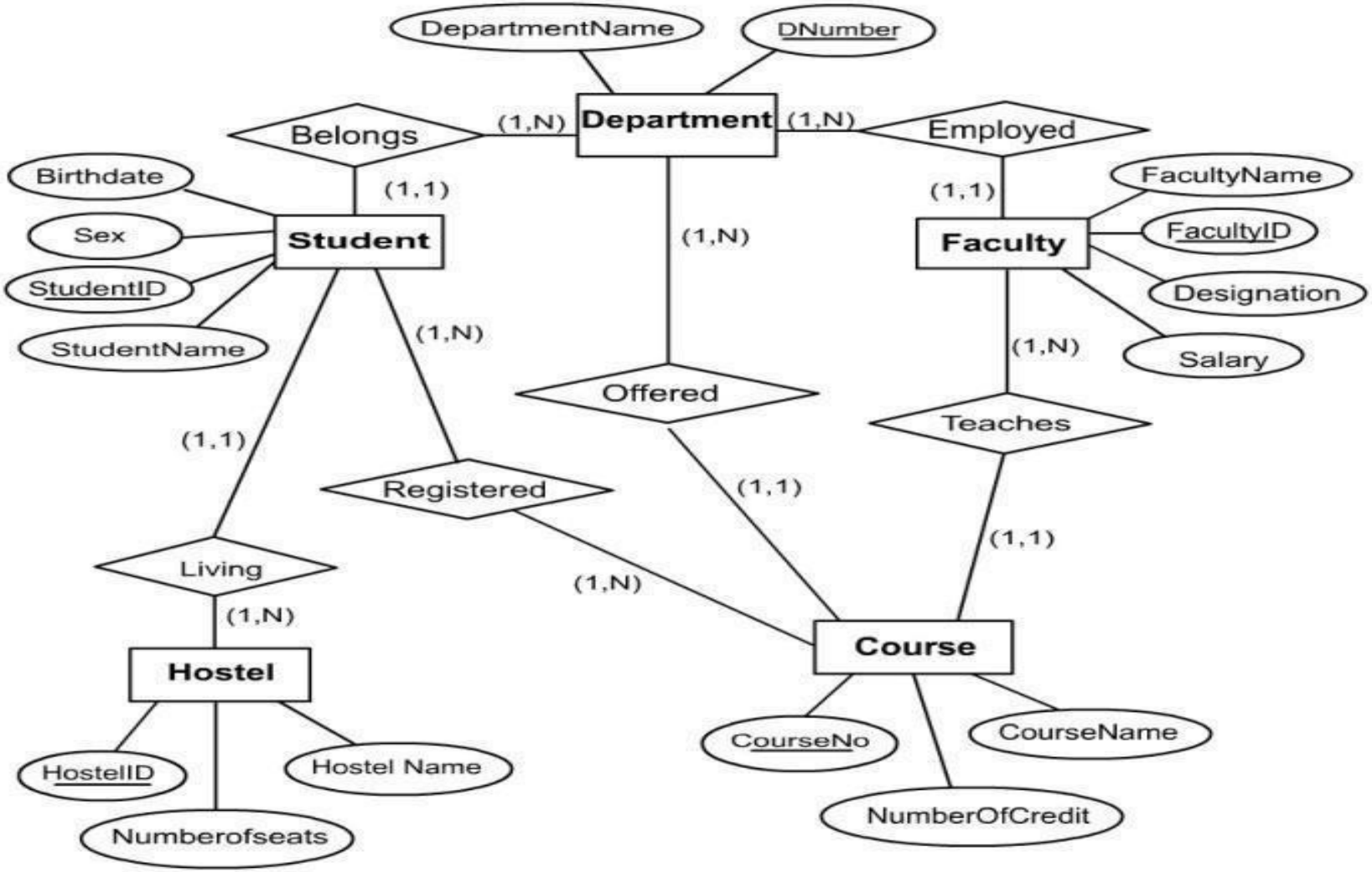


Fig 2.6.10

ER Diagram for Hospital:

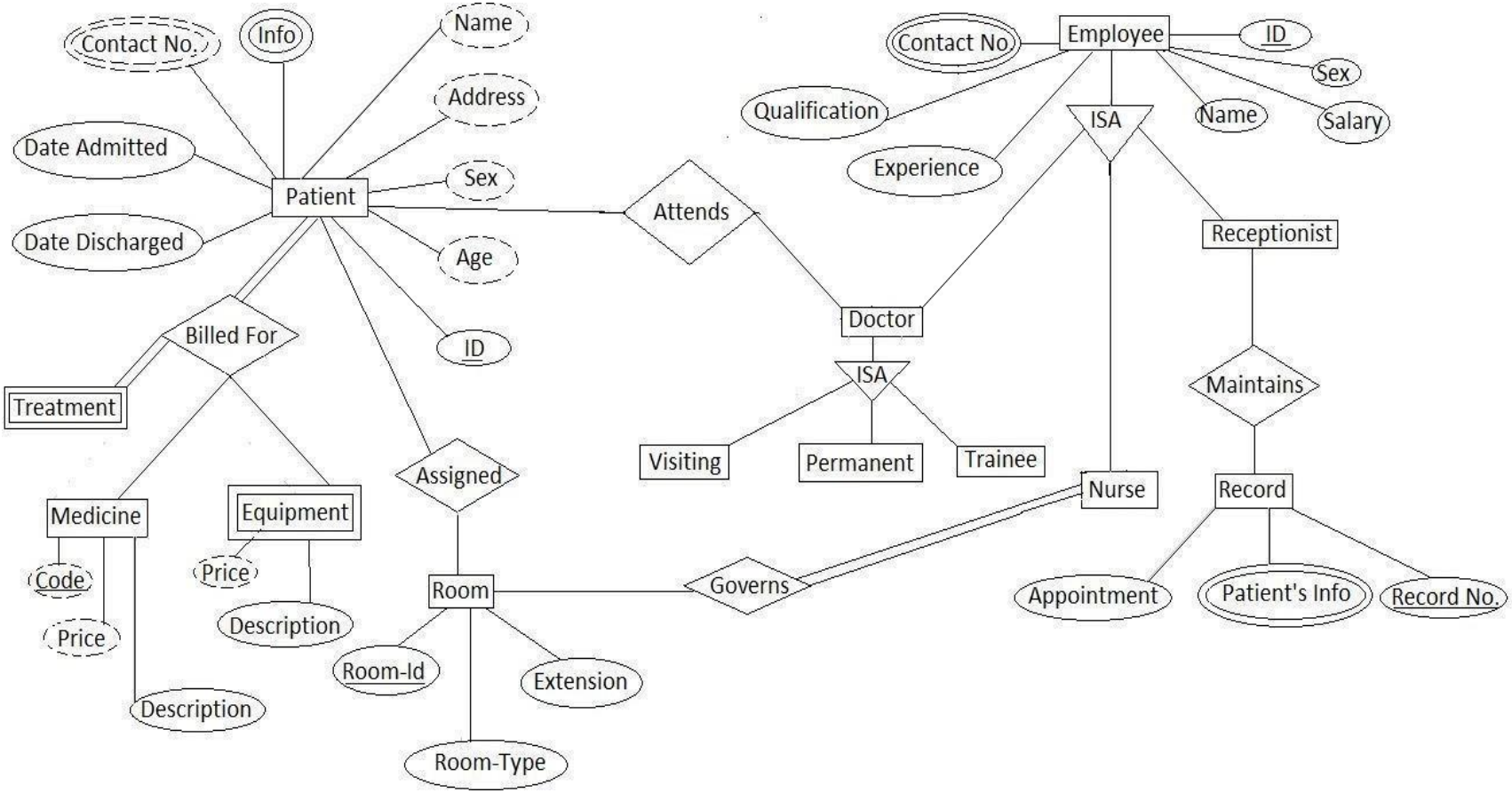


Fig 2.6.11

Quick Reference

The **entity relationship (E-R)** data model is a widely used data model for database design. It provides a convenient graphical representation to view data, relationships, and constraints.

An entity is an object that exists in the real world and is distinguishable from other objects. We express the distinction by associating with each entity a set of attributes that describes the objects.

A relationship is an association among several entities. A **Relationship set** is a collection of relationships of the same type. And an entity set is a collection of entities of the same type.

A Super key of an entity set is a set of one or more attributes that, taken collectively allows us to identify uniquely an entity in the entity set.

Mapping Cardinalities express the number of entities to which another entity can be associated via a relationship set.

An entity set that does nothave sufficient attributes to form a primary key is termed a **weak entity set**. An entity set that has a primary key is termed a **strong entity set.**

Domain: It is a set of permissible values for each attribute simply a column is known as a domain.

Degree: It is number of columns associated with a table. This property is used to describe the total number of columns a table consists.

Cardinality: This is the number of rows in any given table.

Object: An Entity which is a “thing” in the real world with an independent existence.

Literal: Literal is a value that doesn’t have an object identifier. However, the value may have a simple or complex structure. There are 3 types of literal.

Atomic Literal
Collection Literal
Structured Literal

Foreign Key: A Foreign Key is a column whose values are derived from the primary key or unique key of some other tables. It represents the relationships between tables.
Tuple: Tuples are the rows / records of a table. The tuple gives the total set of properties, attributes of an entity which can be used to describe it.

Short Answer Type Questions

What is Entity and Entity set ?
What is Relationship and Relationship set ?
What is Weak Entity and strong Entity ?
What is an attribute ? What are the different types in it ?
What is a Domain ?
What is tuple ?
What is Degree of table ?
What are the symbols used in E-R diagram ?
What are mapping cardinalities ?

Long Answers Type Questions

Explain the mapping constraints with neat diagram.
Draw an ER diagram by showing the relationship between a student and Bank.

Note : Practice more on ER Diagrams using different examples and different symbols like weak , strong entities, attributes etc

UNIT 3

Relational Model

Structure

Table Structure

Domain, Range, degree of a table tuple, types and attributes
Types of keys
Formal query languages
3.5Commercial query languages
3.6CODD rules

Learning objectives

After studying this unit ,the student will be able to

Understand the relational model of database management
Understand about the rules laid by E.F.Codd
Understand the tables, attributes and keys

Introduction

The relational model was invented by Edgar F. Codd as a general model of data, and subsequently promoted by Chris Date and Hugh Darwen among others. In The Third Manifesto (first published in 1995) Date and Darwen attempt to show how the relational model can allegedly accommodate certain "desired" object-oriented features.

Relational model in DBMS is the data and relationships are represented by collection of inter-related tables. Each table is a group of columns and rows, where column represents attribute of an entity and rows represents records.

The **relational model** for database management is an approach for managing data using a structure and language consistent with first-order predicate logic, first described in 1969 by computer scientist Edgar F. Codd, where all data is represented in terms of tuples, grouped into relations. A database organized in terms of the relational model is a relational database.

The purpose of the relational model is to provide a declarative method for specifying data and queries. Users directly state what information the database contains and what information they want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for answering queries.

A table in an SQL database schema corresponds to a predicate variable; the contents of a table to a relation; key constraints, other constraints, and SQL queries correspond to predicates.

However, SQL databases deviate from the relational model in many details, and Codd severely argued against deviations that compromise the original principles.

The relational model's central idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values. The content of the database at any given time is a finite (logical) model of the database, i.e. a set of relations, one per predicate variable, such that all predicates are satisfied. A request for information from the database (a database query) is also a predicate.

Table Structure

Structure of Relational Database:

Relational database systems are the most common DBMS today. These relational DBMSs organize data into separate structures called **tables,** which can be **linked** via common information to make data storage more efficient. A DBMS is like a traditional filing system in that it stores individual groups and pieces of information. Like a filing system, a DBMS consists of separate components, like the cabinet, drawers and folders. A relational DBMS has the
following basic components:

database - the complete collection of information
tables - a group of data items with a common theme
records - an individual data item
fields - a separate piece of information which describe the data item

A table is an accepted visual representation of a relation. Each row in a table has its own unique key. Rows in a table can be linked to rows in other tables by adding a column for the unique key of the linked row (such columns are known as foreign keys).

A relation is defined as a set of tuples that have the same attributes. A tuple usually represents an object and information about that object. Objects are typically physical objects or concepts. A relation is usually described as a table, which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints.

The relational model specifies that the tuples of a relation have no specific order and that the tuples, in turn, impose no order on the attributes. Applications access data by specifying queries, which use operations such as select to identify tuples, project to identify attributes, and join to combine relations. Relations can be modified using the insert, delete, and update operators. New tuples can supply explicit values or be derived from a query. Similarly, queries identify tuples for updating or deleting.

Tuples by definition are unique. If the tuple contains a candidate or primary key then obviously it is unique; however, a primary key need not be defined for a row or record to be a tuple. The definition of a tuple requires that it be unique, but does not require a primary key to be defined. Because a tuple is unique, its attributes by definition constitute a superkey.

Example:
Student table with 3 columns and Five records

Table: Student

Student_id	Student_Name	Student_Age
13635	SASI REKHA	16
13644	LAKSHMI DURGA	17
13669	CHANDINI	16
13675	YAMINI	16
13689	MOUNIKA	16

Table: Course

Student_id	Course_Id	Course_Name
13635	319	C.S.E
13644	319	C.S.E.
13669	311	C.G.A.
13675	303	C.S.
13689	303	C.S.

Here Student_id, Student_Name & Student_Age are attributes of table Student and Student_id, Course_Id & Course_Name are attributes of table Course. The rows with values are the records OR tuples. For each of the above attributes, there is a set of values, which is known as the DOMAIN of the attribute.

In the above relation Student, there are five tuples (Rows), each row identifying the details of a particular student with respect to attributes.

Domain, Range, Degree of a table Tuple, types and attributes

Tuple : "Row of relation (Table) is referred as tuple. Tuple having a set of 'n' numbers of attributes are called as n-tuple.

Range: A range is a common database operation that retrieves all records where some value is between an upper and lower boundary.

Domain: A **domain** is a set of all unique values permitted for an attribute. The values for an attribute or a column are drawn from a set of permitted values known as a Domain. The domain of an attribute contains the set of values that the attribute may assume. Simply,

Domain is set of permissible values for each attribute. Separate attributes have separate domains. In the relation model, no two rows of relation are identical and the ordering of rows is not significant, the domain of Die is 1, 2,3,4,5 and 6. Similarly, domain of coin is Head or Tail.

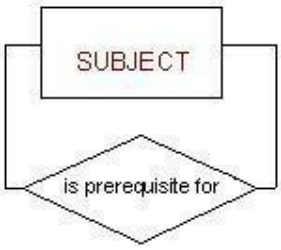
Degree of a table: Degree of a relationship type is the number of entities within each entity type that can be linked by a given relationship type. A relationship's degree indicates the number of associated entities or participants.

A **unary relationship** exists when an association is maintained within a single entity. A **binary relationship** exists when two entities are associated. A **ternary relationship** exists when three entities are associated. Although higher degrees exist, they are rare and are specifically named. Those are:

Unary relationship
Binary Relationship
Ternary relationship

A **Unary Relationship** is when both participants in the relationship are the same entity.

Example: Subjects may be prerequisites for other subjects.

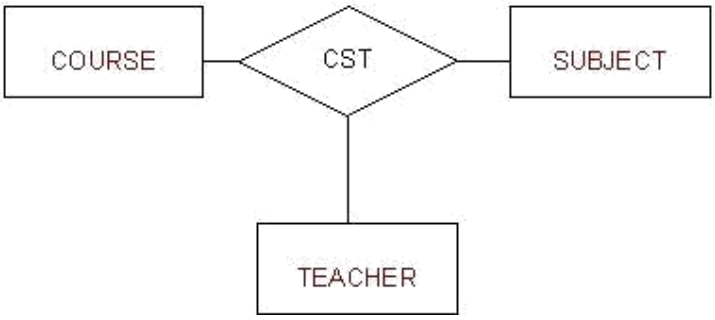


A **Binary Relationship** is when two entities participate, and is the most common relationship degree.

A **Ternary Relationship** is when three entities participate in the relationship.

Example:

The University might need to record which teachers taught which subjects in which courses.



Types of Attributes:

Attributes : Characteristics or properties of an entity is called Attribute. The values for these characteristics are called attribute value. For instance the employee entity might include the following attributes, Employee ID, Employee name etc. For each attribute, there is a set of permitted values, called the **domain,** or **value set**, of that attribute. An attribute of an entity set is a function that maps from the entity set into a domain. Since an entity set may have several attributes, each entity can be described by a set of (attribute, data value) pairs, one pair for each attribute of the entity set.

The attributes can be classified in to

Simple Attribute or Atomic Attribute
Complex/ composite attributes
Single – valued attributes
Multi - valued attributes
Derived attribute
Null Attribute
Key Attribute

An attribute, as used in the E – R model, can be characterized by the following attribute types.

Simple attributes or Atomic Attributes : These attributes have been simple; that is, they have not been divided into subparts.

Example: Assume Student is an **entity** and its **attributes** are Admission No. , Name, Age, Address and Phone no. Here the Admission No. (attribute) of student (entity) cannot further divide. In this example Admission No. is an **atomic attribute**..

Composite attributes : The attributes, which can be sub divided in to sub parts. An attribute that can be divided into smaller independent attribute is known as **composite attribute.**

Example: assume Student is an **entity** and its **attributes** are Name, Age, Address and Phone no. Here the address (attribute) of student (entity) can be further divide into House no, city and so on Student Name, Which can be divided in parts like First name, Middle name and Last name. Note also that a composite attribute may appear as a hierarchy. In the composite attribute address, its component attribute street can be further divided into street_number, street_name, and Door _ number etc.

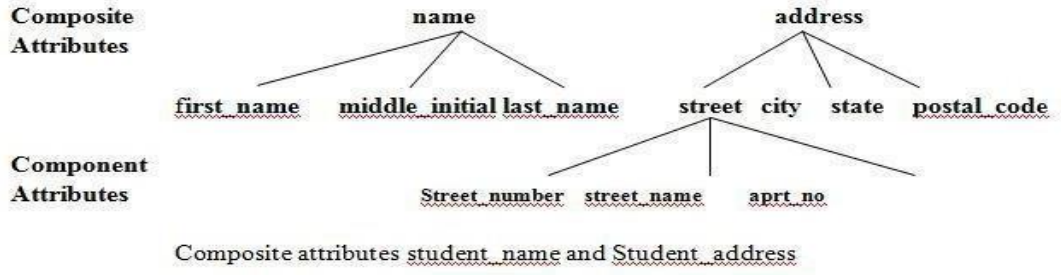


Fig 3.5.2.a

Single valued attributes: An attribute that has only single value for an entity is known as **single valued attribute.**.

Example: assume Student is an **entity** and its **attributes** are Name, Age, Sex, Address and Phone no. Here the Sex (attribute) of student (entity) can have only one value. Here, Sex is **single valued attribute**. i.e. Male or Female similarly, if marital status, Married or Unmarried.

Multi valued attributes: An attribute that can have multiple values for an entity is known as **multi valued attribute.**

Example: assume Student is an **entity** and its **attributes** are Name, Age, Address and Phone no. Here the Phone no (attribute) of student (entity) can have multiple value because a student may have many phone numbers. Here, Phone no is **multi valued attribute**.

Derived attribute: An attribute that can be derived from another attribute is known as **derived attribute.** The value of this type of attribute can be derived from the values of other related attributes or entities. The value of a derived attribute is not stored but is computed when required.

Null Attributes: An attribute takes a null value when an entity does not have a value for it. The null value may indicate "not applicable" – that is, that the value does not exist for the entity. An attribute, which has not any value for an entity is known as null valued attribute.

Key Attribute: An attribute that has unique value of each entity is known as **key attribute. E**xample, every student has unique roll no. Here roll no is **key attribute**.

Types of keys

Key plays an important role in relational database; it is used for identifying unique rows from table. It also establishes relationship among tables. Keys also help uniquely identify relationships, and thus distinguish relationships from each other. A key is an attribute allows us to identify a set of attributes in an entity .

The values of the attribute values of an entity must be such that they can **uniquely identify** the entity. In other words, no two entities in an entity set are allowed to have exactly the same value for all attributes.

Primary Key – A primary key is a column or set of columns in a table that uniquely identifies tuples (rows) in that table. It cannot contain NULL entries. The Primary key of a relational table uniquely identifies each record in the table.

In the following table, there are three attributes: **Stu_ID, Stu_Name & Stu_Age**. Out of these three attributes, one attribute or a set of more than one attributes can be a primary key.

Attribute **Stu_Name** alone cannot be a primary key as more than one students can have same name.

Attribute **Stu_Age** alone cannot be a primary key as more than one students can have same age.

Attribute **Stu_Id** alone is a primary key as each student has a unique id that can identify the student record in the table.

Table: STUDENT

Stu_Id	Stu_Name	Stu_Age
101	Naga Lakshmi	23
102	Asha	24
103	Sindhu	28
104	DhanaLakshmi	29
105	Jagadeeswari	29

Points to be remember while deciding Primary Key

We denote usually denote it by underlining the attribute name (column name).

The value of primary key should be unique for each row of the table.
The column(s) that makes the key cannot contain duplicate values.

The attribute(s) that is marked as primary key is not allowed to have null values.

Primary keys are not necessarily to be a single attribute (column). It can be a set of more than one attributes (columns). For example {Stu_Id, Stu_Name} collectively can identify the tuple in the above table, but we do not choose it as primary key

because Stu_Id alone is enough to uniquely identifies rows in a table and we always go for minimal set. Having that said, we should choose more than one columns as primary key only when there is no single column that can uniquely identify the tuple in table.

Super Key – A super key is a set of one of more columns (attributes) to uniquely identify rows in a table. For example, the 'employee_id' attribute of the entity set employee is sufficient to distinguish one employee entity from another. Thus, 'employee_id' is a super key.

Example: Employee

Emp_SSN	Emp_Number	Emp_Name
987654321	123	Jacob
101010101	127	Prasad
102020202	128	Siva
103030303	129	Venu

The above table has following super keys. All of the following sets of super key are able to uniquely identify a row of the employee table.

{Emp_SSN}
{Emp_Number}
{Emp_SSN, Emp_Number}
{Emp_SSN, Emp_Name}
{Emp_SSN, Emp_Number, Emp_Name}
{Emp_Number, Emp_Name}

Candidate Key: A super key with no redundant attribute is known as candidate key. The following two set of super keys are chosen from the above sets as there are no redundant attributes in these sets.

{Emp_SSN}
{Emp_Number}

Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is: It should not have any redundant attribute. That's the reason they are also termed as minimal super key.

Points to be remember while deciding Super key and Candidate Key

- First we have to identify that all the candidate keys are super keys. This is because the candidate keys are chosen out of the super keys.

How we choose candidate keys from the set of super keys? We look for those keys from which we cannot remove any fields. In the above example, we have not chosen {Emp_SSN, Emp_Name} as candidate key because {Emp_SSN} alone can identify a unique row in the table and Emp_Name is redundant.

Alternate Key Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.

Example

Table: Employee, this table has three attributes: Stu_Id, Stu_Number & Stu _Name.

Stu_Id	Stu _Number	Stu _Name
13648	123	Naga Lakshmi
13649	127	Sridevi
13654	128	Naga Lakshmi
13656	129	Divakar

There are two candidate keys in the above table:
{ Stu _Id}
{ Stu _Number}

DBA (Database administrator) can choose any of the above key as primary key. Lets say Stu _Id is choosen as primary key.

Since we have selected Stu _Id as primary key, the remaining key Stu_Number would be called alternative or secondary key.

Composite Key: A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.

Example

Table: sales This table has four columns (attributes) – cust_Id, order_Id, product_code & product_count.

Customer_Id	Order_Id	Product_code	Product_Count
Cus_01	Ord_01	Spinoza_combo	10
Cus_02	Ord_02	Cb885	2
Cus_03	Ord_03	Cb885	2
Cus_01	Ord_02	Lcs-8075	1

None of these columns **alone** can play a role of key in this table.

Column **cust_Id** alone cannot become a key as a same customer can place multiple orders, thus the same customer can have multiple entires.

Column **order_Id** alone cannot be a primary key as a same order can contain the order of multiple products, thus same order_Id can be present multiple times.

Column **product_code** cannot be a primary key as more than one customers can place order for the same product.

Column **product_count** alone cannot be a primary key because two orders can be placed for the same product count.

Based on this, it is safe to assume that the key should be having more than one attributes:
Key in above table: {cust_id, product_code}

Foreign Key: Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.
Example:
In the below example the Stu_Id column in Course_enrollment table is a foreign key as it points to the primary key of the Student table.

Course_enrollment table:

Course_IdSt	_Id
CSE	101
MPC	102
BIPC	101
CEC	102
HEC	103
AET	102

Student table:

Stu_Id	Stu_Name	Stu_Age
101	Prakash	22
	Chandini	26
	Mounica	25
	Hema	21

The foreign key has nothing to do with the primary key tag of another table, if it points to a unique column (not necessarily a primary key) of another table then too, it would be a foreign key. So, a correct definition of foreign key would be: Foreign keys are the columns of a table that points to the candidate key of another table.

Formal query languages

A Query Language is language in which a user requests information from the database. These languages are typically of level higher than that of a standard programming language. Languages can be classified in to the following two types.

Formal Query language
Commercial Query language

Formal Query Languages are formal in the sense that they are lack of 'syntactic behavior 'of commercial query languages. These can be used for extracting data from the database. These can be further classified as procedural or non-procedural.

In Procedural Language, the user instructs the system, to perform a sequence of operations on the database to compute the desired result. In a non- procedural language, the user describes the information desired without giving a specific procedure for obtaining that information.

Some of the formal Query Languages are listed below:

The Relational Algebra
Tuple Relational Calculus
Domain Relational Calculus

The Relational Algebra: The relational algebra is a formal Procedural Query language. It consists of a set of operations that take one or two relations as input and procedure a new relation as their result. The fundamental operations in the relational algebra are SELECT, PROJECT, UNION, SET DIFFERENCE, CARTESIAN PRODUCT AND RENAME, etc. In addition to other operations namely, SET INTERSECTION, NATURAL JOIN, DIVISION and ASSIGNMENT.

The SELECT, PROJECT and RENAME operations are called UNARY Operations, because they operate on ONE REALTION. The Other operations operate on pairs (two or more) of relations and are called BINARY OPERATIONS.

Tuple Relational Calculus (TRC): Tuple relational calculus is used for selecting those tuples that satisfy the given condition.

Table: Student

First Name	Course	Age
Tulasi	**M.P.C.**	**26**
Lakshmi	**M.P.C.**	**24**
Vijaya	**M.P.C.**	**23**
Samitha	**C.E.C**	**18**
Parvathi	**C.S.E.**	**19**

Write relational calculus query..

Query to display the Course of those students where age is greater than 20 { t.Last_Name | Student(t) AND t.Course > 20 }

In the above query you can see two parts separated by | symbol. The second part is where we define the condition and in the first part we specify the fields which we want to display for the selected tuples.

The result of the above query would be:

course

M.P.C.

Query to display all the details of students where course is '**M.P.C.**' { t | Student(t) AND t. course = ' **M.P.C.**' }

Output:

First Name	course	Age
Tulasi	**M.P.C.**	**26**
Lakshmi	**M.P.C.**	**24**
Vijaya	**M.P.C.**	**23**

Domain Relational Calculus (DRC): In domain relational calculus the records are filtered based on the domains. Again we take the same table to understand how DRC works. Table: Student

First Name	Course	Age
Tulasi	**M.P.C.**	**26**
Lakshmi	**M.P.C.**	**24**
Vijaya	**M.P.C.**	**23**
Samitha	**C.E.C**	**18**
Parvathi	**C.S.E.**	**19**

Query to find the first name and course of students where student age is greater than 20 {< First_Name, Age > | ∈ Student ∧ Age > 20}

Note: The symbols used for logical operators are: ∧ for AND, ∨ for OR and ¬ for NOT.

Output:

First Name	course
Tulasi	**M.P.C.**
Lakshmi	**M.P.C.**
Vijaya	**M.P.C.**

Selection Operation:

Consider the following relation: teach

Name	Address	Subject
Bheemeswara Rao	**Visakhapatnam**	**ISA**
Prasad	**Rajahmundry**	**DBMS**
Sastry Tanikella	**Kothapeta**	**Data Mining**
Suresh	**Rajahmundry**	**Data Ware Housing**
Jaya Prakash	**Nidadavole**	**DBMS**

This is used to fetch rows (tuples) from table(relation) which satisfies a given condition. **Syntax:** $\sigma_F(R)$

Where, σ (sigma) represents the Select Predicate, R is the name of relation (table name in which you want to look for data), and F is the prepositional logic, where we specify the conditions that must be satisfied by the data. In prepositional logic, one can use **unary** and **binary** operators like =, <, >, <=, >=, != etc, to specify the conditions.

To answer the query: is the teacher with name Prasad teaching the course DBMS? We may write

NAME = "PRASAD" subject = "DBMS" (subject)

F(R) is the set of those tuples of R compoments of each of which make the logical expression F True.

Projection Operation: With the help of this operation, any number of columns can be omitted from a table or columns of the table can be rearranged.

Consider the query: print the names and addresses of teachers who are teaching the DBMS course.

course = "DBMS." (teach)

The result is the following new relation

Name AddressSubject

Prasad Rajahmundry DBMS Jaya Prakash Nidadavole DBMS

If we want to get only name and addresses of those teachers, we may write

1.2 = σ course = "DBMS." (teach)

Where π is the projection operation symbol and the subscripts indicate the positions of the attributes of the table.

This operation when performed produces the following relation

Name Address

Prasad Rajahmundry

Jaya PrakashNidadavole

Note: By Default projection removes duplicate data.

Set Union Union operation in relational algebra is same as union operation in set theory, only constraint is for union of two relation both relation must have same set of Attributes.

Let 'n' be the number of columns of the relations R and S, Union of R and S is denoted as R + S. It is the set of tuples, with number of columns n, which are either in R, or in S, or in both R and S.

Example:

Set: R

x	**y**	**z**
p	**z**	**x**
y	**r**	**q**

Set: S

y	**z**	**x**
z	**p**	**r**
y	**r**	**q**

Then R +S =

x	**y**	**z**
p	**z**	**x**
y	**r**	**q**
y	**z**	**x**
z	**p**	**r**

Set Difference : Set Difference in relational algebra is same set difference operation as in set theory with the constraint that both relation should have same set of attributes.

It is denoted by R – S and consists of the set of tuples which are in R but not in S.

R – S =

x	**yz**
p	**zx**

Cartesian Product : Cartesian product between two relations let say A and B, so cross product between A X B will results all the attributes of A followed by each attribute of B. Each record of A will pairs with every record of B.

It is denoted by R X S. If 'R' and 'S' have number of columns 'r' and 's', respectively, then R X S is the set of tuples with arity (number of columns) r + s, whose first r components are composed of the tuples in R and remaining s components composed of the tuples in S.

Example: R X S =

x	**y**	**z**	**y**	**z**	**x**
x	**y**	**z**	**z**	**p**	**r**
x	**y**	**z**	**y**	**r**	**q**
p	**z**	**x**	**y**	**z**	**x**
p	**z**	**x**	**z**	**p**	**r**

p	**z**	**x**	**y**	**r**	**q**
y	**r**	**q**	**y**	**z**	**x**
y	**r**	**q**	**z**	**p**	**r**
y	**r**	**q**	**y**	**r**	**q**

Note: if R has 'n' tuples and S has 'm' tuples then R X S will have 'n*m' tuples.

Set Intersection: Intersection operator is denoted by ∩ symbol and it is used to select common rows (tuples) from two tables (relations).

Let's say we have two relations R1 and R2 both have same columns and we want to select all those tuples(rows) that are present in both the relations, then in that case we can apply intersection operation on these two relations R1 ∩ R2.

Note: Only those rows that are present in both the tables will appear in the result set.

Syntax of Intersection Operator (∩)

Ex: table_name1 ∩ table_name2

Table 1: COURSE

course_id	student_name	Student_id
CSE	Jacob	cs_101
CSE	Divakar	cs_102
MPC	Prakash	mp_101
CEC	Reddy	ce_102
HEC	Veera	he_108
BPC	Tarun	bi_103

Table 2: STUDENT

Student_id	student_name	student_age
cs_101	Jacob	16
cs_102	Divakar	17
mp_101	Prakash	18
ce_102	Reddy	16
he_108	Veera	16
bi_101	Naresh	16
bi_102	Naga lakshmi	17
bi_103	Tarun	17

π student_name (COURSE) ∩ π student_name (STUDENT)

output:

student_name
Jacob
Divakar
Prakash
Reddy
Veera
Tarun

Natural Join ( ): A NATURAL JOIN is a JOIN operation that creates an implicit join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables.

A NATURAL JOIN can be an INNER join, a LEFT OUTER join, or a RIGHT OUTER join. The default is INNER join.

If the SELECT statement in which the NATURAL JOIN operation appears has an asterisk (*) in the select list, the asterisk will be expanded to the following list of columns (in this order):

All the common columns
Every column in the first (left) table that is not a common column
Every column in the second (right) table that is not a common column

An asterisk qualified by a table name (for example, COUNTRIES.*) will be expanded to every column of that table that is not a common column.

If a common column is referenced without being qualified by a table name, the column reference points to the column in the first (left) table if the join is an INNER JOIN or a LEFT OUTER JOIN. If it is a RIGHT OUTER JOIN, unqualified references to a common column point to the column in the second (right) table.

Syntax

TableExpression **NATURAL [{ LEFT | RIGHT } [OUTER] | INNER] JOIN**
{ TableViewOrFunctionExpression **| (** TableExpression **) }**

Examples

If the tables COUNTRIES and CITIES have two common columns named COUNTRY and COUNTRY_ISO_CODE, the following two SELECT statements are equivalent:

SELECT * FROM COUNTRIES NATURAL JOIN CITIES

SELECT * FROM COUNTRIES JOIN CITIES USING (COUNTRY, COUNTRY_ISO_CODE)

The following example is similar to the one above, but it also preserves unmatched rows from the first (left) table:

SELECT * FROM COUNTRIES NATURAL LEFT JOIN CITIES

Division (÷)*:* Division operator A÷B can be applied if and only if:

Attributes of B is proper subset of Attributes of A.

The relation returned by division operator will have attributes = (All attributes of A – All Attributes of B)

The relation returned by division operator will return those tuples from relation A which are associated to every B's tuple.

Table: STUDENT_SPORTS

Roll_number	Sports
1	Badminton
2	Cricket
3	Badminton
4	Badminton

Table: ALL _SPORTS

SPORTS
Badminton
Cricket

Consider the relation STUDENT_SPORTS and ALL_SPORTS given above.

To apply division operator as

STUDENT_SPORTS (÷) ALL_SPORTS

The operation is valid as attributes in ALL_SPORTS is a proper subset of attributes in STUDENT_SPORTS.

The attributes in resulting relation will have attributes {ROLL_NO,SPORTS}-{SPORTS}=ROLL_NO

The tuples in resulting relation will have those ROLL_NO which are associated with all B's tuple {Badminton, Cricket}. ROLL_NO 1 and 4 are associated to Badminton only. ROLL_NO 2 is associated to all tuples of B. So the resulting relation will be:

output:

ROLL_NO

2

Assignment: The assignment operation (←) provides a convenient way to express complex queries.

Write query as a sequential program consisting of
a series of assignments
followed by an expression whose value is displayed as a result of the query.

Assignment must always be made to a temporary relation variable.

Tuple Relational Calculus: Tuple Relational Calculus is a non-procedural query language unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it.

Thus, it explains what to do but not how to do. In Tuple Calculus, a query is expressed as

{ t | P (t) } or{t | condition (t)}

where 't' = resulting tuples,
P(t) = known as Predicate and these are the conditions that are used to fetch 't'

Thus, it generates set of all tuples 't', such that Predicate P(t) is true for 't'.

P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).

It also uses quantifiers:
t ∈ r (Q(t)) = "there exists" a tuple in t in relation r such that predicate Q(t) is true. ∀ t ∈ r (Q(t)) = Q(t) is true "for all" tuples in relation r.

Domain relational **calculus: Domain relational** calculus uses list of attribute to be selected from the relation based on the condition. It is same as **Tuple Relational Calculus** but differs by selecting the attributes rather than selecting whole tuples.

{<EMP_ID, EMP_NAME> | <EMP_ID, EMP_NAME> ? EMPLOYEE Λ DEPT_ID = 10}

Commercial query languages: Commercial Query Languages are needed for the commercial database systems. These languages are more user-friendly. Commercial query languages, besides querying a database, includes features for defining the structure of the data, for modifying data in the database and for specifying security constrains.
Some of the commercial Query languages are listed below

SQL (Structured Query Language)
QBE (Query – by – Example)
Quel (Query Language)

Commercial Query Languages possess such much "syntactic behavior 'compared to formal query languages. Commercial data languages include construct for update, Insertion and deletion of information as well as for quires to the data bases.

QBE is both a query language and the name of a DB system including it. The system is no longer in use, but the language is part of IBM's Query Management Facility (QMF)

Basic Structure

QBE has ``two-dimensional'' syntax.
Queries are expressed by example.

Close correspondence with domain relational calculus

Non-procedural.

Queries are expressed using **skeleton tables**.

User selects the skeletons needed.

User fills in skeletons with example rows.

An example row consists of constants and example elements which are really domain variables.

Domain variables are preceded by an underscore character.

Constants appear without any qualification.

Differences between formal and commercial query languages

Formal Query Language	Commercial Query Language
1. These languages lack of syntactic behavior of other languages. 2. These are not used by commercial database systems. 3. The user can query the tables,insert new tuples (rows in table) delete tuples and update/modify tuples, usingthese languages. Examples: Relational Algebra Tuple Relational Calculus Domain Relational Calculus	1. These languages posses syntactic behavior like high level languages. 2. These are the languages used by the commercial database systems. 3. These languages are used to query a database, defining the structure of data modifying data for specifying security constraints. Examples : SQL, QBE, Quel

CODD rules

EF formulated rules for the RDBMS which are 12 in number an in addition to these rules there is an another rule which is called rule " ZERO". Thus making the count of the rules to 13, based upon we can decide a DBMS to be an RDBMS or not.

0 – Single Foundation Rule
1 – Information Rule
2 – Guaranteed Access
3 – Systematic Treatment of NULL values.
4 – Active online catalogue
5 – Comprehensive data sublanguage.
6 – View updation Rule
7 – High level UPDATE, INSERT, DELETE
8 – Physical Data Independence
9 – Logical Data Independence
10 – Integrity Independence
11 - Distribution Independence
12 – Non – Subversion

Rule 0 : Single Foundation Rule: RDBMS must manage every aspect of the database entirely through itself, using its relational capabilities without using any external language. The processing should not be done with external storage.

Rule 1 : Information Rule: The information can be represented in one and only one way that is tables also known as relations. This rules emphasis fact that the information can be stored in rows and columns. The never the data as to be stored are represented, it has to be in tables form only as it gives easy and flexibility.

Rule 2 : Guaranteed Access: The access to table follows the sequence of table name, tuple attribute. This also states that at the intersection or each column and row there should be one an only one value a data. The value of a data must be logically addressable by using combination of data name, column name and value. Every step of data access is identified where the access to the work area involve qualifying the user name and password.

Rule 3 : Systematic Treatment of NULL values: One should be able to operate with NULL values. The operation with the NULL values should be performed using single command. The treatment of NULL should be independent of data type. NOT NULL value should be provided if gives of these NULL values are included there should force no problems for applications using and manipulating them.

Rule 4 : Active On – Line catalogue: RDBMS should maintain data dictionary tables to keep track of current state of the database. These are special tables which keep track of the current state of the database. These tables contains information about table techniques, views column definitions synonyms and every other type of objects updated automatically.

Rule 5 : Comprehensive Data Sub Languages: RDBMS should have comprehensive Data Definition Language, Data Control Language, Data Manipulation Language. All the operation on database should be supported by the data language which is path and parcel of the package.

Rule 6 : View Updation Rule: One can feel that any view can be updated, but in real practice one cannot update all views as some views are based on aggregating and virtual columns which make them impossible to be update.

Rule 7 : High Level UPDATE, INSERT, DELETE: An RDBMS must be capable of doing more than just retrieving the relational data. It should able to do insert, Update and Delete data items with the use of single command for each operation. It should able to do this operation are more than one row also.

Rule 8 : Physical Data Independence: The recording on the data should be left to the description of the type of system used that is whenever data is retrieved it should be independent of the storage structure and representation a change in the storage strategy should no effect the performance of data.

Rule 9 : Logical Data Independence: The data should be independent of logic involved in programming if at all the database design is changed then the programs should be independent of these changes. That is once the programs are committed by effort should not effect by this effort should not be effect should logic depending level.

Rule 10 : Integrity Level Independence: The data available should force no limitations in terms of integration. The limits of the system used should only be taken into consideration. The integrity constraints specific to particular relational database must definable in relational data sub-language and storage in the data dictionary not in application programs.

Rule 11 : Distribution Independence: This is one of the most popular and important aspect for any RDBMS. This implies fact the system should look like a centralized system to the user even if it is distributed across. It is important as the user should face difficulties. While accessing data, if it is at remote server the RDBMS should not take care of it imposing no problems to the user.

Rule 12 : Non – Subversion Rule: Incase if RDBMS users help of any low language it must not bypass any integrity rules or constraints of the relational language thus any operation must be governed by the relation rules.

Normalization: The basic objectives of normalization is to reduce redundancy which means that information is to be stored only one storing information or data several times leads to wastage of storage space and increase on the total of the data stored.

Types of Normal forms:

1) First Normal Form 2) Second Normal Form 3) Third Normal Form
4) Boycee codd Normal Form 5) Fourth and Fifth Normal Forms

Quick Reference

A **Relational database** is a database that has a collection of tables of data items, all of which is formally described and organized according to the relations

A **domain** is a set of all unique values permitted for an attribute.

"Row of relation (Table) is referred as **tuple**. Tuple having a set of 'n' numbers of attributes are called as n-tuple.

A **range** is a common database operation that retrieves all records where some value is between an upper and lower boundary.

A **super key** is a set of one of more columns (attributes) to uniquely identify rows in a table.

A super key with no redundant attribute is known as **candidate key**.

A **primary key** is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.

Formal Query Languages are formal in the sense that they are lack of 'syntactic behavior 'of commercial query languages. These can be used for extracting data from the database. These can be further classified as procedural or non-procedural.

Some of the formal Query Languages are listed below:

The Relational Algebra
Tuple Relational Calculus
Domain Relational Calculus

Commercial Query Languages are needed for the commercial database systems. Commercial query languages, besides querying a database, includes features for defining the structure of the data, for modifying data in the database and for specifying security constrains.

Some of the commercial Query languages are listed below

SQL (Structured Query Language)
QBE (Query – by – Example)
Quel (Query Language)

Commercial Query Languages possess such much "syntactic behavior 'compared to formal query languages. Commercial data languages include construct for update, Insertion and deletion of information as well as for quires to the data bases.

The fundamental operations in the relational algebra are SELECT, PROJECT, UNION, SET DIFFERENCE, CARTESIAN PRODUCT AND RENAME, etc. In addition to other operations namely, SET INTERSECTION, NATURAL JOIN, DIVISION and ASSIGNMENT.

The SELECT, PROJECT and RENAME operations are called UNARY Operations, because they operate on ONE REALTION.

UNION, SET DIFFERENCE, CARTESIAN PRODUCT, etc operations operate on pairs (two or more) of relations and are called BINARY OPERATIONS.

A key is an attribute allows us to identify a set of attributes in an entity .

Short Answer Type Questions

What is a Relational Database?
What is Domain in Relational Table
What is Degree of a table in Relational Model
What is Tuple?
What is super key?
What is Candidate key?
What is Primary Key?
What are the Formal Query Languages?
What are the Commercial Query Languages?
What are fundamental operations in Relational Algebra?
What are Unary operations?
What are Binary operations?

Long Answer Type Questions

Explain about Relational Data Model
What is an attribute? Write about types in it.
What is Key? Write about types of Keys.
Write the Differences between Formal and Commercial Languages.
What are the Codd rules in Relational Model.
Write short notes on Fundamental Operations in Relational Algebra

UNIT 4

S Q L

Structure

Introduction to SQL
data types
DDL,DML and DCL Commands
Data Constraints
Set operators and joins
Sub queries and database objects.

Learning Objectives

After studying this unit, student will able to learn

Different types of data types in SQL.
Understand DDL, DML and DCL commands.
Understand different set operator and joins.

Introduction to SQL: The name SQL, pronounced as 'ess – cue – ell' or 'sequel', is the abbreviation for Structured Query Language. The SQL consists of a asset of facilities for defining, accessing and managing relational database. In order to understand why SQL has become so popular and important, it is helpful to have an idea of the major developments in database technology over the past 20 to 30 years.

In the 1970s, SQL was developed by the Raymond FF. Boyce and Donald D. Chamberline at IBM with the name of SEQUEL. It was designed for manipulating and retrieving data stored in the original quasi-relational database management system of IBM. However, the first commercial implementation of SQL was introduced in June 1979 by Relational Software for VAX computers.

SQL stands for Structured Query Language. IBM was able to demonstrate SQL, which could be used to control relational database.

The SQL implemented by ORACLE Corporation is 100% complaint with the ANSI/ISO standard SQL data language. Oracle's database language is SQL, which is used for storing and retrieving information in Oracle. A table is a primary data base object of SQL that is used to store data. A table holds data in the form of row and columns.

SQL is used as their standard database language by all the relational database management systems like Oracle, Informix, Posgres, SQL server, MySQL, MS Access, and Sybase.

To communicate with the database, SQL supports the following commands

Data definition language : Create, alter, drop, grant, and revoke commands.

Data manipulation language : Insert, select, delete and update commands.
Transaction control language : Commit, save point and rollback commands.

Advantages of SQL:

Non procedural language, because more than one record can be accessed rather than one record at a time.
It is the common language for all relation database (i.e) portable and it requires only small modification to make use of in other database.
Very simple commands for querying , inserting, deleting and modifying data and objects.

High speed: Using the SQL queries, the user can quickly and efficiently retrieve a large amount of records from a database.

No coding needed: In the standard SQL, it is very easy to manage the database system. It doesn't require a substantial amount of code to manage the database system.

Well defined standards: Long established are used by the SQL databases that are being used by ISO and ANSI.

Portability: SQL can be used in laptop, PCs, server and even some mobile phones.

Interactive language: SQL is a domain language used to communicate with the database. It is also used to receive answers to the complex questions in seconds.

Multiple data view: Using the SQL language, the users can make different views of the database structure.

Disadvantages of SQL

Along with some benefits, the Structured Query Language also has some certain disadvantages:

Difficult Interface: SQL has a complex interface that makes it difficult for some users to access it.

Partial Control: The programmers who use SQL doesn’t have a full control over the database because of the hidden business rules.

Implementation: Some of the databases go to the proprietary extensions to standard SQL for ensuring the vendor lock-in.

Cost: The operating cost of some SQL versions makes it difficult for some programmers to access it.

data types:

Following are the various data types, recognized by the Oracle engine and permitted to be used for creating an Oracle table column

CHAR (size): This data type is used to store character strings values of fixed length. The size in brackets determines the number of characters the cell can hold. The maximum number of characters. This data type can hold is 255 characters. ORACLE compares CHAR values using blank – padded comparison semantics i.e. if a value that is inserted in a cell of CHAR data type is shorter than the size it is defined for then it will be padded with spaces on the right until it reaches the size characters in length. In case if the use enters a value larger than the specified length then the database would return an error.

VARCHAR (size) (Or) VARCHAR 2(size): The data type is used to store variable length alphanumeric data. The maximum this data type can hold is 2000 charcters. One difference between this data type and the **CHAR** data type is ORACLE compares VARCHAR values using non – padded comparison semantics i.e., the inserted values will not be padded with spaces.

NUMBER(P,S): The NUMBER data type is used to store numbers (fixed or floating point). Numbers of virtually any magnitude may be stored up to 38 digits of precision. Numbers as large as 9.99 * 10 to the power of 124, i.e. 1 followed by 125 zeros can be stored. The precision ,(P) determines the maximum length of the data, whereas the scale, determines the number of places to the right of the decimal if scale is omitted then values are stored with their original precision upto the maximum of 38 digits.

DATE: This data type is used to represent date and time. The standard format is DD – MON – YY as an 21 – MAR – 19. To enter dates other than the standard format, use the appropriate functions. Date Time stores date in the 24 – hour format. By default, the time in a date field is 12:00:00 am, if no time portion is specified. The default date for a date field is the first day of the current month.

LONG: This data type is used to store variable length character string containing upto 2GB. LONG data can be used to store arrays of binary data in ASCII format. LONG values cannot be indexed, and the normal character functions such as SUBSTR cannot be applied to LONG values.

RAW / LONG RAW: the RAW / LONG RAW data types is used to store binary data such as digitized picture or image. Data loaded into columns of these data types are stored without any further conversion. RAW data type can have a maximum length of 255 bytes. LONG RAW data type can contain up to 2GB. Values stored in columns having LONG RAW data type cannot be indexed.

TWO DIMENSION MATRIX CREATION:

One needs to communicate with the Oracle engine to create a two dimensional matrix (i.e., for the storage and manipulation of data. This must be done using the natural language of the Oracle Engine i.e. SQL.

Using an SQL sentence we can command the engine to create a table with cell parameters of our choice. Hence, it is necessary to understand how to construct standard SQL sentences. A look at a generic SQL sentence will help understand SQL syntax.

The Generic SQL Sentence Construct.

Verb:

[Clause 1] < para 1:1 para 1:2 para 1:n>,

[Clause 2] < para 2:1 para 2:2 para 2:n>,….. [Clause n];

THE GRAMMAR OF SQL SYNTAX:

An SQL statement starts with a Verb, This verb may have additional nouns and adjectives.

Each verb is followed by a number of clauses.

Each clause has one or more parameters. Beyond this no further decomposition is allowed. Ie.e.parameters cannot in turn have sub – parameters.

A space separates clauses within an SQL statement.

A comma separates parameters within clause.

A semi colon is used to terminate the SQL statement.

DDL,DML and DCL Commands

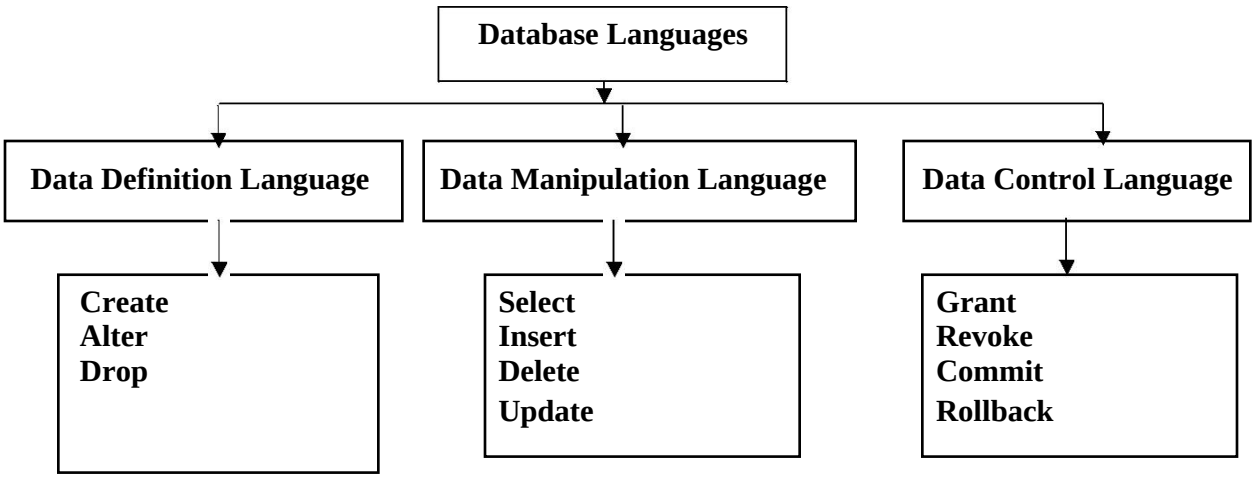


Fig.4.3

A DBMS must provide appropriate languages and interfaces for each category of users to express database queries and updates.. Database languages are used to read, update, store and maintain data in a database on computer. There are several such languages that can be used for this purpose; one of them is SQL (Structured Query Language).

A database system provides three different types of Languages, one will specify the schema, and other will express database queries, updates and maintenance.

Those are

Data-Definition Languages (DDL)
Data-Manipulation Language (DML)
Data – Control Language (DCL)

Data-Definition Languages (DDL):

DDL is used for specifying the database schema. It is a language that allows the users to define data and their relationship to other types of data. It is also used to specify the structure of each table, set of associated values with each attribute, integrity constraints, security and authorization information for each table and physical storage structure of each table on disk.

The operations that we can perform on database using DDL:

To create the database instance
To alter the structure of database
To drop database instances

The result of compilation of DDL statements is a set of tables that is stored in a special file called ‘Data dictionary’ or “data directory.

A data dictionary is a file that contains metadata, i.e. Data about data. This file is consulted before actual data are read or modified in the database system.

The storage structure and access methods used by the database system are specified by a set of definitions in a special type of DDL called a ‘data storage and data definition language’. The result of consultation of these definitions is a set instruction to specify the implementation details of the database schemas. Which are usually hidden form the users.

The DDL commands are

CREATE
ALTER
DROP

CREATE: To create a new database table.

Syntax: **CREATE** table <table_name> (column_name(1) data_type(size)**,** column_name(2) data_type(size)**, ...**column_name(n) data_type(size))**;**

Example: SQL> CREATE table emp (emp_id varchar2(10), emp_name varchar2(20), emp_sal varchar2(10));

In a table

We should specify a unique column name.
We should specify proper datatype along with its width.
We can include "Not null' condition when needed, by default it is 'Null'.
While naming a table the first letter should be an alphabet.
Oracle reserved words cannot be used to name a table.
Maximum length for table name is 30 characters.
Two different tables should not have the same name.
Underscore, numerals and letters are allowed but not blank space and single quotes.

ALTER: ALTER command is used to change or alter the properties of a database table.

Syntax: ALTER table <table_name> ADD (new_column_name data_type(size));

ALTER table <table name> modify (column definition.);

Example: SQL> ALTER table student add (group_name varchar2(10));

We proceed in the following way to modify the width of an existing column emp_id from varchar2 (10) to varchar2 (12) belonging to customer table.

Example

SQL > alter table emp modify (emp_id varchar2(12));

The following example explains how to add a column type long to the customer table.

Alter table command needed in the following situations

When a user wants to add a new column.
When user wants to modify the existing column definition say from.
For Defining Integrity Constraints.
Null to Not null or to change the width of the datatype or datatype itself.
To include to drop integrity constraints (integrity constraints are dealt in the later part of this session).

Using the ALSTER TABLE clause the following tasks cannot be performed:

Change the name of the table.
Change the name of the column.
Drop a column.
Decrease the size of a column if table data exists.

Defining Integrity Constraints In The Alter Table Command

We can also define integrity constraints, using the constraint clause, in the **ALTER** table command. Oracle will not allow constraints defined using **ALTER TABLE,** to be applied to the table if data in the table violates such conditions.

If a primary constraint was being applied to a table in retrospect and the column has duplicate values in it, the Primary key constraint will not be set to that column.

Examples:
Adding PRIMARY KEY data constraint on the column emp_id belonging to the table emp.

ALTER TABLE emp ADD PRIMARY KEY emp_id;

Add FOREIGN KEY constraint:

ALTER TABLE *name*
ADD CONSTRAINT *column_name*
FOREIGN KEY *column_name*
ON DELETE *action_type*
ON UPDATE *action_type*

Dropping Integrity constraints in the alter table command:

We can drop an integrity constraint if the rule that it enforces is no longer true or if the constraint is no longer needed. Drop the constraint using the ALTER TABLE command with the DROP clause.

Examples for dropping integrity constraints: Drop
the PRIMARY KEY constraint from a table. ALTER
TABLE <table_name> DROP PRIMARY KEY;

Drop FOREIGN KEY constraint on a column in a table.
ALTER TBALE <table_name> DROP CONSTRAINT <reference>;

Note: Dropping UNIQUE and PRIMARY KEY constraints drops the associated indexes.

DROP: DROP command is used to **remove or delete** a table from the database.

Syntax: DROP table <table_name>;
Example:SQL> DROP table student ;

If this statement is successfully executed then the massage 'table dropped' would be displayed.

The following command would delete all the record from the table.

Command to truncate a table

Truncate table <table-name>;

Example
SQL > truncate table student;

Implementation of this command would delete all the rows associated with the table, only the structure of the table remains.

Examining Objects created by a user:

To find the table or tables create by the user
Syntax: SELECT * FROM TAB**;**

To find the column details of a table created:
Syntax: DESCRIBE table_name;
(OR)
DESC table_name**;**

Data-Manipulation Language (DML)

A data manipulation language (DML) is a language permit the users to access and manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

The operations that we can perform on database using DML:

Retrieving data from a database
Inserting data into a table
Updating existing data within a table
Delete all records from a database table
UPSERT operation (insert or update)
call a PL/SQL or Java subprogram
interpretation of the data access path
concurrency Control

It includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

A DML is language which enables users to access or manipulate data. There are basically two types.

Procedural DML : This requires a user to specify what data are needed and how to get those data from existing database.
Non procedural DML : Which require a user to specify what data are needed 'without' specifying how to get those data.

Nonprocedural DMLs are usually easier to learn and use than procedural DMLs. A user does not have to specify how to the data, these languages may generate code that is not as that produced by Procedural DML. Hence we can make remedy this difficulty by various optimization techniques.

A Query is a statement, a request for retrieval information. The portion of a DML, that involves information retrieval is called a 'Query Language'.

DML is used to perform typical manipulations like retrieval insertion and deletion and modification of the data.

The DML commands are

INSERT
DELETE
UPDATE
SELECT

INSERT: INSERT command is used to insert data values in to a database table.

Syntax: INSERT into table_name (coumn_name, column_name) values (expression) **;**

Example:
SQL> INSERT into emp (emp_id, emp_name, emp_sal) values (E1001, 'sastry', 2000)**;**

DELETE: DELETE command is used to delete the data from the database table.

Syntax: DELETE from table_name WHERE <search condition> **;**
Example: DELETE from emp WHERE emp_sal < 5000;

UPDATE: UPDATE" command is use to change or modify database table. We can perform UPDATE operation in two ways.

To update all the rows / records of a table.
To update a single or set of records of a table.

Syntax:
UPDATE table_name SET column_name = expression, column_name = expression
WHERE column_name = expression;

Example: SQL> UPDATE emp SET sal = 3200 WHERE ename = 'smith';

SELECT: Once data has been inserted into a table, the next most logical operation would be to view that has been enter. The "SELECT" command is used to achieve this. "SELECT" command is used to retrieve data from a database table. We can SELECT the data from a table for two purposes.

To select all rows / records of a table.
To select specific records of a table.

To select all rows / records of a table:

Syntax:SELECT * FROM <table_name>;
Example: SQL> SELECT * from emp;

To select specific records of a table.

Syntax: SELECT <column_name 1> **,** …..**,** <column_name n> FROM <table_name> WHERE search_condition;

Example: SQL> SELECT epic , emp_name from employee WHERE dept='production' **;**

Data – Control Language (DCL) & Transaction Control Language (TCL)

A data Control language (DCL) is a language is used for granting and revoking user access on a database. The changes in the database that we made using DML commands are either performed or rollbacked using TCL. This is used to control the kind of access to the database and also for transaction control.

To grant access to user.
To revoke access from user.
To persist the changes made by DML commands in database.
To rollback the changes made to the database.

The DCL commands are

GRANT
REVOKE
COMMIT
ROLLBACK
SAVE POINT

GRANT: GRANT command is used to provide authorization or access permission to a particular database user.The database administrator can only use this GRANT command.

Syntax: GRANT <PERMISSION_TYPE > to <USER_NAME> **;**
Example: GRANT dba to scott**;**

REVOKE: REVOKE command is used to revoke object privileges that the user previously granted directly to revoke.

Syntax: REVOKE { object privileges} ON object_name FROM user_name**;**

COMMIT: This command is used to save the modifications performed on a database table.

Syntax: COMMIT**;**
Example: commit**;**

ROLLBACK: "ROLLBACK" command is used to take the database to its previous stage. If we are made any modifications performed on the database.

Syntax: ROLLBACK**;**
Example: ROLLBACK;

SAVE POINT: "SAVE POINT" is used to rollback portions of the crrent set of transactions.

Syntax TO SET THE SAVEPOINT: SAVEPOINT expression**;** ROLLBACK to SAVEPOINT CONDITION

Data Constraints:

There are two types of data constraints that can be appled to data being inserted Oracle table. One type of constraint is called an i/o constraint (input / output). This data constraint determines the speed at which data can be inserted or extracted from an Oracle table the other type of constraint is called a business rule constraint.

I / O Constraints: The input / output data constraint, is further divided into two distinctly different constraints.

The Primary key Constraint: Here the data constraint attached to a table column (or Columns) ensures.

That the data entered in the table column (or Columns) is Unique across the entire column (or Columns).

That none of the cells belonging to the table column (or columns) or left empty. **The Foreign Key Constraint:** This constraint establishes a relationship between records (i.e. data) across a **Master** and a **detail** table. This relationship ensures.

Records cannot be inserted into a **detail** table if corresponding records in the **master** table do not exist.

Records of the **master** table cannot be deleted if corresponding records in the **detail** table

Oracle has a NOT NULL and UNIQUE as column constraints. The NOT NULL column constraint ensures that a table column cannot be left empty. The UNIQUE constraint ensures that the data across the entire table column is unique (it means no repeating values can be inserted into the column).

The Unique column constraint permits multiple entries of the NULL values into the column. These NULL values are clubbed at the top of the column in the order in which they were entered into the table. This is the essential difference between the Primary Key and the Unique constraints when applied to a table column(s).

Business Rule constraints: Oracle allows the application of **business rules** to table column. Business managers determine business rules. These rules are applied to data prior the data being inserted into table columns. This ensures that the date (records) in the table have integrity. Thus, business processes initiated after business managers analyze this data will result in business growth.

Business rules can be implemented in Oracle by using CHECK constraints. Business rule validation checks are performed when the user performs a write operation on the table that is **insert, update** or **delete** data. Any insert, delete or update statement causes the relevant constraint to be evaluated. The constraint must be satisfied for the statement to succeed. Thus these constraints ensure integrity of the data in the tables.

Constraints can be connected to a **column** or a **table** by the CREATE TABLE or ALTER TABLE command. Constraints are recorded in Oracle's data dictionary.

Conceptually, Data constraints are connected to a column, by the Oracle engine, as **flags.** Whenever user attempts to load the column with data, the Oracle engine will observe

the flags and recognize the presence of constraints. Then, the Oracle engine will apply the defined constraint, to the data being entered.

If the data being entered into a column fails any of the data constraint checks, the entire record is rejected. The Oracle engine will then flash an appropriate **error message** to the user.

Oracle allows programmers to define constraints at:

Column Level
Table Level

Column Level Constraints: If data constraints are defined along with the column definition when creating or altering a table structure, they are **column level** constraints. Column level constraints are applied to the **current column.** The current column is the column that immediately **precedes** the constraint. It means, They are local to a specific column.

A column level constraint **cannot** be applied if the data constraint spans across multiple columns in a table.

Table Level Constraints:

If data constraints are defined after defining all the table columns when creating or altering a table structure, it is a **table level** constraint. Table level constraint **must** be applied if the data constraint spans across multiple columns in a table.

Constraints are stored as part of the global table definition by the Oracle engine in its **system tables.**

The SQL syntax used to attach the constraint will change depending upon whether it is a column level or table level constraint.

NULL value concepts: Often there may be records in a table that do not values for every field , either because the information is not available at the time of data entry or because the field is not applicable in every case. If the column was created as NULLABLE (i.e. the default column construction of Oracle), Oracle will place a NULL value in the column in the absence of a user defined value.

A **NULL** value is different from a blank or a zero. **NULL** values are treated specially by Oracle. A **NULL** value can be inserted into the columns of any data type.

Principles of NULL values:

Setting a NULL value is appropriate when the actual value is unknown, or when a value would not be meaningful.
A **NULL** value is not equivalent to a value of zero if the data type is number and spaces if the data type is **character.**
A NULL value will evaluate to NULL in any expression e.g. NULL multiplied by 10 is **NULL.**
NULL value can be inserted into columns of any data type.
If the column has a NULL value, Oracle ignores the UNIQUE, FOREIGN KEY, CHECK constraints that may be attached to the column.

NOT NULL constraint defined at the column level: When a column is defined as "**not null"**, then that column becomes a **mandatory** column. It implies that a value must be entered into the column if the record is to be accepted for storage in the table.

Syntax: columnname datatype (size) NOT NULL

Example: Create a table student with the following mandatory fields. Admnno, name, sscno, aadhar, group columns.

CREATE TABLE student admnno number(6) NOT NULL, name varchar2(15) NOT NULL, sscno number(10) NOT NULL, aadhar number(10) NOT NULL, group varchar2(11) NOT NULL, telno number(10));

The **NOT NULL** constraint can only be applied at column level. Although **NOT NULL** can be applied as **CHECK** constraint, however Oracle recommends that this be not done.

The UNIQUE constraint: The purpose of a unique key is to ensure that information in the column(s) is unique, means, a value entered in column(s) defined in the unique constraint must not be repeated across the column(s). A table may have many unique keys.

UNIQUE constraint defined at the COLUMN level:
Syntax: columnname datatype(size) UNIQUE

UNIQUE constraint defined at the TABLE level:

Syntax: UNIQUE (columnname, [, columnname, …..])

Example: Create a table student such that the unique constraint on the column admnno is desired as a table level constraint.

CREATE TABLE student(admnno number(6), name varchar2(20), group char(10), village varchar2(10), UNIQUE(admnno));

The PRIMARY KEY constraint: The primary key is one or more column(s) in a table used to uniquely identify each row in the table. A **primary key column** in a table has special attributes:

It defines the column as a mandatory column. It means, the column cannot be left blank. The NOT NULL attribute is active.
The data held across the column MUST be UNIQUE.

A single column primary key is called a **simple** key. A multicolumn primary key is called a **composite** primary key. The only function of a primary key in a table is to uniquely identify a row. Only when a record cannot be uniquely identified using the value in a single column, will a composite primary key be defined.

PRIMARY KEY constraint defined at the COLUMN level:
Syntax: columnname datatype(size) PRIMARY KEY

PRIMARY KEY constraint defined at the TABLE level:
Syntax: PRIMARY KEY (columnname [, columnname, …..])

Example: CREATE TABLE employee (empid number(10), empname char(20), department varchar2(10), PRIMARY KEY (empid));

The FOREIGN KEY constraint: Foreign keys represent relationships between tables. A foreign key is a column (or a group of columns) whose values are derived from the **primary key** or **unique key** of some other table.

The table in which the foreign key is defined is called a **foreign table** or **detail table.** The table that defines the **primary or unique** key and is referenced by the foreign key is called the **Primary Table or Master Table**.

The master table can be referenced in the foreign key definition by using the REFERENCES adverb. If the name of the column is not specified, by default, Oracle references the primary key in the master table.

Insert or Update operation in the foreign table:

The existence of a foreign key implies that the table with the foreign key is **related** to the master table from which the foreign key is derived. A foreign key must have corresponding primary key or unique key value in the master table.

Delete operation on the primary table:

Oracle displays an error message if the user tries to delete a record in the master table **WHEN** corresponding records exists in the detail table.

The default behavior of the foreign key can be changed by using the ON DELETE CASCADE option. When the ON DELETE CASCADE option is specified in the foreign key definition, if the user deletes a record in the master table, all corresponding records in the detail table along with the record in the master table will be deleted.

Rules for Foreign Key / References constraint:

Rejects an INSERT or UPDATE of a value, if a corresponding value does not currently exist in the master key table.
If the ON DELETE CASCADE option is set, a DELETE operation in the master table will trigger the DELETE operation for corresponding records in the detail table.
Rejects a DELETE for the master table if corresponding records in the DETAIL table exist.
Must reference a PRIMARY KEY or UNIQUE column(s) in primary table.
Will automatically reference the PRIMARY KEY of the MASTER table if no column or group of columns specified when creating the FOREIGN KEY.
Requires that the FOREIGN KEY column(s) and the CONSTRAINT column(s) have **matching** data types.
May reference the same table named in the CREATE TABLE statement.

FOREIGN KEY constraint defined at the column level:
Syntax: columnname datatype(size) REFERENCES tablename [(columnname)] [ON DELETE CASCADE]

FOREIGN KE constraint defined the table level:
Syntax: FOREIGN KEY (columnname [, columname]) REFERENCES tablename [(columnname [, columname])

Assigning User Defined Names to Constraints:

Oracle assigns a unique name to each constraint. The convention used by Oracle is

SYS_C**n** where **n** is a numeric value that makes the constraint name unique.

Constraints can be given a unique user – defined name along with the constraint definition. A constraint can be dropped by referring to the constraint by its name. Under these circumstances a user defined constraint name becomes very convenient.

If Oracle generated name are to be used, it becomes difficult to search for identifying the required constraint to be dropped. Hence, user named constraints simplifies the task of dropping constraints. A constraint can be given a user – defined name by preceding the constraint definition with the reserve word CONSTRAINT and a user – defined name.

Syntax: CONSTRAINT <constraintname> <constraint definition>

The CHECK constraint: CHECK constraint must be specified as a logical expression that evaluates either to TRUE or FALSE.

A CHECK constraint takes longer to execute as compared to NOT NULL, PRIMARY KEY, FOREIGN KEY or UNIQUE. Thus CHECK constraints must be avoided if the constraint can be defined using Not Null, Primary Key or Foreign Key constraint.

CHECK constraint defined at the column level:
Syntax: columnname datatype (size) CHECK (logical expression)

CHECK constraint defied at the table level:
Syntax: CHECK (logical expression)

When using CHECK constraints, consider the ANSI / ISO standard which states that a CHECK constraint is violated only if the condition evaluates to **False**. A check constraint is not violated if the condition evaluates to **True.**

If the expression in a check constraint does not return a true / false, the value is **Intermediate** or **Unknown. Unknown values do not violate a check constraint condition.** For example, consider the following CHECK constraint for costprice column in the product table.

CHECK (costprice>0)

At first sight, this rule may be interpreted as "do not allow a row in the **product** table unless the **costprice is greater than zero.**

However, note that if a row is inserted with a **null** costprice, the row does not violate the CHECK constraint because the entire check condition is evaluated as unknown.

Rules for CHECK constraints:

A check integrity constraint requires that a condition be **true** or **unknown** for the row to be processed. If an SQL statement causes the condition to evaluate to **false**, an appropriate error message is displayed and processing stops. A CHECK constraint has the following limitations.

The condition must be a Boolean expression that can be evaluated using the values in the row being inserted or updated.
The condition cannot contain subqueries or sequences.
The condition cannot include the SYSDATE, UID, USER or USERENV SQL functions.

DEFAULT value concepts:

At the time of table creation a **default value** can be assigned to a column. When the user is loading a **record** with values and leaves this column empty, the Oracle engine will automatically load this column with the default value specified. The data type of the default value should match the data type of the column. We can use the **DEFAULT** clause to specify a default value for a column.

Syntax: columnname datatype (size) DEFAULT (value);

The data type of the default value should match the data type of the column.
Character and date values will be specified in single quotes.
If a column level constraint is defined on the column with a default value, the **DEFAULT VALUE CLAUSE** must precede the constraint definition. Thus the syntax will be:

Columnname datatype (size) **DEFAULT** value constraint definition

Operators in SQL * Plus

The following are the operators supported by SQL * Plus

Arithmetic operators
Comparison operators
Logical operators

Arithmetic operators

To perform calculations based on number, values we include arithmetic expressions in SQL commands. An arithmetic expression consist of column names with number datatype and an arithmetic operator connecting them. The arithmetic operator are (addition) -) (subtraction) * (multiplication) and (division). Consider the following examples.

Example

SQL > select prodid, std price , mini price, std price + miniprice from price where enddate = '4 Jan 89';

We know the std price miniprice is not a column in the table price, yet SQL * plus would display it as a separate column. Arithmetic can also be performed in a where clause.

If there are several operator in an arithmetic expression then the precedence of each operator must be known and have equal higher precedence whereas + and - have equal lower precedence.

An example given below illustrates the precedence of operators.

Example

SQL > select ordid, itemid, 100 * (actual price + qty) from item where prodid = 111;

In the above example only after adding actual price with qty it is multiplied by 100. Parenthesis is omitted then multiplications will be performed first followed by addition. Thus we can control the order of evaluations by using parenthesis.

We now move on to study about comparison operators.

Comparison operators

Comparison operators are used in conditions to compare one expression with another. The comparison operators are=,I=,<,>, <=, >=, between (to check between any two values) in (to match with any of the values in the list) like (to match a character pattern and is null (to check whether it is null).

The last four operators mentioned above can also be used for checking the NOT conditions like NOT BETWEEN , NOT LIKE and so on. The following examples illustrates the comparison greater than.

Example

SQL > select * from order _info where total > 7000;

The above examples displays the rows for which the total is greater than 7000. An example discussed below is used to check for negations (i.e. check for NOT conditions).

Example

SQL > select * from order_info where not (custid = 2 or custid = 4);

The above command will display all columns where custid is neither 2 nor 4. The IN operator can be sued to select rows that match one of the values in a list.

Example
SQL > select * from order_info where shipdate in (‘12 January 88’, ‘30 -Jan 67’);

The above examples list all the columns in a table for which ship date lies in the given list. When we search for characters values using IN operator , the column name must exactly match with the values present in the list. In the case of LIKE operator which is used to search a character pattern, we need not know the exact character value. The LIKE operator recognizes special characters like % and .The former can match zero or more character while the later matches exactly one characters.

Example
SQL > select name, address, city from customer where name like ‘v%’;

The above command will display the columns name address, city and phone whose name begins with ‘v’.

Example
SQL > select * from customer where name like ‘V_P’;

The above command lists all the columns from customer table whose names are three letters, starting with ‘V’ ad ending with ‘P’.

Logical Operator

A logical operators is a used to combined the result of two condition as to produce a single result. The logical operators are AND, NOT and OR.

The following exemplified and AND operator which displays all the columns from order_info table provided both the conditions mentioned below are satisfied.

Example
SQL > select * from oder_info where shipdate > ‘4-Jan-11 AND total < 9000;

We have now gone through the various operators supported by SQL * Plus let us have a look at the order in which these operators are evaluated.

Operator Procedure
The precedence of the operators discussed above are given below

Arithmetic operators	-Highest precedence
Comparison operators	
NOT logic operator	
AND logic operator	
OR logic operator	-lowest precedence

Set operators and joins:

Set Operators

Set operators combine the results of two queries into a single one. The following set operators aid SQL in joining queries to retrieve rows. Now let us discuss them in detail.

Union
Union all
Intersect
Minus

The column in the select statements joined using the set operators should adhere strictly to the norms mentioned below.

The queries which are related by a set operator should have the same number of columns and the corresponding columns must be of the same datatype.
Such a query should not contain any column of type long.
The label under which the rows are displayed are those from the first select statement.

Union

The union operator returns all distinct rows select by either query. The following example combines the results of two queries with the union operator which eliminate duplicate rows.

Example: SQL > select custid from customer union select ordid from order_info;

Union all

The 'union all' operator returns all rows selected by either query including duplicates. The following example combines the result with the 'union all' operator, which does not eliminate duplicate rows.

Example
SQL > select custid from customer union all select ordid from order_info;

Intersect

Intersect operator returns only rows that are common to both the queries.

The following example is illustrative of the above statement.
Example: SQL > select ordid from order_info intersect select itemid from item;

Minus

Minus operator returns all distinct rows selected only by the first query and not by the second. The following example illustrates this.

Example: SQL > select ordid from item minus select ordid from order_info;

While using the 'order by' clause, it must follow the last select statement and we must order by integer and not by column name. Consider the following example.

Example

SQL > select ordid, custid from order_info union select itemid, prodid from item order by 2;

The above example displays distinct rows selected by either query, ordered by the second column stated in both the queries. Since the column names are different in the above queries, we use an integer in the 'order by' clause instead of a column name.

Relating data through join concept

The purpose of a join is to combine the data spread across tables. A join is actually performed by the 'where' clause which combines the specified rows of tables. The syntax for joining tables is follows.

Select columns from table1, table2, where logical expressions;

4.5.2 joins

The logical expression specifies how the table are joined. There are basically three different types of joins . They are

Simple join
Self join
Outer join

4.5.2.a Simple join

It is the most common type of join. It reviews rows from two tables having a common column and is further classified into equi-join and non-join. Let us discuss them in detail.

Equi-join

A join which is based on equalities is called an equi-join. Consider the following example.

Example

SQL > select * from item, order_info where item. ordid = order_info ordid;

If the above statement, item ordid=info . Ordid perform the joint operation. It retrieves rows from both the tables provided they both have the same ordid as specified the 'where' clause. Since the 'where' clause uses comparison operator equal to (=), to perform a joint, it is said to be an equijoin.

Non equi-join

A non-equi join specifies the relationship between columns belonging the different tables by making use of the relation operator (>,<,<=,>=,< >) other than =. The following example is illustrative of this.

Example

SQL > select * from customer, order_info where customer custid> order_info, custid and customer, repid = 10;

The following example join the rows of customer table to that of order_info table provided custid's belonging to customer table are greater than custid's belonging to order_info table. Yet it retrieves rows only when the second expression, i.e., repid = 10 is true.

Table aliases

To prevent ambiguity in a query we include table names in the select statements. Table aliases are used to make table queries shorter and more readable. As a result, we give in alias to the table in the 'from' clause and use it instead of the table name throughout the query. The following example is illustrative of the same.

Example

SQL > select c. *, o. * from customer c, order_info where c.custid> o.custid and repid = 10;

The above example is same as example 2 but uses tables aliases, where 'o' refers to order_info table and 'c' refers to customer table. Table aliases which are defined in the 'from' clause are separated from the table name by spaces.

4.5.2. b Self join

Joining of a table to itself is known as a self join, i.e., it joins one raw in a table to another. It can compare each row of the table to itself and also with other rows of the same table. The following example will illustrate this concept.

Example

SQL > select o. * from order _info o, order_info u where o.ship-date > = u.ord_date and o.custid = u.custid;

PAPER – IIRELATIONAL DATABASE MANAGEMENT SYSTEM

The above example will display only those rows from the order_info table whose ship-date is either greater than or equal to the order-date provided they both have the same custid.

4.5.2.c Outer Join

The outer join extends the result of a simple join. An outer join returns all the rows returned by simple join as well as those rows from one table that do not match any row from the other table. The symbol, (+) represent outer join. The following example helps us for better understanding.

Example :

SQL > select order_info. ordid, prodid, item. ordid from item, order_info where oder_info. ordid (+) = item. ordid;

The above example will also retrieve rows from item table which does not have any matching records in the order_info table because of the presence of an outer join (+).

Usage of Subqueries

Nesting of queries, one within the other, is termed as subquery. A statement containing subqeury is called a parent statement. Subqeuries are used to retrieve data from tables which depend on the values in the table itself.

The following example is illustrative this.

Example

SQL > select repid, address, name from customer where name = (select name from customer where repid = 10);

Sub queries and database objects.

Nesting of queries, one within another, is termed as subquery. The above example used a subquery which returned a single value. Subqeuries can also return more than one values. In such we should include operators like **any, all, in** or **not in,** between the comparison operator and the subquery.

The following example illustrate the usage of the **any** operator.

Example

SQL > select name, city, address from customer where repid < any (select repid from customer
where credit limit = 45345);

In the above example the subqeury will display that area which has max credit equal to 45345. The main query will display details about customer, if area is lesser than any of the values returned by the subquery.

Example

SQL > select name, city, address from customer where repid all (select repid from customer where credit limit = 45345);

In the above example the subquery will display area which has maxcredit equal to 45345. The main query will display details about customer only if area is lesser than all the values returned by the subquery.

Multiple Subqueries

Oracle places no limit on the number of queries included in a where clause. Consider the following example which illustrates multiple queries.

Example

SQL > select name, repid from customerwhere repid = (select repid from customer where custid = 3)or creditlimit> (select creditlimit form customer where credit = 2);

The above example will select the columns name, repid from customer provided one of the above subqueries is true.

A subquery itself can contain a subqeury. The following example is illustrative of this statement.

Example : SQL > select * from order-info where ordid = (select ordid from order_info where custid = (select custid from order_info where total = 7000));

A subquery can retrieve information from more than one table. Consider the following example.

Example :

SQL > select name, city, address from customer where custid in (select custid from order_info o, item i where o.ordid = i.ordid));

In the above example the subquery refers to two table, order_info and item. It will display rows from customer table for custid returned byte subquery.

Correlated subquery

A subquery is evaluated once for the entire parent statement whereas correlated subqueries is evaluated once per row processed by the parent statement. Consider the for following example which returns rows total which are greater than avg (total) which reference to ordid.

Example :

SQL > select ordid, orderdate, shipdate, from order_info o where shipdate <any (select orderdate from order_info where o.ordid =ordid);

The behavior of a correlated subquery is evaluated as follows

Ordid of the row is first determined.
Ordid is then used to evaluate parent query.
If that row's shipdate is greater than any of the order dates, the row is returned.
The supplementary is evalauted once for each row of the order_info table until all the rows of the table have been tested.

More Data Objects

Once of the database objects namely, table was the topic of discussion in previous sessions. In the session we will be discussing the database objects mentioned below

View
Synonym
Sequence
Index

View

A view is an imaginary table and it contains no data and the tables upon which a view is based are called base tables. The advantages of view are as follows.

They provide table security by restricting access to predetermined set of rows or columns of a table.

They simplify commands for the use because they allow them to select information from multiple tables.

They provides data in a different perspective than that of base table by remaining columns without affecting the base table.

The syntax for creating a view is given below

Create view <view name> (column alias name...) as query (with check option constraint);

Example :

SQL > create view custview (customer, city, representative) as select custid, city, repid from customer where repid = 20 with check option;

In the above example the view only about representative whose repid is 20. Because of the check option we should insert only those customer whose repid is 20.

We all know how to perform manipulations on table data. The same is applicable to views also. Let us discuss them in details.

To display the contents of the view , select statement is used.

Example :

SQL > select * from cust view ;

Similarly we can perform updations, deletions and insertions in a view with certain restrictions. To illustrates this, consider the following views which are derived from the same table item.

Example :

SQL > create view ve1 (item, product) as select itemid, prodid from item;

Example :

SQL > create view ve2 (quantity price) as select qty, actual_price from item.

We can insert rows into both the views, provided other column in the base table accept null values. If those column were assigned as not null, then we cannot insert rows. We can perform updations or deletions on ve1 because itemid is declared as primary key which could be used to identify a row. Only after identifications, the corresponding can be deleted or updated. In the case of ve2 we can perform updation or deletion with some restrictions. We need to specify which row we actually mean by using the where clause. Joining of table i also possible in a view. But we cannot update, delete or insert through a view if it selects columns from more than one table. The following example creates a view from two tables.

Example :

SQL > create view joinview (customername, maximumcredit, total) as select name, creditlimit , total from customer, order_info where customer. custid = order_info.custid;

In the above example the view join view will contain information about customer and also details about maximum credit than he can avail. Since this information is spread across two table, we use the concept of join to retrieve them.

Function in view

Single row functions comprising number, character, date, group function and expression can also be used in view. Consider the following example which illustrates the usage of expression n view.

Example : SQL > create view discount (quantity, ordid, price) as select qty, ordid, actualprice/50 from item ;

The above example finds the discount offered for a product.

Synonym

A synonym is a database object which is used as an alias (alternative name) for a table view or sequence. They are used to

Simplify SQL statement.
Mask the name and owner of an object
Provide public access to an object.

Synonym can either private or public,. The former is created b the use, which is available only to that person whereas the latter is created by the DBA, which can be availed by any database user, The syntax for creating a synonym is a given below.

Create (public) synonym <synoynm_name for <table _name> (.@database link)

Consider the following example which represents in the table customer with a different name say, cust.

Example : SQL > create synonym cust for customer;

Sequences

A sequence is database object which can generate unique, sequential integer values. It can be used to automatically generate primary key or unique key value. A sequence can be either an ascending or a descending sequence. The syntax for creating a sequence is as follows.

Create sequence <seq_name> increment by n start with n (max value n) (min value n) (cycle/ noncycle) (cache/ noncache);

For creating sequence we have to define the following terms.

Increments by n : 'n' is an larger which specifies the interval between sequence number. The default is 1 if n is positive then the sequence ascends and if it is negative the sequence descends.

Start with n : Specifies the first sequence number to be generated.

Minvalue n : Specifies the minimum value of the sequence. By default, it is 1 for an ascending sequence and 10e26-1 for a descending sequence.

Max value : It specifies the maximum value that the sequence can generate. By default it is - 1 and 10e27-1 for descending and ascending sequence respectively.

Cycle : Specifies that the sequence continues to generate values after reaching either max or min value.
No cycle: Specifies that the sequence cannot generate more values after reaching either its max or min value . The default value is 'no' cycle.

Cache: Allows faster generations of sequence number. Oracle preallocates sequence number of keep than in memory for faster access. The cache values must be less than max value minus min value and by default it is 20.

No cache : The default value 'noncache' does not preallocate sequence number for faster access.

Example :

```
SQL > create sequence custeq
      increment by 1 start with 1
      max value 10
      min values 1
      cycle cache 4;
```

After creating a sequence we can access it values with the help of pseudo columns like currval and nextval. A pseudo column behaves like a table. We can select values from pseudo columns but cannot perform manipulations on their values.

Nextval : Nextval will return initial value of the sequence when referred to , for the first time. Later reference to nextval will increment the sequence value by INCREMENT BY clause and return the new value.

Currval : Currval returns the current value of the sequence which is the value returned by the last reference to nextval. The following example uses custeq to insert values in the id column belonging to the temp table created for this example.
Example :

```
SQL > Create table temp (id number (3), name varchar (15))
; SQL > insert into temp values (custeq.nextval, 'vijay') ;
```

To find out the current sequence number, we issue the following SQL statement .

Example :
SQL > select custeq .currval from temp;

Alter sequence
We can alter the sequence when we want to do the following
Set or eliminate min or max value.
Change the increment value.
Change the number of cached sequence number.

The following example will change the max value of the sequence from 10 to 15.

Example :
SQL > alter sequence custeq max values 15;

Index

An index is a database object. Its purpose is to increase the performance of data retrieval and thereby provide faster access path to the table data. Indexes can be created on more than one column of a table. The syntax is given below.

Create (unique) index <index_name> on <table_name> (column, column..);

Where the keyword 'unique' ensure that the tables does not allow two rows with identical values in all indexed columns. Square brackets indicate that 'unique' is optional.

Consider the following example which will create an index to the prodid column of the price table, to give direct access to the rows in customer table.

Example :
SQL > Create index ind_pri on price (prodid);

Drop databse object

We can remove a database object from the databse by issuing the following

syntax. drop database object <object_name>;

The following example a view named discount from the databse.

Example :
SQL > drop view discount ;

Quick Reference

Data types in SQL are CHAR , VARCHAR,VARCHAR 2, NUMBER, DATE, LONG, RAW / LONG RAW.

The DDL commands are CREATE, ALTER, DROP.

The DML commands are INSERT, DELETE, UPDATE, SELECT.

The DCL commands are GRANT, REVOKE, COMMIT, ROLLBACK, SAVE POINT.

Nesting of queries, one within another, is termed as subquery.

Short Answer Type Questions

What are different data types in SQL.

Write are different DDL commands.

Write are different DML commands.

Write are different DCL commands.

What is sub query ?

Long Answer Type Questions

Explain in detail three DDL commands with examples.

Explain any four DML commands with examples.

Explain any four DCL commands with examples.

Write in detail about different SET operators and Joins.

UNIT 5

Structure

Software Development and Life cycle

Definition of system, analysis and design

Study of software life cycle

Requirement analysis, design, development, testing, implementation and maintenance

Learning Objectives

After studying this unit, the student will be able to

Know about what is a system, design.
Know about the software development life cycle.
Know about the software testing.
Know about the system implementation.

Introduction:

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product.**

Software Engineering is the application of a systematic, disciplined, Qualifiable approach to the development, operation and maintenance of software that is the application of Engineering to software.

Definition of system, analysis and design:

Generally, A system is a collection of elements or components that are organized for a common purpose.

A **system** is a set of interacting or interdependent components forming an integrated whole or a set of elements (often called *'components'*) and relationships which are different from relationships of the set or its elements to other elements or sets.

The word “System” is derived from the greak word “Systema” which means an organized relationship.
Ex: Financial accounting system, net Banking system etc.,

A softwaresystem is a system on intercommunicating components based on software forming part of a computer system (a combination of hardware and software). It "consists of a number of separate programs, configuration files, which are used to set up these programs, system documentation, which describes the structure of the system, and user documentation, which explains how to use the system".

A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective. The study of system concepts has three basic implications:

A system must be designed to achieve a predetermined objective.
Interrelationships and interdependence must exist among the components.
The objectives of the organization as a whole have a higher priority than the objectives of its subsystems.

Sub – System: One of the number of component parts of a system. All the subsystems must function together in an integrated manner for the system to operate as designed.

Characteristics of a system:

Organization: It implies structure and order. It is the arrangement of components that helps to achieve objectives.

Interaction
Interdependence
Integration
Central Objective

Interaction: It refers to the manner in which each component functions with other components of the system.

Interdependence: It means that parts of the organization or computer system depend on one another. They are coordinated and linked together according to a plan. One subsystem depends on the output of another subsystem for proper functioning.

Integration: It refers to the holism of systems. It is concerned with how a system is tied together.

Central Objective: A system should have a central objective. Objectives may be real or stated. Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operate to achieve another. The important point is that users must know the central objective of a computer application early in the analysis for a successful design and conversion.

Some systems share common characteristics, including

A system has structure, it contains parts (or components) that are directly or indirectly related to each other.
A system has behavior, it contains processes that transform inputs into outputs (material, energy or data).
A system has interconnectivity: the parts and processes are connected by structural and/or behavioral relationships.

A system's structure and behavior may be decomposed via subsystems and sub-processes to elementary parts and process steps.

Elements of a System: The basic elements of software system are:

Outputs and inputs
Processors
Control
Feedback
Environment
Boundaries and Interfaces

Outputs and inputs: A major objective of a system is to produce an output that has value to its user. In order to get a good output, inputs to system must be appropriate. It is important to point out here that determining the output is a first step in specifying the nature, amount and regularity of the input needed to operate a system.

Processors: It is the element of a system that involves the actual transformation of input into output. It is the operational component of a system. Processors may modify the input totally or partially, depending on the specifications of the output. In some cases, input is also modified to enable the processor to handle the transformation.

Control: The control elements guide the system. It is the decision-making subsystem that controls the pattern of activities governing input, processing, and output.

Feedback: Feedback measures output against a standard in some form of cybernetic procedure that includes communication and control. Feedback may be positive or negative, routine or informational. Positive feedback reinforces the performance of the system. It is routine in nature. Negative feedback generally provides the controller with information for action.

Environment: The environment is the "supra-system" within which an organization operates. It is the source of external elements that impinge on the system. In fact, it often determines how a system must function.

Boundaries and Interfaces: A system should be defined by its boundaries- the limits that identify its components, processes, and interrelationships when it interfaces with another system

Types of System

Physical or Abstract Systems
Open or Closed Systems
Man-made Information Systems

Physical or Abstract Systems: Physical systems are tangible entities that may be static or dynamic in operation. Abstract systems are conceptual or nonphysical entities. They may be formulas of relationships among sets of variables or models – the abstract conceptualization of physical situations.

Open or Closed Systems: An open system has many interfaces with its environment. It permits interaction across its boundaries; it receives inputs from and delivers outputs to the outside. A closed system is isolated from environment influences.

Man-made Information Systems: An information system is the basis for interaction between the user and the analyst. It provides instructions, commands, and feedback. It determines the nature of relationships among decision makers. From this basis, an information system may be defined as a set of devices, procedures, and operating systems designed around user-based criteria to produce information and communicate it to the user for planning, control and performance.

System Analysis: According to "Merriam - Webster" dictionary system analysis as "the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way". The terms analysis and synthesis stem from Greek, meaning "to take apart" and "to put together," respectively. System analysis is used in every field where something is developed. Analysis can also be a series of components that perform organic functions together, such as system engineering. System engineering is an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed.

System analysis is, process of gathering and interpreting facts, diagnosing problems and using the information to recommend in the improvement of the system.

System Analysis is conducted with the following objectives:

Identify the customer's need.
Evaluate the system concept for feasibility.
Perform economic and technical analysis.
Allocate functions to hardware, software, people, database and other system elements
Establish cost and schedule constraints
Create a system definition that forms the foundation for all subsequent engineering work.

System Analyst: System Analysis is done by System Analyst.

An individual who analyses a system by the use of scientific techniques in order to determine where and how the improvements can be made with a view to meet objectives n a more efficient an economical manner.

An analyst uses analysis and design techniques to solve business problems using information technology. Systems analysts may serve as change agents who identify the organizational improvements needed, design systems to implement those changes, and train and motivate others to use the systems. Although they may be familiar with a variety of programming languages, operating systems, and computer hardware platforms, they do not normally involve themselves in the actual hardware or software development. They may be responsible for developing cost analysis, design considerations, staff impact amelioration, and implementation timelines.

Role of System Analyst:

Identify, understand and plan for organizational and human impacts of planned systems.
Ensure that new technical requirements are properly integrated with existing processes and skill sets.

Plan a system flow from the ground up.

Interact with internal users and customers to learn and document requirements that are then used to produce business required documents.

Write technical requirements from a critical phase.

Interact with software architect to understand software limitations.

Whenever a development process is conducted, the system analyst is responsible for designing components and providing that information to the developer.

Help programmers during system development, e.g. provide use cases, flowcharts, UML and BPMN diagrams.

Contributes in the preparation of user manuals and document requirements.

Design: The good of system design is like planning a new business or replace or improve the performance of existing system. It like a blue print for a building.

Software design is an iterative process through which requirements are translated into a "**BLUEPRINT"** for constructing the software. The design is represented at a high level of abstraction i.e., a level that can be directly traced to specific data, functional, and behavioral requirements.

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development.

Basically the system design is of three types

Architectural design
Logical design
Physical design

Architectural design: The architectural design of a system emphasizes the design of the system architecture that describes the structure, behavior and more views of that system and analysis.

Logical design: The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the environment of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

Physical design: The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

Input requirement,
Output requirements,
Storage requirements,
Processing requirements,
System control and backup or recovery.

Other design Methodologies:

Rapid application development (RAD): Rapid application development (RAD) is a methodology in which a system designer produces prototypes for an end-user. The end-user reviews the prototype, and offers feedback on its suitability. This process is repeated until the end-user is satisfied with the final system.

Joint application design (JAD): Joint application design (JAD) is a methodology which evolved from RAD, in which a system designer consults with a group consisting of the following dignities:

Executive sponsor
System Designer
Managers of the system

JAD involves a number of stages, in which the group collectively develops an agreed pattern for the design and implementation of the system.

Study of Software life cycle or System Development Life Cycle

Software Development Life Cycle (**SDLC**) is a process used by the **software** industry to design, develop and test high quality software. **SDLC** is a framework defining tasks performed at each step in the **software development** process. ISO/IEC 12207 is an international standard for **software life-cycle** processes.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

Purpose of SDLC:

The primary reasons to say SDLC is important for developing a software system.

It offers a basis for project planning, scheduling, and estimating
Provides a framework for a standard set of activities and deliverables
It is a mechanism for project tracking and control
Increases visibility of project planning to all involved stakeholders of the development process
Increased and enhance development speed
Improved client relations
Helps you to decrease project risk and project management plan overhead

For the development of a new software system or to modify an existing system the following phases should be considered.

Requirement Analysis
Feasibility Study.
Design
Development of Software
Testing
Implementation and maintenance.

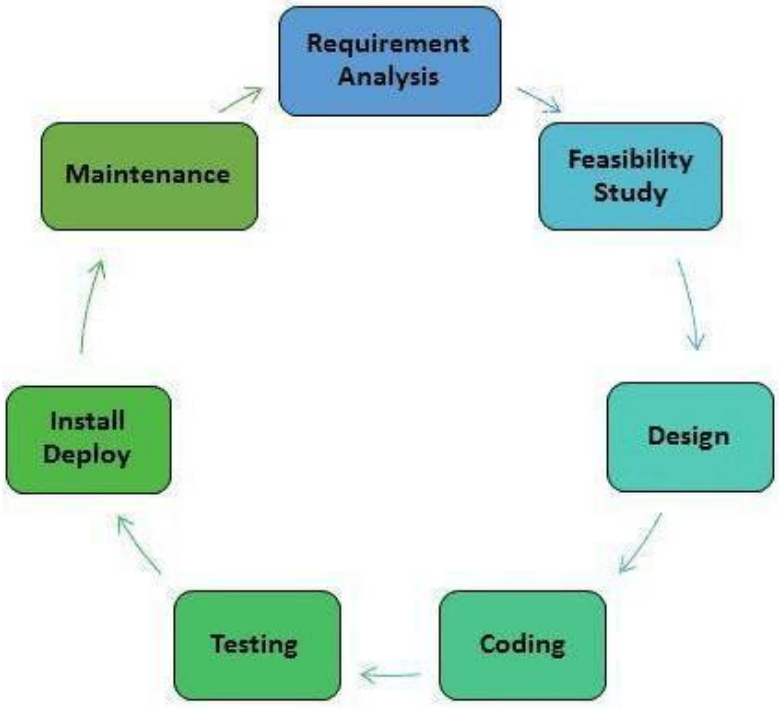


Fig. 5.2.1

Stage 1: Requirement Analysis: The requirement analysis is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and identification of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project. Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

Before developing any system, investigation can be considered. The project request must be examined to know what the client needs. The important aspect of the preliminary investigation is the request for information system feasibility.

Stage 2 Feasibility study: Once the requirement analysis phase is completed the next step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

Economic: Can we complete the project within the budget or not?
Legal: Can we handle this project as cyber law and other regulatory framework / compliances.
Operation feasibility: Can we create operations which is expected by the client?
Technical: Need to check whether the current computer system can support the software
Schedule: Decide that the project can be completed within the given schedule or not.

Defining Requirements: Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Developing information system is a detailed understanding of all important facts of the business needs.

Fact finding techniques are

Review of written Documents
Onsite observation
Interviews
Questionnaires

Stage 3: Designing: The most challenging phase of the SDLC is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications often called logical design and physical design. In design phase various types of Actors will be identified**.**

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Following are the most important and popular SDLC models followed in the industry

Waterfall Model
Iterative Model
Spiral Model
V-Model
Fish Model
Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

Stage 4: Developing the software: The system development is based on design specification. The software developers may install software (or) they write new custom design, programs known as physical design. In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product: This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment (Implementation) and Maintenance: Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

The following concepts also took place in this phase

Training and Transition
Operations and maintenance
Evaluation

Requirement analysis, design, development, testing, implementation and maintenance

Requirement Analysis: Before developing any system, investigation can be considered. The project request must be examined to know what the client needs. The important aspect of the preliminary investigation is the request for information system feasibility.

Developing information system is a detailed understanding of all important facts of the business needs. Analyst working with employees a managers must study the business process to answer the following questions.

What is being done.
How is it being done.
What type of transactions (or) decisions take place.
Does problem exist.
If a problem exist, what is the effect
If a problem exist what is the reason.

Design: The most challenging phase of the SDLC is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications often called logical design and physical design. In design phase various types of Actors will be identified . Different types of diagrams used in designing a software are as follows.

Data Flow Diagram
Object Diagram
Sequence Diagram
Collaboration Diagram
State Chart Diagram
Deployment Diagram

The following are the few Software Designing Models

Water Fall Model
Iterative Mofdel
V – Model
Fish Model
Spiral Model.
Big Bang Model

Development of Software: The system development is based on design specification. The software developers may install software (or) they write new custom design, programs known as physical design.

Testing: After completion of development the system is used experimentally to ensure that the software does not fail i.e., that software will run according to the user expectations for testing raw data for processing and examine the result. The system testing checks the readiness, and accuracy of the system to access, update and retrieve data from new files.

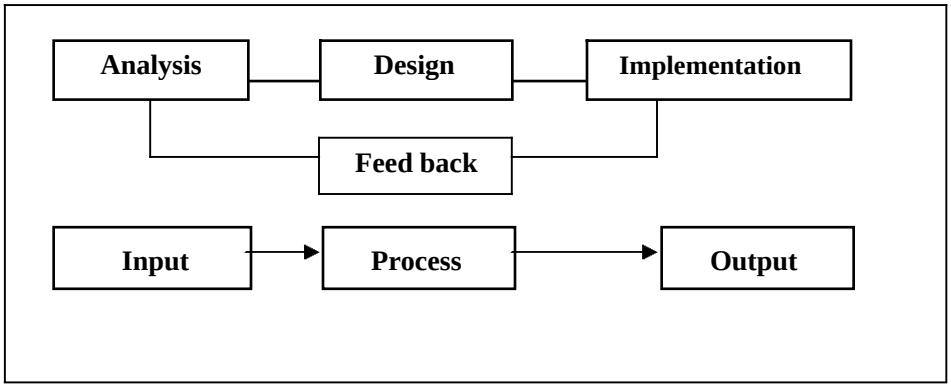


Fig 5.3.1

System testing does not test the software but rather the integration of each module in the system. It also tests to fine dependencies between the system and its original objectives current specifications and system documentation.

Testing of each component of a computer system by using actual post data (or) by pothetical data and analyzing the result in order to ensure that the system trouble free.

Types of Software Testing: Basically there are 3 different types of testing is involved in testing of a software system. Those are

Unit Testing
White Box Testing
Black Box Testing

Unit Testing: Unit Testing mainly focus on the verification of the smallest unit of software design or the module of a system. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of a module. The unit test is normally white – box oriented, and the step can be conducted in parallel for multiple modules.

White – Box Testing: White box testing, sometimes called **GLASS BOX TESTING,** is a test case design method that uses the control structure of the procedural design to derive test cases.

Black – Box Testing: Black – Box testing deals with the functional requirements of the software. i.e. Black – Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

PAPER – IIRELATIONAL DATABASE MANAGEMENT SYSTEM

Other Types of Testing:

Peak Load testing
Storage testing
Performance time testing
Recovery testing
Procedure testing
Human factor testing.

Peak Load testing: These are critical types in many systems particularly on line systems.
Ex: Banking systems this is real time.

Storage Testing: Analyst specifies a capacity for the system when it is designed and constructed storage testing. Often require entering data until capacity is reached.

Performance Time Testing: This test will be conducted prior to implementation to determine how long it takes to receive a response to inquiry, make a backup of a file or send a transmission and receive a response.

Recovery Testing: Analyst must always assume that the system will fail and data will be damaged or lost. By creating a failure or data loss event where the users forced to reload and recover from a backup copy, Analyst can readily determine whether recovery procedures and adequate, the best designed plans usually or adjusted.

Procedure Testing: Analyst should concentrate on removing diskettes before powering down, instructions about when to depress the enter key etc., and explain to user or the should give proper messages to user on screen with regarding procedures.

Human Factor Testing: This testing includes finding answers to questions about how people will react to the system in ways not anticipated.

Implementation and Maintenance:

The implementation phase is less creative than system design. It is preliminary concerned with user training, site preparations and file conversion. This method lastly but provides assurance against errors in the candidate system.

Training and Transition: Once a system has been stabilized through adequate testing, the SDLC ensures that proper training on the system is performed or documented before transitioning the system to its support staff and end users. Training usually covers operational training for those people who will be responsible for supporting the system as well as training for those end users who will be using the system after its delivery to a production operating environment.

After training has been successfully completed, systems engineers and developers transition the system to its final production environment, where it is intended to be used by its end users and supported by its support and operations staff.

Operations and maintenance: The **deployment** of the system includes changes and enhancements before the decommissioning or sunset of the system. **Maintaining** the system is an important aspect of SDLC. As key personnel change positions in the organization, new changes will be implemented. There are two approaches to system development: the traditional approach (structured) and **object oriented.** Information engineering includes the traditional system approach, which is also called the structured analysis and design technique. The object oriented approach views information system as a collection of objects that are integrated with each other to make a full and complete information system.

Evaluation: The final phase of the SDLC is to measure the effectiveness of the system and evaluate potential enhancements.

Data Dictionary: Data Dictionary is a repository that contains descriptions of all data objects consumed or produced by the software.

Quick Reference

System is an orderly grouping of interdependent components linked together to approach a Specific object or goal. Ex: Railway reservation system , Net banking system.

Sub System is One of the number of component parts of a system. All the subsystems must function together in an integrated manner for the system to operate as designed.

Analysis is detailed study of a Software system.

A person who analyses a system by the use of scientific techniques in order to determine where and how the improvements can be made with a view to meet objectives in a more efficient and economical manner is called System Analyst.

Fact finding techniques are

Review of written Documents
Onsite observation
Interviews
Questionnaires

The basic elements of software system are:

Outputs and inputs
Processors
Control
Feedback
Environment
Boundaries and Interfaces

Some of the Software Designing Models

Water Fall Model
Iterative Model
V – Model
Fish Model
Spiral Model.
Big Bang Model

Methods of system implementation

Direct Conversion
Parallel conversion
Phase conversion.

Different types of diagrams used in designing a software are as follows.

Data Flow Diagram
Object Diagram
Sequence Diagram
Collaboration Diagram
State Chart Diagram
Deployment Diagram

Data Dictionary is a repository that contains descriptions of all data objects consumed or produced by the software.

Short Answer Questions

What is System?
What is Sub – System?
What is System Analysis?
Who is System Analyst?
What are the fact finding techniques?
Write the elements of a software system.
What are different software design models can be considered while designing a software?
What are the methods of system implementation?
What are the different diagrams used while designing a software system?
What is Data Dictionary?

Essay Questions

Explain different stages of Software Development Life Cycle in detail?
What is Testing? Write different types of testing can be considered while testing a software product?

COMPUTER SCIENCE & ENGINEERING

Paper – III

Data Communications And Computer Networks

INDEX

UNIT 1

Data Communication

Objective:

After studying this unit, the student will be able to
Know about data communication.
Know about modes & types of data transmission.
Know about different types of communication channels.
Know about parallel and serial interfaces
Understand about different transmission modes.
Uses of Multiplexing.

Definition of Data Communications

Data communications is nothing but the exchange of information or data between two devices via some transmission medium. This transmission medium can be either wired or wireless. In data communication, the devices which are communicating must be the part of a communication system which is made up of with the combination of hardware (physical equipment) and software (programs). The effectiveness of a data communication mostly relies on four fundamental characteristics. They are

Delivery

Accuracy

Timeliness and

Jitter.

Explanation:

Delivery: The system should deliver the data to the correct destination i.e., from where the system had received the request to send the data. At the receiver side the data must be received by the requested device or user only.

Accuracy: The data which is sending from the system should beaccurate. In some cases the data may be alteredin the process of transmission and left uncorrected which is useless.

Timeliness: The system should deliver the data in a timely mode. If the data is delivered late then it is useless. In some cases like, audioand video, timely delivery means sending the data in the same order as they are produced, without any major delay. This type of delivery is known as real-timetransmission.

Jitter. Jitter means the variation in the arrival time of packets (data is send or received in the form of packets). Simply it is an irregular delay in thedelivery of videooraudio packets.

Frequently, there will be some cases where the data must be sent to the local circuitry which consist a computer. In such cases, the distances may be huge. Unfortunately, as the distance between the communication devices increase the strength of the signal will decrease, which results in the loss of some data and maybe some noise can also include in it, because of that, accurate transmission of data becomes very difficult. Although we have some precautions for data exchange within the computer but, the main problem occurs when we have to transfer the data to devicewhich is outside the computer network. In such case, distortion and noise will be so severe that information will be lost.

Components:

Mainly data communications systems consist of five components.

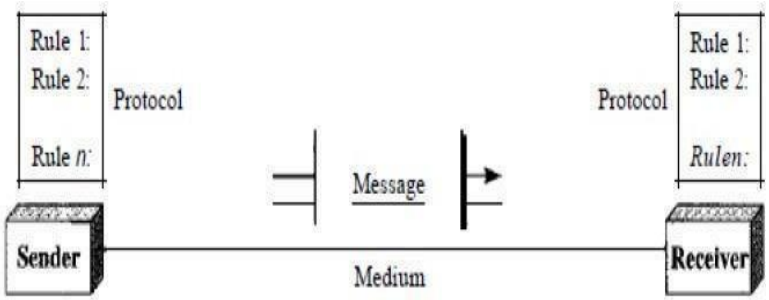


Fig 1.1: Architecture of the Data Communication System

Message: The message is nothing but some data which is to be communicated or shared with others. Data may be present in the form of text, pictures,numbers, video and audio.

Sender. The sender is the one who is responsible for sending the data message. Sender may be acomputer, telephone handset,workstation etc.

Receiver. The receiver is the one who receives the response (data) from the sender. Receiver may be acomputer, telephone handset,workstation etc.

Transmission medium. The transmission medium is nothing but, it isthe path by which the data travels from sender to receiver. This transmission medium maybe guided (wired) or unguided (wireless). Some of the examples of transmission media are coaxial cable,twisted-pair wire, fiber-optic cable, and radiowaves

Protocol. Simply a protocol is the set of rules which controls the data communications. It is an agreement between the devices which are communicating.

Modes of Data Transmission

Transmission of data can be occurred from one device to another device by using some means of communication media. Data transfer occurs from one device to another device through some electromagnetic or light waves in an encoded form known as signals. Data transmissions can occur across the network in two types of forms, namely

Analog signal.
Digital signal

Analog Data or Analog Signal: The term **analog data** refers about thedata which is continuous. For **example,** consider an analog clock which consists of an hour, minute, and second's hands which gives information in continuous form. In that analog clock themovement of the hands are continuous. Analog data, it maybe the sounds made by a human voice, will take continuous values. If someone speaks then, an analog wave will be created in the air, which can be captured by a microphone and can be converted to an analog signal or sampled. It can also be converted to a digitalsignal.

Digital Data or Digital Signal: The term D**igital data** refers to information which has distinct states.

For **example**, consider a digital clock that reports the values of hours and minutes when they change suddenly from 9:10 to 9:11. Digital data will take distinct values only. For example, Consider the data storage in our computer, data willbe stored in the form of Os (zeros) and 1s(ones). If we want to transmit that data across the medium first we have to convert that digital data to a digital signal or it can be modulated into an analog signal.

Analog and Digital Signals:

Signals can either be analog or digital. An analog signal will have infinitely numerous levels of intensity over a period of time. As the analog wave transfers from value A to value B, it has to pass through and will include an infinite number of values in its path. Unlike analog signal,Digital signal will have only a limited number of defined values. Even though each value is any number, it is as simple as 1 and O (digital data represented in 0 or 1). One of the simplest ways to represent this digital signal is by plotting them on a graph paper consisting of a pair of perpendicular axes. The vertical axis in the graph represents the values of signal or the strength of a signal,whereas the horizontal axis in the graph represents time.

The figure given below illustrates both analog signal and a digital signal. The curve in the analog signal has to pass through an endless number of points, whereas the vertical lines of the digital signal, illustrate the sudden jump which the signal will makes from value to value.

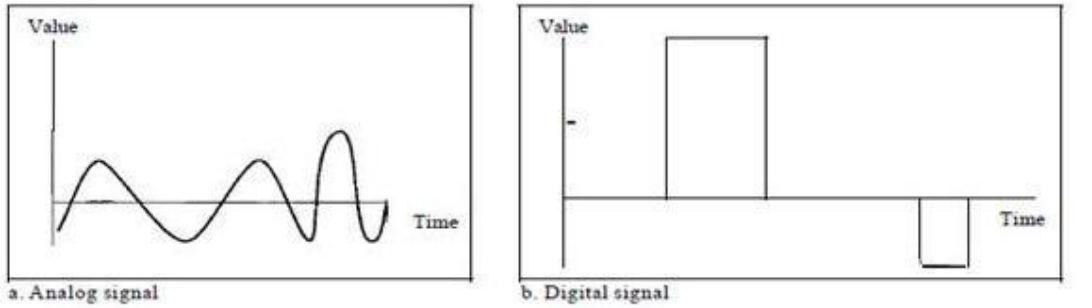


Fig 1.2 Representation of Digital and Analog Signal

Periodic and Nonperiodic Signals:

A periodic signal will complete a pattern within a calculable time frame, known as a period, and it will repeat that pattern over a successive identical periods. The completion of a full pattern is known as a cycle. A nonperiodic signal will change without displaying a pattern or cycle that repeats overatime.

Periodic Analog Signals:

Periodic analog signals can be classified into two types. They are simple and composite.
A simple periodic analog signal is a sine wave which cannot be decomposed into any other simpler signals.
A composite periodic analog signal is the composition of multiple sinewaves.

Sine Wave

One of the most fundamental form of periodic analog signal is the sine wave. When we visualize it,it will be a simple oscillating curve. This sine wave change over the course of a cycle will be smooth and consistent which will be continuous and rolling flow. Figure below illustrates a sine wave.

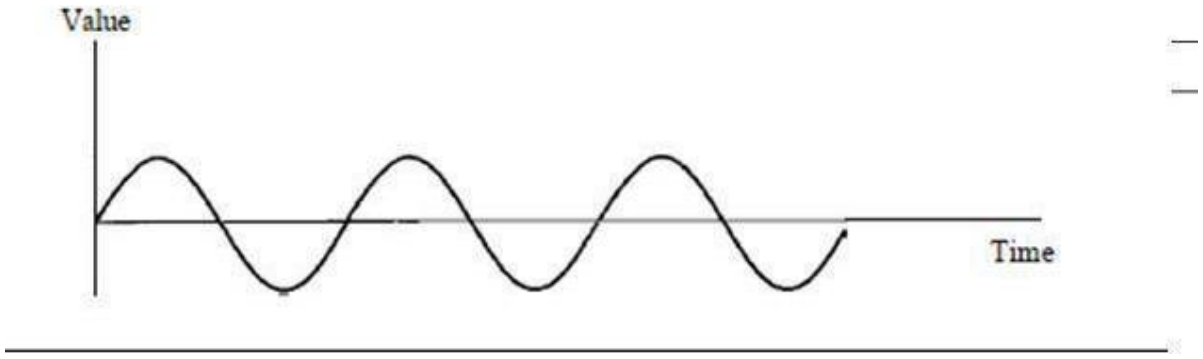


Fig 1.3 Representation of Sine Wave

Characteristics of Signals:

PeakAmplitude

The absolute value of the highest intensity of the signal is known as the peak amplitude of that signal. It is proportional to the energy that it carries. For an electric signal, peak amplitude will be measured in volts. Figure below illustrates the peak amplitude of the two signals.

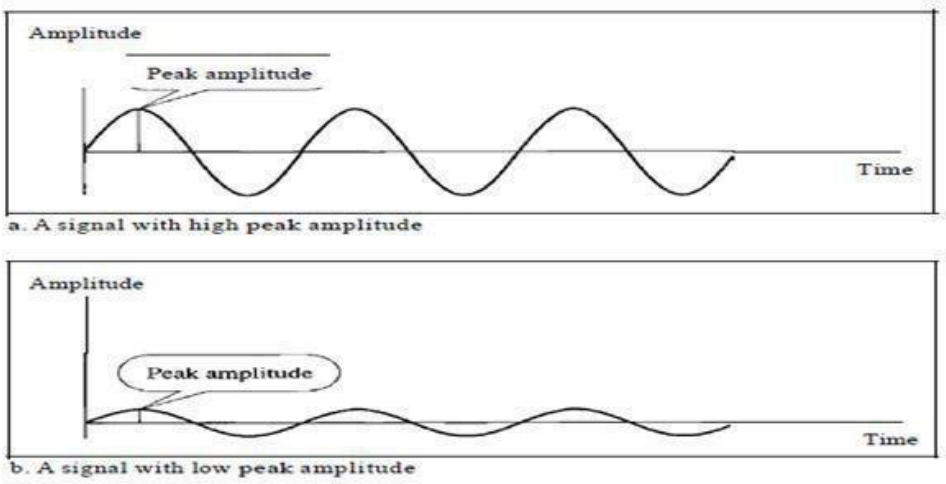


Fig 1.4 Two Signals with same phase and frequency with different amplitudes

Period andFrequency

Simply, Period refers to the amount of timein seconds and a signal needs to complete 1 cycle. Frequency is nothing but the number of periods completed in 1 s. Understand that frequency and period aresimply one characteristic defined in two different ways.

Period is the inverse of frequency, and frequency is the inverse of period. Consider the formula given below

$$f=1/T \text{ and } T=1/f$$

Where,

$f \rightarrow$ frequency and

$T \rightarrow$ Time period

Period is measured or expressed in seconds.

Frequency is measured or expressed in Hertz (Hz), which is cycle per second.

Phase

Simply, phase will describe the position of the waveform with respect to the time O. If we want to shift the wave either backward or forward along the time axis in the graph then, the phase will

describe how much amount it should be shifted. It will indicate the status of the first cycle. Phase is measured either in degrees or in radians [360° is 2n rad; 1° is 2n/360 rad, and 1 rad is 360/(2n)]. Consider an example shift,A phase shift of 360° will correspondto a shift of a complete period. Consider another phase shift of 180° which will be correspond to a shift of one-half of a complete period.

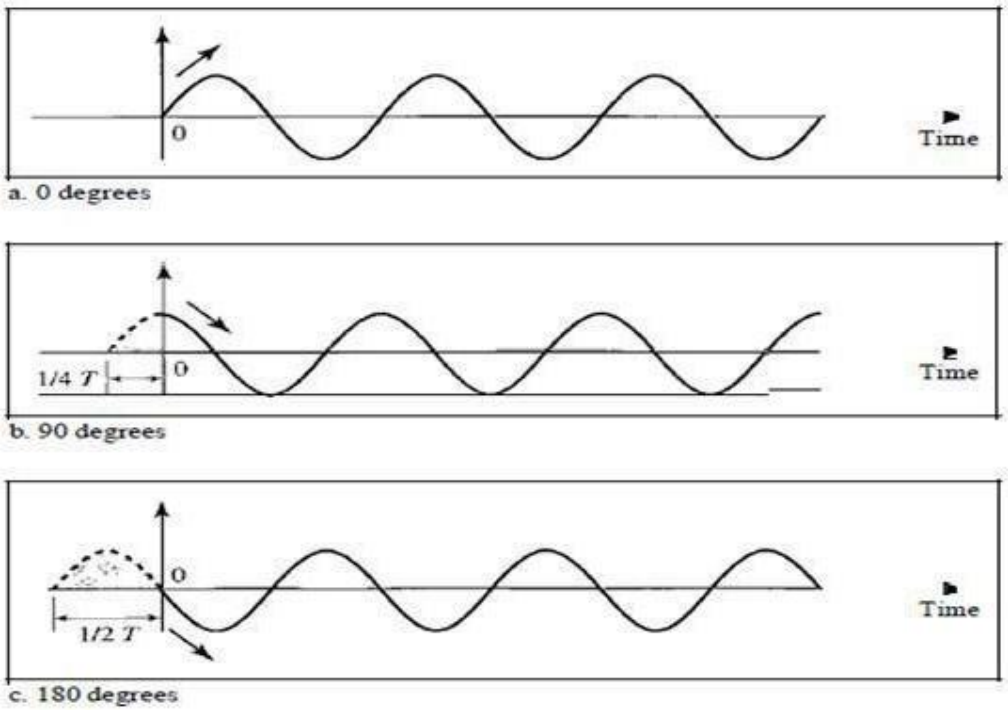


Fig 1.5 Three Signals with same phase and frequency with different amplitudes

In first sine wave, sine wave with a phase of 0° started at the time 0 with a zero amplitude. The amplitude will be either increasingor decreasing with respect to time.

II. In figure b, A sine wave with a phase of 90° started at time 0 with a peak (high) amplitude. The amplitude will be decreasing or increasing with respect to time.

III. In figure c, A sine wave with a phase of 180° started at time 0 with a zero amplitude. The amplitude is decreasing and after that it increases or decreases with respect to time.

4. **Wavelength**

Wavelength is also one of the characteristics of a signal which travels through a transmission medium. Itwill bind the frequency or period of a simple sine wave to the propagation speed of the medium. Even though the frequency of a signal is independent of the transmission medium, the wavelength will depends on both the frequency and the medium. Wavelength is one of the common properties of any

type of signal. In data communications, frequently we will use wavelength to describe the transmission speed of light in an optical fiber. Simply the wavelength is the distance which a simple signal can travel in one period of time.

We can calculate wavelength if the propagation speed (the speed of light) and the period of the signal are given. However, since period and frequency are related to each other (period is inversely proportional to the frequency),

if we denote wavelength by λ,

propagation speed by c (speed of light)

and frequency by f then,

Wavelength=Propagation speed * Period = propagation speed/frequency

λ=c/f

The wavelength is generally measured in micrometers (microns) instead of meters.

DIGITAL SIGNALS

Apart from representing the information in analog signal we can also represent the information in digital signal.

For example, a 1 can be encoded as a positive or high voltage and a 0 as zero or low voltage.

In digital signal we have more than two levels.

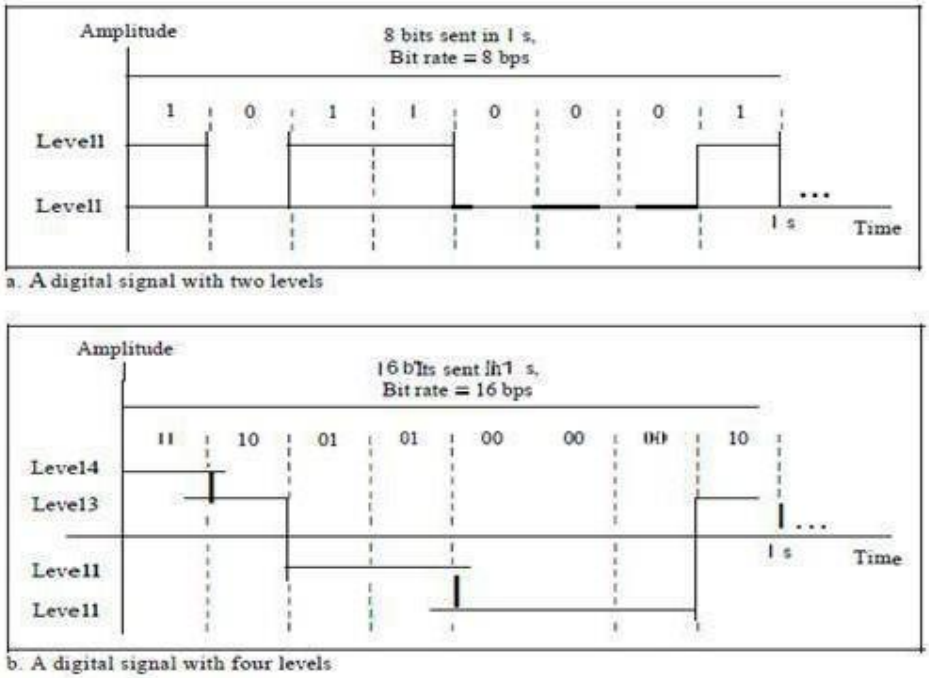


Figure 1.6 Shows two signals, one with two levels and the other with four.

In the above figure we are sending 1 bit per level in part a of the figure 3.16 and we are sending 2 bits per level in part b of the figure 3.16.

In general, if a signal consists of L levels then, each level needs **log2L** bits.

Types of Communication

In data communication we have 3 types of communications. They are simplex, half-duplex and full-duplex.

Simplex:

In simplex mode we can see that the communication is in unidirectional (one direction only).It is one of the very first serial connections between computers is simplex connections.Simplex is unidirectional i.e., only one of the two devices in the link can transmit the information or data; the other can only receive it.

Keyboards and traditional monitors are some of the examples of simplex devices. The keyboard can only give input, whereas the monitor can only receive the output. The simplex mode of communication will use the entire capacity of the channel to send information or data in onedirection.

Half-Duplex:

In half-duplex mode of communication, each station or both the computer in the link can receive or transmit the information, but not at the same time. If one device is sending the data then, the other can only receive it, and vice versa. In a half-duplex mode of transmission, the entire capacity of a channel is taken over by the device which is transmitting the data at that time. Some of the examples are Walkie-talkies and CB (citizens band) radios.This mode of communication is used when there is no

need of communication in both directions simultaneously.In this the entire capacity of the channel will be utilized for each direction.

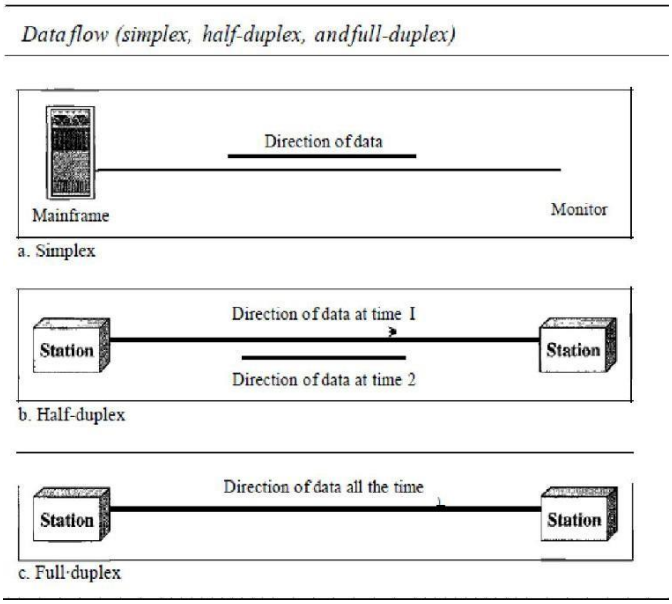


Fig 1.7 Simplex, Half Duplex and Full Duplex

Full-Duplex:

In this full-duplex mode of communication, both the stations can transmit and receive the information simultaneously. It is simply like a two way street with traffic flow in both directions simultaneously. In this full-duplex mode, signals going in one direction will share the capacity of the channel: with the signals which are going in opposite direction. One of the most common examples of full-duplex communication is the telephone network, becausewhen two people are communicating in a telephone

line, both can talk and listen simultaneously. This mode of full-duplex communication is used when the communication is required in both the directions simultaneously. The capacity of the channel will be divided between the twodirections.

Point-to-Point Communication

The dedicated communication link between the two systems or processes is known as a point-to-point connection. Let the wire is connected directly between two systems. Exclusively, to communicate among them, those systems or processes use that wire. Moreover, broadcasting which is the opposite of point- to-communication. It is a system which transmits to many. The telephone call is a circuit-oriented. Where, point-to-point link between two phones. However, the telephone calls are usually multiplexed across telephone company trunks. The circuit may virtual itself. A point-to-point communication session is engaged between the users.

A connection between two systems across a switched network is called End-to-End Connection. For example, the Internet is made up of a mesh of routers. From one router to the next to reach their destinations, packets follow a hop-by-hop path. Whereas Physical point-to-point link between routers will be in each hop. Therefore, multiple point-to-point links is included in a routed path. The end-to-end path is called a virtual circuit that crosses a predefined set of point-to-point links in the ATM and frame relay environment

Fig 1.8 Point to Point Communication

1.3.2 Point-to-Multipoint Communication

A Communication which is fulfilled through a definite and specific form of one-to-many connections which offers several paths from one single location to various locations refers to as Point-to-multipoint (PMP) communication. Point-to multipoint abbreviated as PTMP, P2MP or PMP. Moreover, PMP is used to develop private connectivity to offices in remote locations, long-range wireless backhaul solutions for different sites, and last-mile broadband access. Also, PMP used in Tele-communications. In IP telephony and wireless Internet by means of gigahertz radio

frequencies widely used. These are employed in distribution amenities, huge corporate campuses, school districts, public safety applications, etc. The PMP communication consists of a central base station that supports several subscriber stations. These will offer network access from a single location to multiple locations. These will permit to use the same network resources between them. The base station bridge or Root Bridge is located at the central location. The data passes between the wireless bridgeclients should initially go through the root bridge.

PMP can easily deploy when compared to the deployment of a point-to-point network. Because the equipment should be deployed only at the new subscriber's site. All the remote sites should come within the visibility and range of the base station. Hills, trees and other obstructions make point-to-multipoint nods which are unsuitable for office and residential coverage. PMP systems are classified into a single system and bi-directional systems. A point-to-multipoint network is fit for either customers or backhaul operations which are in need of a high-speed and reliable connection. But,Itworried about paying for unused dedicated capacity. The drawback of point-to-multipoint node topology is not able to interconnect with other nodes due to thedirectional antenna.

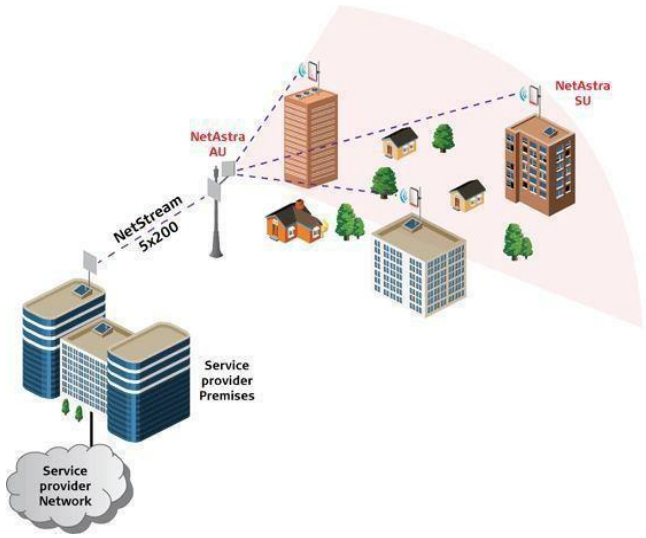


Fig 1.9 Point to Multi Point Communication

1.4Bandwidth

The amount of data that can be transferred from one point to another within a network in a specific amount of time is called as Bandwidth. It is expressed as a bit rate and measures in bits per second (bps). It refers to the transmission capacity of a connection and is an important component when determining the quality and speed of a network.

There are so many ways to measure bandwidth. Some of the measurements are used to calculate current data flow, while others measure maximum flow, typical flow, what considered to good flow.

Bandwidth is also a key notion in several technological fields. In signal processing, it is used to describe the difference between the upper and lower frequencies in a transmission. Examples are such as a radio signal, and measured in hertz (Hz).

Bandwidth is compared to water flowing through a pipe. Bandwidth would be the rate at which water (data) flows through the pipe (connection) under different conditions. We may measure gallons

per minute, instead of bits per second. The amount of water can flow through the pipe represents the maximum bandwidth, while the amount of water that is currently flowing through the pipe represents the current bandwidth.

1.4.1.1 Measurement

Bandwidth measured in bits per second and expressed as bps. However, today's networks have much higher bandwidth than can be expressed by using small units. Now it is common to see higher numbers that are denoted with metric prefixes, such as Mbps, Gbps, Tbps.

Where Mbps is megabits per second, Gbpsis gigabits per second, or Tbps is terabits per second.

M= mega = 1,000 bits

G = giga = 1,000 mega = 1,000,000,000 bits

T = tera = 1,000 giga = 1,000,000,000,000 bits

After terabit, there are petabit, exabit, zettabit, and yottabit, each of them are representing an additional power of 10.

Bandwidth also expressed as bytes per second. This is denoted with a capital B. For example, 10 megabytes per second can be expressed as 10 MB/s or 10 MBps.

One byte is equal to eight bits.

Thus, 10 MB/s = 80 Mb/s.

The same metric prefixes can be used with bytes as with bits. Thus, 1 TB/s is one terabyte per second.

Bit Rate

Bit rate governs the speed of data transmission. A very necessary thought in data communication is how briskly we can send data, in bits per second, over a channel. Bit rate depends upon 3 factors they are,

The bandwidth available
Number of levels in digital signal
The quality of the channel – level of noise

Noiseless Channel

Nyquist Bit Rate –The Nyquist bit rate formula defines the theoretical maximum bit rate for a noiseless channel.

BitRate = 2 * Bandwidth * log2(L)

Where **Bandwidth** is the bandwidth of the channel, L is the number of signal levels used to represent data, and **Bit Rate** is the bit rate in bits per second.Bandwidth is a fixed quantity. So, It doesn't change. Thus, the data rate is directly proportional to the number of signal levels. Increasing in levels of a signal may decrease the reliability of the system.

Examples:

Input1 : Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. What can be the maximum bit rate? **Output1 :** BitRate = 2 * 3000 $\log_2(2)$ = 6000bps

Noisy Channel :

Shannon Capacity –

In reality, The channel is always noisy. So, We cannot have any noiseless channel. Shannon capacity isused to determine the theoretical highest data rate for a noisy channel Capacity = bandwidth * $\log_2(1 + SNR)$

Examples:

Input1 : A telephone line normally has a bandwidth of 3000 Hz (300 to 3300 Hz) assigned for data communication. The SNR is usually 3162. What will be the capacity for this channel? **Output1 :** C = 3000 * $\log_2(1 + SNR)$ = 3000 * 11.62 = 34860 bps

1.5 Communication Channel

A transmission **medium** can be defined as anything that can carry information from a source to the destination. Let us consider an example. The transmission medium for two people having a dinner conversation is that the air. The air is also used to convey the message in a smoke signal or semaphore. For a written message, the transmission medium would be a mail carrier, a truck, or an airplane.

In data communications, the definition of the information and the transmission medium is more distinct. The transmission medium is generally free space, metallic cable, or fiber-optic cable. The information is a signal which is the result of a transformation of data from another form.

Guided Media

Guided media provide a channel from one device to another which includes twisted-pair cable, coaxial cable, and fiber-optic cable. A signal traveling along any one of these media directed and contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic (copper) conductors which accept and transport signals in the form of electric current. An Optical fiber is a cable which accepts and transports signals in the form of light.

1. Twisted-PairCable

A twisted pair consists of two conductors (normally copper), each conductor with its own plastic insulation and also twisted together as shown below

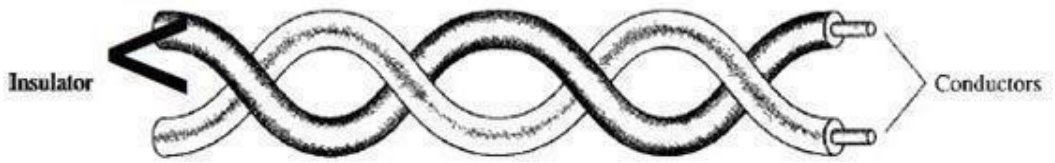


Fig 1.10 Twisted-Pair Cable

One of the wires used to carry signals to the receiver. Moreover, The other is used only as a ground reference. The receiver uses the difference between them. In addition to the signal sent by the sender on one of the wires, interference means noise and crosstalk may affect both wires and will create unwanted signals. If the two wires are parallel, then the effect of the unwanted signals is not same in both wires because they are at different locations related to the noise or crosstalk sources (one is closer and the other is far away). This will result in a difference at the receiver. By twisting the pairs, a balance will maintain. Suppose in one twist, one wire is close to the noise source and the other is afar in the next twist, the reverse is true. Twisting makes it probable that both wires are equally affected by external impacts like noise or crosstalk which means that the receiver calculates the difference between the two and receives no unwanted signals. The unwanted signals mainly canceled out. From the above analysis, it is clear that the number of twists per unit of length has some impact on the quality of thecable.

Applications

Twisted-pair cables used in telephone lines to provide voice and data channels. The local loop-line connects subscribers to the central telephone office commonly consists of unshielded twisted-pair cables. The DSL lines used by the telephone companies toprovide high-data-rate connections, and also use the high-bandwidth capability of unshielded twisted- pair cables. Local-area networks, such as lOBase-T and lOOBase-T also use twisted-pair cables.

2. CoaxialCable

Coaxial cable or coax carries signals of higher frequency ranges than those which are in twisted pair cable because the two media constructed differently. Instead of having two wires, coax has a central core conductor of solid or stranded wire generally copper enclosed in an insulating sheath, which is encased in an outer conductor of metal foil, braid, or a combination of two. The outer metallic wrapping serves as a shield against noise and as the second conductor completes the circuit. This outer conductor enclosed in an insulating sheath, and it is protected by a plastic cover.

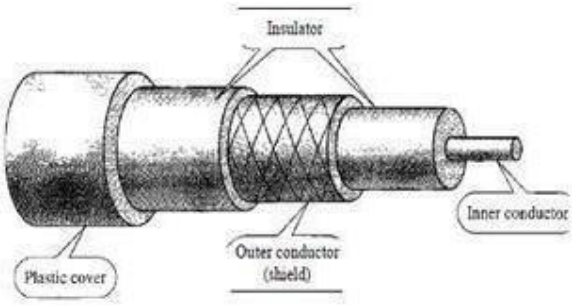


Fig 1.11 Twisted-Pair Cable

Applications

Coaxial cable widely used in analog telephone networks where a single coaxial network could able to carry 10,000 voice signals. After that, It was used in digital telephone networks where a single coaxial cable could able to carry digital data up to 600 Mbps. Coaxial cable in telephone networks largely replaced today with fiber-optic cable. In Cable TV networks, coaxial cables are used. In the traditional cable TV network, the complete network used coaxial cable. Later, cable TV providers replaced most of all the media with fiber-optic cable. Moreover, Hybrid networks use coaxial cable only at the network boundaries, near the consumer places. However, Cable TV uses RG-59 coaxial cable. Another application of coaxial cable is in traditional Ethernet LANs. Due to its high bandwidth, and accordingly high data rate, It was chosen for digital transmission in Ethernet LANs.

Fiber Optic Cable: Thefiber-optic cable is made up of glass or plastic and transmitting the signals in the form of light. To understand the optical fiber, firstly, we need to explore several features of the nature of light. We know that the Light travels in a straight line. It is moving through a single uniform If a ray of light is traveling through one substance suddenly enters then the another substance of the different density, then the ray changes direction. As shown in the figure, how a ray of light changes its direction while going from a denser to a less densersubstance.

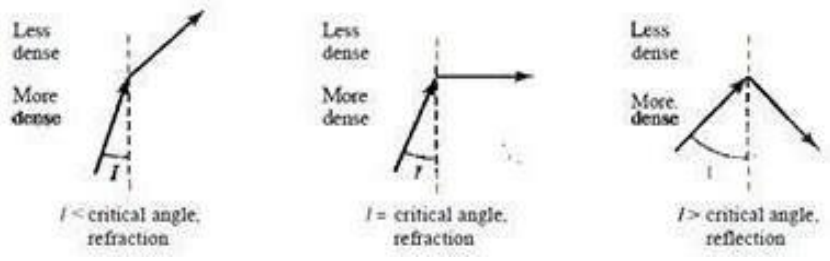


Fig 1.12 Bending Light Wave

As shown below, if the angle of incidence I means the angle of the ray makes with the line perpendicular to the interface between the two substances which is less than the critical angle, that the ray refracts and moves closer to the surface. If the angle of incidence I is equal to the critical angle, then the light bends along the interface. If the angle is greater than the critical angle, then the ray reflects which makes a turn and travels again in the denser substance. More that the critical angle is a property of the substance, and its value differs from one toanother substance.

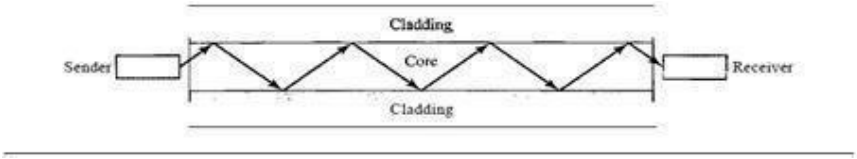


Fig 1.13 Optical fiber

Moreover, Optical fibers use reflection to escort light through a channel. A glass or plastic core surrounds with a cladding of less dense glass or plastic. The difference in density of the two materials that should be a beam of light moving through the core and is reflected off the cladding instead of refracted into it. Observe the above figure.

Cable Composition

The below figure shows the structure of a typical fiber-optic cable. The outer jacket is made up of either PVC or Teflon. Inside the jacket, Kevlar shores to make the cable strong. Kevlar is a strong material which is used in the fabrication of bulletproof property. Below the Kevlar, there is another plastic coating to shield the fiber. The fiber is at the center of the cable which consists of cladding and core.

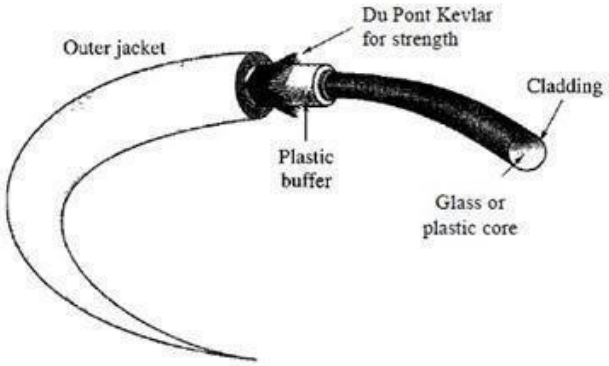


Fig 1.14 Fiber Optics Construction

Applications

Fiber-optic cable is frequently found in backbone networks because its wide bandwidth is more cost- effective. Today, we can able to transfer the data at the rate of 1600 Gbps with wavelength-division multiplexing (WDM),.The SONET network provides a backbone. Moreover, some of the cable TV companies use the both combination of optical fiber and coaxial cable. Thus, it is creating a hybrid network. Where, The Optical fiber is providing the backbone structure while the coaxial cable provides the connection to the user establishments. In Fact, This is a cost-effective configuration. Since the narrow bandwidth requirement at the user end does not explain the use of optical fiber. Local-area networks like 100Base-FX network which is a Fast Ethernet and 1000Base-X also use fiber-opticcable.

Advantages and Disadvantages of Optical Fiber

Advantages

Fiber-optic cable has many advantages over metallic cable either twisted pair or coaxial.

Higher bandwidth. Fiber-optic cable supports splashy higher bandwidths than either twisted-pair or coaxial cable.

Dataratesandbandwidthutilization over fiber-optic cable are not limited by the medium but they are limited by the signal generation and reception technology available.

b. **Less signal attenuation**. Fiber-optic transmission distance is crucially greater than guided media. A signal can run for 50 km without requiring the regeneration. We need repeaters every 5 km for coaxial or twisted-paircable.

c. **Immunity to electromagnetic interference**. Electromagnetic noise doesn't affect the fiber-optic cables.

d. **Resistance to corrosive materials**. Glass is more resistant to corrosive materials thancopper.

Light weight. Fiber-optic cables are lighter than the coppercables.

Greater immunity to tapping. Fiber-optic cables are more resistant to exploiting than copper cables. Copper cables create antenna effects that can be easily exploited.

Disadvantages

There are some disadvantages in the use of optical fiber.

a. **Installation and maintenance**. Fiber-optic cable is proportionally new technology. Its installation and maintenance require skill or competence that is not yet availableeverywhere.

b. **Unidirectional light propagation**. Propagation of light is unidirectional. If we need bidirectional communication, two fibers will be needed.

c. **Cost**. The cable and the interface are comparatively more high cost than the other guided media. If the demand for bandwidth is not high, then frequently the use of optical fiber cannot berationalized.

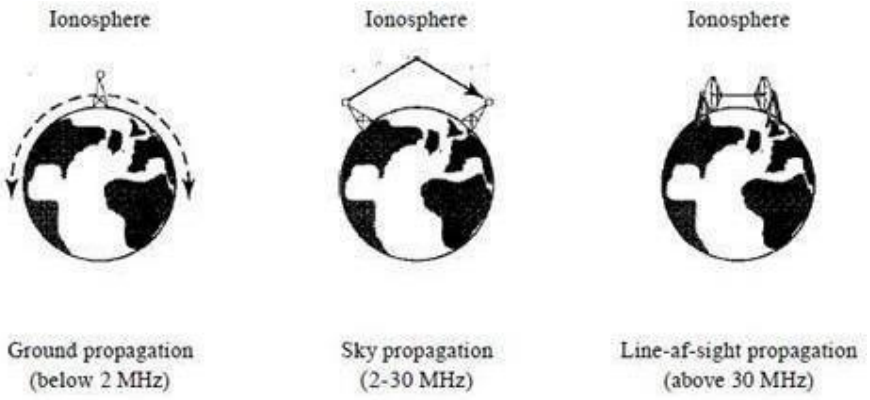


Fig 1.15 Propagation Methods

Unguided Media: Wireless

Unguided media transports the electromagnetic waves without using the physical conductor. This type of communication is frequently referred to as wireless communication. Signals normally broadcast through free space. Thus are available to anyone who has device which is capable of receiving them. Unguided signals can travel from the source to destination in different ways. They are ground propagation, sky propagation, and line-of-sight propagation. Radio waves will travel through the lowest portion of the atmosphere, clutching the earth in ground propagation. These low- frequency signals spread in all directions from the transmitting antenna and follow the curve of the planet. The distance depends on the amount of power in the signal.

The greater the power, the greater the distance. Higher-frequency radio waves radiate upward into the ionosphere where they reflected back to earth in sky propagation. This type of transmission permits for greater distances with minimum output power. In line-or-sight propagation,very high-frequency signals transmitted in straight lines directly from antenna to antenna. Antennas must be directional, facing each other, and either tall or close enough together not to be affected by the inclination of the earth. Line-of-sight propagation is difficult because radio transmissions cannot be focused entirely.

1. Radio Waves

Radio Waves are ranging in frequencies between 3 kHz and 1 GHz. Radio waves are the Omnidirectional. When an antenna transmits radio waves, then these waves are propagated in all directions which mean that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves can be received by any receiving antenna. The Omnidirectional property has also a disadvantage. The radio waves transmitted by one antenna are vulnerable to interference by another

antenna may send signals using the same frequency or band. Radio waves can propagate in the sky mode and those can travel long distances. This will make radio waves a good candidate for long-distance broadcasting such as AM radio. Radio waves with low and medium frequencies can transfix walls. AM radio can receive signals inside a building, this is an advantage. It is also a disadvantage why because we cannot sheathe a communication to just inside or outside a building. The radio wave band is narrow less than 1 GHz when compared to the microwave band. This band is divided into sub bands where the sub bands are also narrow, essential to a low data rate for digital communications.

Omni directional Antenna

Radio waves use Omni directional antennas send out signals in all directions. We can have several types of antennas based on the wavelength, strength, and the purpose of transmission,. Observe below figure shows an Omnidirectional antenna.

Applications

The characteristics of Omnidirectional ,radio waves make them useful for multicasting. There is one sender but many receivers. The examples of multicasting are AM and FM radio, television, maritime radio, cordless phones, and paging

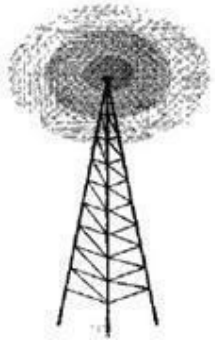


Fig 1.16 Omni directional Antenna

2. Microwaves

Frequencies of Electromagnetic waves between I and 300 GHz are called microwaves which are unidirectional. When an antenna transmits microwave waves which can be narrowly focused. This means that the sending and receiving antennas need to be arranged in an order. It has a disadvantage. A pair of antennas aligned without interfering with another pair of aligned antennas. The following describes some of the characteristics of microwave propagation:

a. Microwave propagation is line-of-sight. The towers with the ascend antennas need to be in direct sight of each other. The towers that are far apart should be very tall. The inclination of the earth , and other blocking interferences do not allow two short towers to communicate by using microwaves. Repeaters are frequently needed for long distancecommunication.

Very high-frequency microwaves cannot transfix the walls. This will be a disadvantage if receivers are insidebuildings.

The microwave band is comparatively wide almost 299 GHz. Therefore wider sub bands can assign, and a high data rate is possible

For the Use of some portions of the band requires permission fromauthorities.

Unidirectional Antenna

Microwaves need unidirectional antennas to send out signals in one direction. Two types of antennas are used for microwave communications. They are the parabolic dish and the horn. A parabolic dish antenna is based on the geometry of a parabola. Every line parallel to the line of symmetry reflects the curve at angles such that all the lines intersect in a common point called the focus. Moreover, The parabolic dish works as a channel catching the wide range of waves and directing them to a common point. In the Similar Way, More of the signals are recovered than possible with a single-point receiver. Outgoing transmissions are broadcast through a horn aimed at the dish. These microwaves will hit the dish and are bounced outward in turnaround of the receipt path. A horn antenna looks like a gigantic scoop. Outgoing transmissions are broadcast up resembling a handle and bounced outward in a series of narrow parallel beams by the curved head. Received transmissions collected by the scooped shape of the horn similar to the parabolic dish, and are bounced down into thestem.

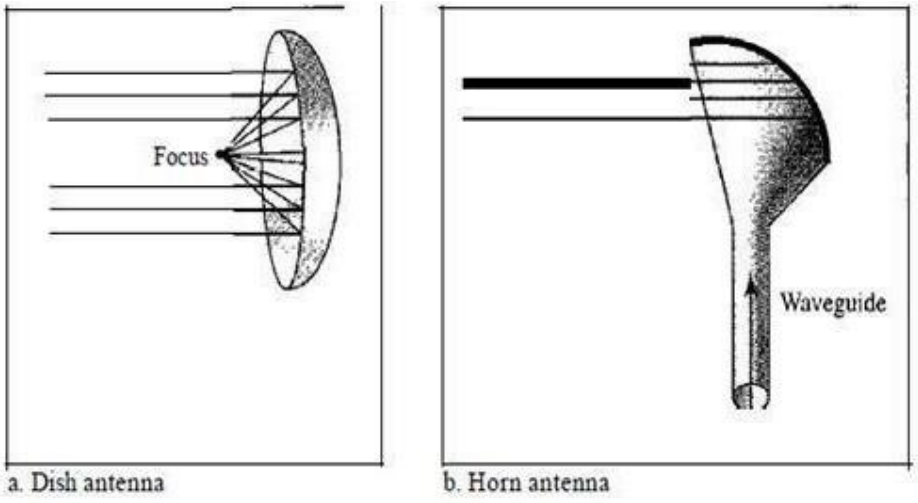


Fig 1.17 Unidirectional Antenna

3. Infrared

Infrared waves with frequencies from 300 GHz to 400 THz and the wavelengths from 1 mm to 770 nm can be used for short-range communication. Infrared waves with high frequencies cannot transfix

walls. This will prevent the interference between one system and another. A short-range communication system in one room will not be affected by another system in the next room. When we use our infrared remote control, we do not cramp with the use of the remote by our neighbors. However, this makes infrared signals useless for long-range communication. In addition to this, we cannot use infrared waves outside a building because of the sun rays contain infrared waves and can interfere with the communication.

Applications

The infrared band with 400 THz has an excellent potential for data transmission. Such a wide bandwidth can be used to transfer the digital data with a very high data rate. The Infrared Data Association (IrDA), an association for sponsoring the use of infrared waves, has developed the standards for using these signals for communication between the devices like keyboards, mice, PCs, and printers. Some manufacturers provide a special port called the IrDA port which allows a wireless keyboard to communicate with a PC. The standard originally defined a data rate of 75 kbps for the distance up to 8 m. The standard defines a data rate of 4Mbps recently.

Infrared signals defined by IrDA transmit through line of sight. The IrDA port on the keyboard shouldpoint to the PC for the transmission to occur.

Summary:

Data communication is the transmitting of electronic data over some media. The media might be cables, microwaves.

Four basic elements will need for any communication system.

Sender

Medium

Receiver

Protocols

Data may be transfer from one device to another by means of some communication media. The electromagnetic or light waves transmits data from one device to another device in the form of signals.

The transmission of data in the form of electrical signals or continuous waves is called analog signal or analog data transmission.

The transfer of data in the form digital signal or digital data transmission. Digital signals consist of binary digits 0 & 1.

A communication network is a set of devices also referred as nodes and connected by communication links. It will provide a service and the transfer of information between users located at different geographical points.

A point-to-point connection is a dedicated communication link between two systems or processes.

Point-to-multipoint (PMP) communication refers to communication that is fulfilled through a distinct form of one-to-many connections,

Simplex communication is unidirectional communication which is permanent.

Half-duplex link can communicate in only one direction at a time. Two way communications is possible but not concurrently.

Full duplex communication is two-way communication achieved over a physical link and has the ability to communicate in both directions concurrently.

Bandwidth in computer networking refers to the data rate which is supported by a network connection or interface. One most commonly expresses bandwidth in terms of bits per second (bps).

A twisted pair consists of two insulated copper wires about 1 mm thick. The wires are twisted together in a helical form.

Microwave communications a method of transmitting information or energy by the use of radio waves whose wavelengths are measured in small numbers of centimeter called as microwaves.

Optical fiber carries more information than conventional copper wire and is in general not subject to electromagnetic interference and the need to retransmit signals.

Synchronous data transmission is a data transformation method in which a continuous stream of data signals is guided by timing signals which are generated by an electronic clock to ensure that the transmitter and the receiver are in synchronized with one another.

Asynchronous transmission works in spurts and must insert a start bit before each data character and a stop bit at its termination to inform the receiver where it begins and ends.

The Isochronous format for data transmission is a procedure or protocol in which each information character or byte is individually synchronized or framed by the use of Start and Stop Elements, also referred to as start bits and stop bits.

A parallel interface refers to a multiline channel, each line capable of transmitting several bits of data simultaneously

A serial interface uses a serial port, a single line capable of only transmitting one bit of data at a time; a computer mouse connection is a good example.

In simplex mode the communication can take place only in one direction. The receiver receives the signal from the transmitting device.

In half-duplex mode the communication channel is used in both directions, but only in one direction at a time.

In full duplex the communication channel is used in both directions at the same time.

Multiplexing is a technique of concurrent transmission of multiple signals over a single data link.

Space-division multiplexing is achieved by multiple antenna elements forming a phased array antenna, space-division multiplexing simply implies different point-to-point wires for different channels.

FDM achieves the combining of many digital signals into one medium by sending signals in several specific frequency ranges over a single medium.

Time-division multiplexing (TDM) is a digital (analog/rare cases) technology.

Code division multiplexing (CDM) or spread spectrum is a class of techniques where different channels simultaneously share the same frequency spectrum, and this spectral bandwidth is much higher than the bit rate or symbol rate.

Short Answer Type Questions

1. What is data communication ? List types of data communications.
2. Write various modes of data transmission.
3. Define bandwidth.
4. List various communication channels.
5. What is Asynchronous Data Transmission ? Write various forms of Data Transmission ?

Long Answer Type Questions

1. Explain various types of data communications.
2. Write about transmission modes
3. Explain synchronous and Asynchronous and Isochronous
4. Explain about various communication channels.

UNIT 2

Network types and Topologies

Learning Objectives

After studying this unit, the student will be able to

Understand about computer network and its usage.

Understand about the advantages of computer network.

Understand about different types of networks.

Understand about different Network topologies

Introduction:

Computer network is the set of computers connected together for the purpose of resources sharing. Internet connection is one of the most common resources shared today. Some of the other shared resources are printer and a file server. We can consider internet itself as a computer network.

Computer Network Definition:

A computer network is a set of computers interconnected with each other. Computers which are in a network are known as nodes.

We can say two computers are interconnected only when they exchange information. The connection between those computers can be acquired via a copper wire or fiber optics or communication satellite or microwaves.

The connection between computers may be wired (most commonly the Ethernet cable) or wireless(through radio waves).

Applications of computer networks:

Resource Sharing:

The main goal of resource sharing is to make all the equipment (like printers etc.) and especially data available to everyone in the network without depending on the physical location of the user and the resource.

Server-Client model:

Every company will maintain some database either to store the information of the employee or to store the company information and if employeeswant to access it remotely, can access by using this model. In this model, the entire data is stored insome computers known as Servers. These computers are centrally stored and maintained by the system administrator. The employees will have simple machines, known as Clientson their desks using those systems they can access the data remotely.

Communication Medium:

Computer network can act as a powerful communication medium.Almost every company that has computers which are connected in a network has e-mail (electronic mail) through which they (employees) can communicate.

Commerce:

Because of computer networks a variety of online business transactions, popularly known as E-commerce. Some of the uses of this e-commerce are buying or selling items, managing bank accounts, paying bills, transferring funds etc.

Advantages of Computer network

1) Resource Sharing

Resource sharing is one of the major advantages of computer network. This benefit is a boon for many larger companies that need to share huge numbers of resources among many people. Now-a-days the technology is mainly involved in computer-based work; it is guaranteed that the resources they want to get accesscan be completely accessed by connecting to the computer network which other employees are already using.

Enhance in Storage Capacity

Computer network means interconnection of different computers by which we can easily share many files. By the help of this computer network the problem of storage capacity is resolved. Since you will be sharingyour information, files and other resources with different people, you have to ensure that all the data and content are properly stored in the system.

File Sharing

Because many computers are interconnected in a network, a user can share files from one workspace to another workspace easily.It is a time saving process. Apart from that, every employee in an organization can access the files and they can update the information in the files stored in the company database, by which it will be upto-date.

4) Highly flexible

It is very flexible, as it allows users to explore everything about many important things, like software without affecting the functionality. Apart from that, people can access all the information they want.

5) Increased Cost Efficiency

There are many software applications are available in the market which are costly and take time for installation. Computer networks resolve this issue as the software can be stored or installed on a system or a server and can be used by the different workstations.

6)Reliability

Reliability is process of backing up of information. Due to some reason equipment crash, and so on, the information gets undermined or inaccessible on one PC, another duplicate of similar information is accessible on another workstation for future use, which prompts smooth working and further handling without interruption.

2.1.1 Disadvantages:

Security Issues:

One of the major limitations of computer network is the security. If it is an individual computer then physical access will be necessaryfor any kind of data stealing. However, if a computer is connected in a

network, simply a computer hacker can get illegal access by using different hacking tools. In order to prevent such type of security issues in many big organizations, they are using much network security software.

2) More presence of computer viruses and malwares

There will be some instances where the stored files are corrupted due to some computer viruses. Any computer system in a network gets affected by computer virus, there is a possible threat of other systems getting affected too. Viruses get spread on a network easily because of the interconnectivity of workstations. Such spread can be dangerous if the
computers have important database which can get corrupted by the virus. The network administrators must conduct regular check-up for the system, along with the stored files.

Dependent on the Main File Server

If the main File Server of the network breaks down, then all the systems will become useless. If it is a big network then the File Server must be a powerful computer, which makes it expensive.

4) Expensive Set Up

The initial set up cost of a computer network can be high depending on the number of computers to be connected. Costly devices like routers, switches, hubs, etc., can add up to the bills of a person trying to install a computer network. He will also have to buy NICs (Network Interface Cards) for each of the workstations, in case they are not inbuilt.

Lack of Robustness

If a PC system's principle server separates, the whole framework would end up futile. Also, if it has a bridging device or a central linking server that fails, the entire network would also come to a standstill. To manage these issues, gigantic systems ought to have a ground-breaking PC to fill in as document server to influence setting to up and keeping up the system less demanding.

6) Lacks of independence

PC organizing includes a procedure that is worked utilizing PCs, so individuals will depend a greater amount of PC work, rather than applying an exertion for their jobs that needs to be done. Beside this, they will be subject to the primary document server, which implies that, in the event that it separates, the framework would end up futile, making clients inactive.

Types of Networks

Network allows different computers to connect and communicate with each other through a medium.

Major types of network mediums are LAN, MAN and WAN. Users will select any one of the medium depending on the area of network they want to cover. One of the major differences between them is the geographical area they cover.

LAN covers the smallest geographical area. MAN will cover an area greater than LAN and WAN will cover the largest geographical area than all. We have other types of Computer Networks also like PAN,SAN,EPN,VPN...etc.

Types of Networks

Local Area Networks (LAN)
Personal Area Networks (PAN)
Home Area Networks (HAN)

Metropolitan Area Networks (MAN)
Wide Area Networks (WAN)
Campus Networks
Enterprise Private Networks
Backbone Networks (BBN)
Global Area Networks (GAN)
SAN (Storage Area Network)
EPN (Enterprise Private Network)
VPN (Virtual Private Network)
The Internet

Local Area Network:

It is also known as LAN. It is mainly designed for small physical area covered organizations. Main advantage of using LAN is, it is easy to design and to troubleshoot. Standalone computers and workstations can be connected to each other by LANs.

LAN is a simple network. It is just theinterconnection of at least two computers, for sharing files and network in between them, while if the connection is for entire building then it will become a complex problem.

Other uses of LAN network is to share resources like shared hard-drive,printers etc.

Characteristics of LAN:

LAN is a private network. It is not subject to any tariffs or any other regulatory controls

When compared to the WAN, LAN's will operate at relatively high speed.

In LAN we have different types of Media Access Control (MAC) methods in a LAN, mainly Ethernet, Token ring.

LANs will connect computers in a single block,building or campus, i.e. they are operated in a limited geographical area.

Applications of LAN:

In LAN we can make one computer as a server which serves all the remaining computers commonly known as clients.

By using LANs we can make a local connection between the workstations in a building to let the employees or the other people of that organization to communicate with each other without any internet facility.

By using LANs we can share common resources like printers etc.

Personal Area Networks:

It is also known as WPAN. It is a low-range and wireless network which covers an area of some meters. We can see this sort of network in linking peripheral devices (like cell phones, printers and home appliances). It may be the connection between any two nearby computers without any hard-wired connection.

Home Area Network (HAN)

A **h**ome **a**rea **n**etwork (**HAN**) is a network contained within a user's home that connects a person's digital devices, from multiple computers and their peripheral devices to telephones, VCRs, televisions, video games, home security systems, smart appliances, fax machines and other digital devices that are wired into the network.

Metropolitan Area Network:

It was introduced in 1980s. It is also known as MAN and uses the same technology as LAN. It is developed to extend its coverage over the entire city. It can be the connection of number of LANs into a larger network or simply it can be a single cable. It is mainly handled and operated by single private company.

It may be the connectionbetween two or more computers which are apart but exist in the same or different cities. It covers a larger geographical area and it may also serve as an ISP (Internet Service Provider). MAN is mainly designed for those customers who need high-speed internet connectivity.

Characteristics of MAN:

MAN generally covers towns and cities (50 km)

The communication medium used for Metropolitan Area Network are cables, optical fibers etc.

Advantages of MAN:

Main advantage of using MAN is it provides efficient and fast communication through high-speed medium, such as fiber optic cables.

It acts as a back bone for larger networks and provides a greater access to WANs.

In MAN for transmission of data they use dual bus which is used to transmit the data in both the directions at the same time.

Wide Area Network:

It is also known as WAN. It can be private or may be public leased network. If we want to share the network throughout the country or any larger geographical area then this WAN is used. It is not an easy process to design and maintain WAN.

The data rate in WAN is slow about a 10th of LAN's speed, because it involves greater distance and many number of servers and terminals etc. There are two types of WAN: Switched WAN and Point-to-Point WAN

Characteristics of WAN:

Generally WAN covers greater distances(like states, countries, continents).

In WAN we use satellite, public telephone networks as communication medium which are linked by routers.

Advantages of WAN

One of the main advantages is instant messaging to anyone in the network.
These messages include images, audio.

Costly things (like printers or phone lines to the internet) can be commonly shared by all the computers connected in the network.

Campus network

A campus network is a computer network made up of an interconnection of local area networks (LAN's) within a limited geographical area. The networking equipments (switches, routers) and transmission media (optical fiber, copper plant, Cat5 cabling etc.) are almost entirely owned (by the campus tenant / owner: an enterprise, university, government etc.).

Example

University campus-based campus network, the network is likely to link a variety of campus buildings including; academic departments, the university library and student residence halls.

Enterprise private network

An enterprise private network is a network build by an enterprise to interconnect various company sites, e.g., production sites, head offices, remote offices, shops, in order to share computer resources.

Backbone network

A Backbone network (BBN) is a part of a computer network communications that interconnects different pieces of network and providing a path for the exchange of information between different LANs or sub networks. A large corporation that has many locations may have a backbone network that ties all of the locations together

Example

if a server cluster needs to be accessed by different departments of a company that are located at different geographical locations. The pieces of the network connections (for example: ethernet, wireless) that bring these departments together is often mentioned as network backbone. Network congestion is often taken into consideration while designing backbones.

Global area network (GAN)

A global area network (GAN) is a network used for supporting mobile communications across an arbitrary number of wireless LANs, satellite coverage areas, etc. The key challenge in mobile communications is handing off the user communications from one local coverage area to another local coverage area.

Storage area network(SAN)

This network connects servers directly to devices to store data. SAN moves storage resources off the common user network and reorganizes them into an independent, high-performance network. So, each server is allowed to access shared storage. This can involve Fibre-channel connection, similar to Ethernet, to handle high-performance disk storage for application.

Enterprise private network(EPN)

It is a computer network built to share computer resources among different sites (such as production sites, offices and shops) of a business. Some of the advantages of an enterprise private network are:

The messages are secure because they are encrypted.
They are cost effective and scalable.
They help to centralize IT resources.
They enable business continuity.

Virtual private network(VPN)

It is an extended private network which spreads over the internet. Users can send and receive data across shared or public networks. It uses public wires- usually the internet to connect to a private network usually a company private network. The benefit of a secure VPN is its level of security to the connected systems, whereas the other network infrastructure alone cannot provide it.

The Internet

The Internet is a global system of interconnected governmental, academic, corporate, public, and private computer networks. It is based on the networking technologies of the Internet Protocol Suite. It is the successor of the Advanced Research Projects Agency Network (ARPANET) developed by DARPA of the United States Department of Defense. The Internet is also the communications backbone underlying the World Wide Web (WWW).

Definition of Network Topologies

A Network Topology is nothing but the pattern in which the computers or various hardware elements are connected in a computer network(mostly LAN) is called a network topology. It defines how the systems (nodes) in the computer network are connected with each other

The network topologies can be physical or logical. Physical topology refers to the physical shape or design structure of a network including the nodes, devices and cables. Logical topology refers to how data is transferred or exchanged between computers within the network. Generally, the term network topology refers to the physical topology.

Types and Structure of Topologies

Different types of Network topologies are:

Bus Topology
Ring Topology
Star Topology
Mesh Topology
Tree Topology
Hybrid Topology
Internet

Explanation with advantages and disadvantages:

Bus Topology:

Bus Topology is one of the network topologiesin which a single cable acts as the backbone to link all the computers (nodes) in the network.

If the cable has exactly has two end points then it is known as Linear Bus Topology.

Data transmission is possible only in one direction at a time.

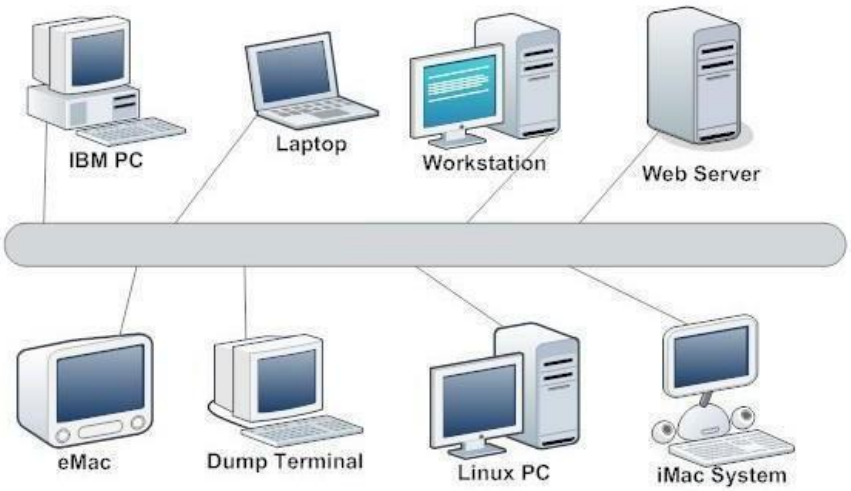


Fig 2.1 Bus Topology

Features of Bus Topology:

Bus topology transmits data in one direction only.

Every computer or node is connected to a single back bone cable.

Advantages of bus topology:

If we want to connect N devices in a network by using bus topology methodology, then the number of cables required for connection is 1 which is called as the backbone cable and apart from that one cable we also need N drop lines (connection line between the computer and the main backbone cable).

This is cheaper when compared with other topologies because of the low cost of the back bone cable.

Disadvantages of Bus Topology

If the backbone cable fails then the whole network connected to that back bone cable will fail.

The back bone cable will have a limited length.

RING Topology:

It is known as ring topology because of the arrangement of the nodes in it. In this the nodes or computers are arranged in ring shape.

In this topology, each node isconnected to exactly two neighboring devices, which creates a circular network structure. If one node tries to communicate withanother node which is not adjacent to it then, the data has to travel through all the nodes present in between them.

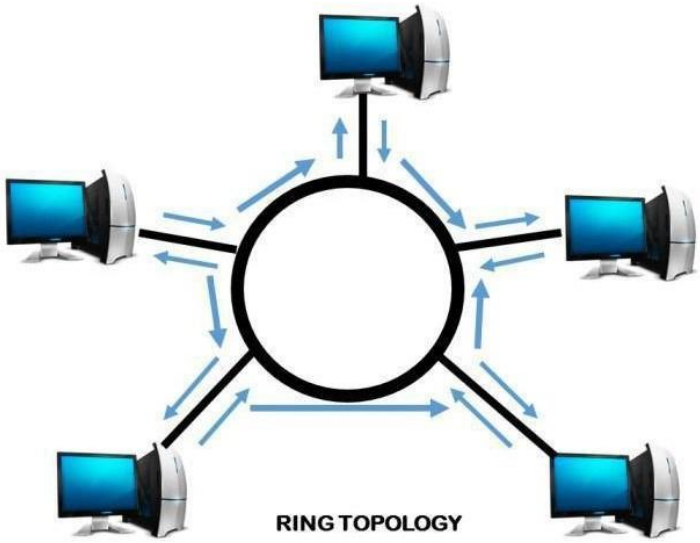


Fig 2.2 Ring topology

Features of Ring Topology

Consider an example: If the first node wants to send data to the last node and assume that the number of intermediate nodes are 99, then the data has to pass through all the 99 nodes to reach the destination node.In the process of transferring data between all the 99 nodes some data may lose to prevent that data loss large number of repeaters are used in the network.

In Ring topology the transmission of data is in one direction only. If we want to make it bidirectional then we have to establish two connections between each node.It is known as Dual Ring Topology.

In Dual Ring Topology, we have two ring networks, and data flow is in opposite direction.The main advantage of this Dual ring topology is, if one ring is failed in the network then, the second ring will act as the backup.

Advantages of Ring Topology:

In this ring topology the possibility of collision is very less.
For installation it takes very less amount as well as to expand also.

Disadvantages of Ring Topology:

Troubleshooting is difficult.

Simple addition or deletion of node will disturb the entire network.

If one node or computer is failed then it will disturb the entire network.

STAR Topology:

In Star topology all the computers will be connected to a single hub individually by a cable. This hub will be the central node and all others nodes will be connected to this central node

The hub, it may be Passive hub or Active hub (types of hubs will be discussed in unit-3).

Active hubs consist of repeaters in them.

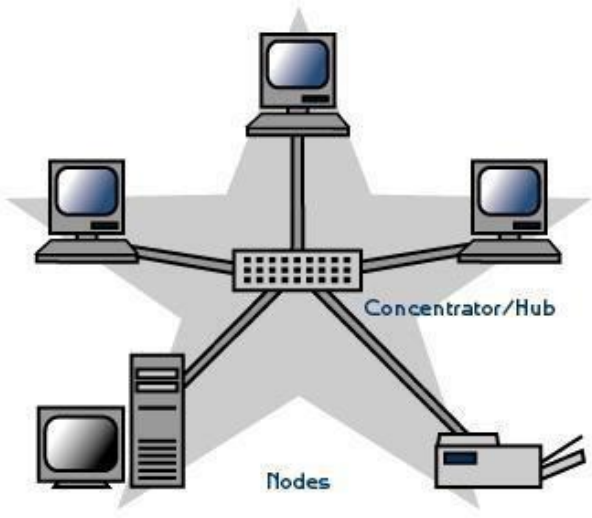


Fig 2.3 STAR topology

Features of Star Topology

Every node or computer in the network has its connection to the hub.

Hub,mainly Active hubs acts as a repeater for data.

Advantages of Star topology:

If we want to connect N devices in a network in star topology methodology, then we need N cablesto connect. So, it is easy to establish the connection.

If the central hub on which the entiretopology depends on, fails then, the whole network will be crashed.

Installation cost is high.

Entire performance of the network is based on the central hub.

Mesh Topology

In Mesh topology, a node has point-to-point connection with each and every node.

In this, every node is connected to remaining nodes in the network through a particular cable.

If there are n devices in the Mesh topology then the number of physical channels will be n(n-1)/2.

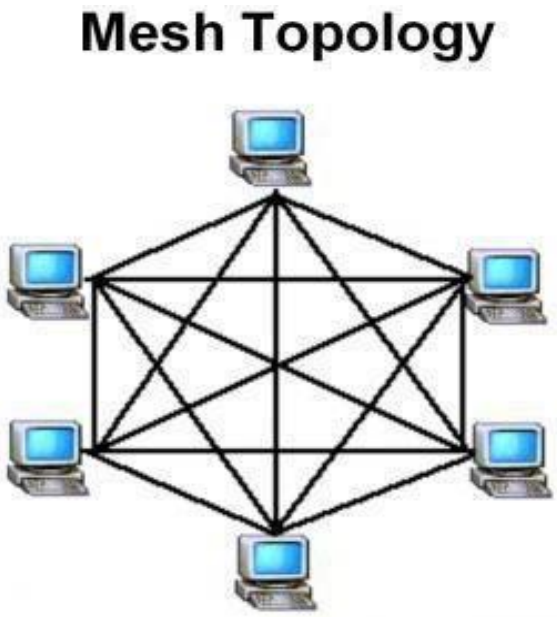


Fig 2.4 Mesh Topology

We have two types of Mesh Topologies. They are:

Partial Mesh Topology: In this type of topology some of the nodeswill be connected in the same manner like the mesh topology but, some nodes are only connected to two or three other nodes.

Full Mesh Topology: In this type of topology each and every node in the network must and should be connected to all the other nodes or devices in the network.

Features of Mesh Topology

Robust

Fully connected.

Advantages of Mesh Topology

Each node had an individual connection to all the remaining nodes because of that they can send their own data load.

Easy to troubleshoot.

It provides privacy and security of data.

Disadvantages of Mesh Topology

It is difficult to Install and configure.

Highly expensive because, bulk wiring is required, hence it is suitable for less number of computers or nodes.

Hybrid Topology:

Hybrid topology is the collection of two or more different topologies which are discussed above. In

the below example diagram you can find the star topology, ring topology and the bus topology

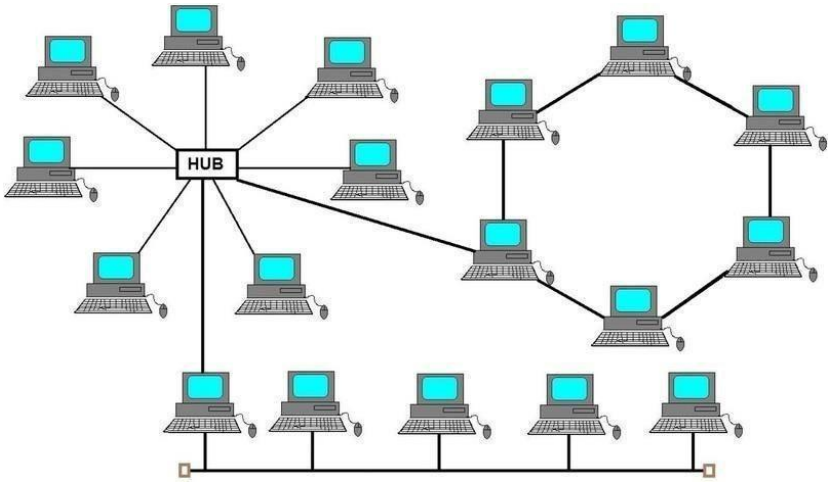


Fig 2.5 Hybrid topology

Features of Hybrid Topology

Hybrid topology is the combination of two or topologies which are same or different

It is the combination of different topologies because of that it also consist of the advantages and disadvantages of that particular topologies

Advantages of Hybrid Topology

Trouble shooting is easy.

It is effective and flexible.

Its size can be increased easily.

Disadvantages of Hybrid Topology

Design will be complex

Cost effective.

Tree topology:

Tree Topology is nothing but a special type of structure in nodes or computers are arranged like the branches of a tree.

In this tree topology, there will be only one connection is established between any two connected nodes or computers because; any two computers in the networkwill have only one mutual connection. It forms parent and child hierarchy.

Features of Tree Topology

It will be the perfecttopology if workspaces are located in groups.

It is mainly used in Wide Area Network (WAN).

Advantages of Tree Topology

Extension of nodes is possible and it is an easy.

It is easy to maintain.

It is easy to detect errors.

Disadvantages of Tree Topology

It is highly expensive.

In this, if more nodes are added then, the maintenance will be difficult.

If the central hub fails then, the network will fail

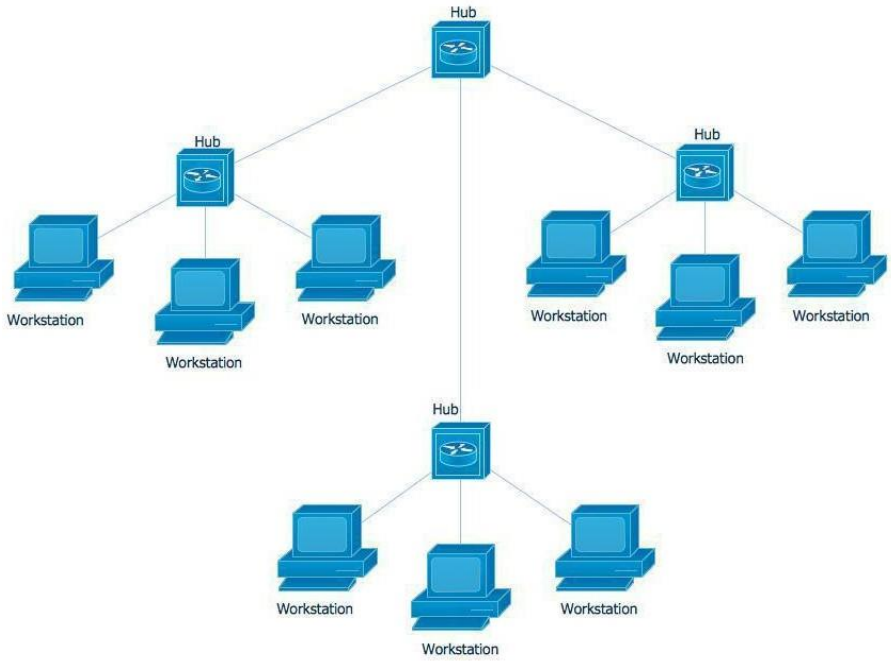


Fig 2.6 Tree Topology

The Internet

The Internet is a worldwide collection of computer networks,cooperating with each other to exchange data using a common software standard.
Through telephone wires and satellite links, Internet users can share informationin a variety of forms.

The size, scope and design of the Internet allows users to

Connect easily through ordinary personal computers and local phonenumbers.
Exchange electronic mail (E-mail) with friends and colleagues withaccounts on the Internet.
Post information for others to access, and update it frequently
Access multimedia information that includes sound, photographicimages and even video
Access diverse perspectives from around the world.

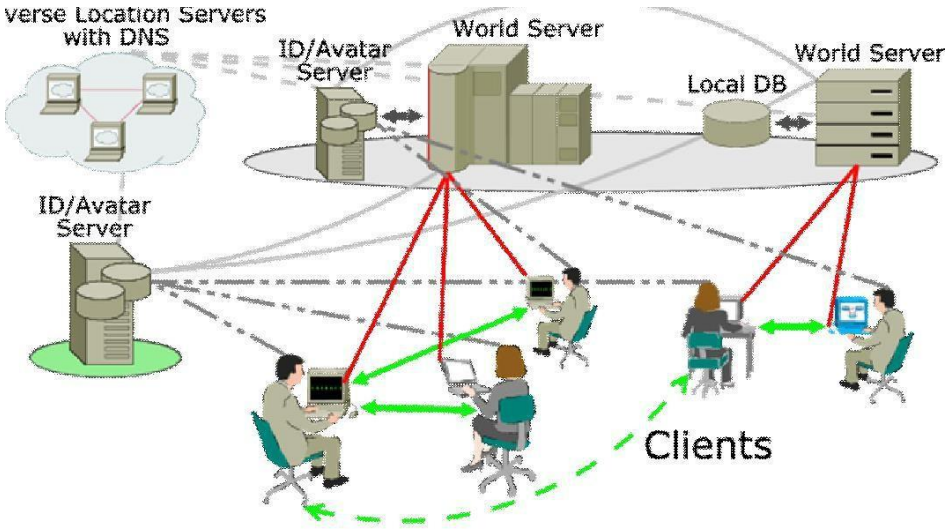


Fig 2.7 The Internet

Advantages

You have information at your fingertips, and you don't even have to know how to spell, or know your ABCs. Someone out there knows the answers to everything you can think of.
You can send mail across the world within minutes where snail mail used to take 3-4 days in the states and 2 weeks overseas.
You are connected to a lot of people who have similar interests if you know the right place to find them on the internet. A lot more difficult in person.

Disadvantages

You cannot always trust the answers you get over the internet when you are looking for information.
You spend too much time in front of a machine, and not enough time in front of "real" people.
You need a credit card if you want to buy something either to pay a third party sight, or to pay direct the company you are interested in. IN person, you can always pay by cash.

Advantages and disadvantages of Network Topologies

Advantages and disadvantages of Network Topologies are depends on the structure of topology to establish network. the following table summarize the advantage and disadvantage of different topologies in the computer networks.

Types of Topology	Advantage	Disadvantage
Bus	• It is cost effective. • Cable required is least compared to other network topology.	• Cables fails then whole network fails. • If network traffic is heavy or nodes are more the

	• Used in small networks. • It is easy to understand. • Easy to expand joining two cables together.	performance of the network decreases. • Cable has a limited length. • It is slower than the ring topology.
RING	• Transmitting network is not affected by high traffic or by adding more nodes, as only the nodes having tokens can transmit data. • Cheap to install and expand	• Troubleshooting is difficult in ring topology. • Adding or deleting the computers disturbs the network activity. • Failure of one computer disturbs the whole network.
STAR	1. Fast performance with few nodes and low network traffic. 2. Hub can be upgraded easily. 3. Easy to troubleshoot. 4. Easy to setup and modify. 5. Only that node is affected which has failed, rest of the nodes can work smoothly.	1. Cost of installation is high. 2. Expensive to use. 3. If the hub fails then the whole network is stopped because all the nodes depend on the hub. 4. Performance is based on the hub that is it depends on its capacity
Mesh	1. Each connection can carry its own data load. 2. It is robust. 3. Fault is diagnosed easily. 4. Provides security and privacy.	1. Installation and configuration is difficult. 2. Cabling cost is more. 3. Bulk wiring is required.
Hybrid	1. Reliable as Error detecting and trouble shooting is easy. 2. Effective.	1. Complex in design. 2. Costly.

	3. Scalable as size can be increased easily. 4. Flexible.	
Tree	1. Extension of bus and star topologies. 2. Expansion of nodes is possible and easy. 3. Easily managed and maintained. 4. Error detection is easily done.	1. Heavily cabled. 2. Costly. 3. If more nodes are added maintenance is difficult. 4. Central hub fails, network fails.

Summary

A computer network is a set of computers interconnected with each other.
Computers which are in a network are known as nodes.

Some of the applications of computer networks are Resource sharing, Server-client model, Communication medium and commerce.

Resource Sharing:The main goal of resource sharing is to make all the equipment (like printers etc.) and especially data available to everyone in the network without depending on the physical location of the user and the resource.

Server-Client model:In this model, the entire data is stored in some computers known as Servers. These computers are centrally stored and maintained by the system administrator.

The employees will have simple machines, known as Clients on their desks using those systems they can access the data remotely.

Communication Medium:Computer network can act as a powerful communication medium. Almost every company that has computers which are connected in a network has e-mail (electronic mail) through which they (employees) can communicate.

Commerce:Because of computer networks a variety of online business transactions, popularly known as E-commerce. Some of the uses of this e-commerce are buying or selling items, managing bank accounts, paying bills, transferring funds etc.

Some of the advantagesof Computer network are: Resource Sharing, Enhance in Storage Capacity, File Sharing, Highly flexible, Increased Cost Efficiency**,** Reliabilityetc,.

Some of the disadvantages of Computer network are: Security Issues, More presence of computer viruses and malwares, Dependent on the Main File Server,Expensive Set Up**,** Lack of Robustness, Lacks of independenceetc,.

Network allows different computers to connect and communicate with each other through a medium.

Major types of network mediums are LAN, MAN and WAN.

LAN is a simple network. It is just the interconnection of at least two computers, for sharing files and network in between them, while if the connection is for entire building then it will become a complex problem.

Characteristics of LAN are: It is a private network. It is not subject to any tariffs or any other regulatory controls
When compared to the WAN, LAN's will operate at relatively high speed.
Advantages of LAN are: In LAN we can make one computer as a server which serves all the remaining computers commonly known as clients.

By using LANs we can share common resources like printers etc.

It is also known as WPAN. It is a low-range and wireless network which covers an area of some meters. We can see this sort of network in linking peripheral devices (like cell phones, printers and home appliances).
A home area network (HAN) is a network contained within a user's home that connects a person's digital devices, from multiple computers and their peripheral devices to telephones, VCRs, televisions etc
It was introduced in 1980s. It is also known as MAN and uses the same technology as LAN. It is developed to extend its coverage over the entire city. It can be the connection of number of LANs into a larger network or simply it can be a single cable.
Characteristics of MAN are: It generally covers towns and cities (50 km)

The communication medium used for Metropolitan Area Network are cables, optical fibers etc.

Main advantage of using MAN is it provides efficient and fast communication through high-speed medium, such as fiber optic cables.

It acts as a back bone for larger networks and provides a greater access to WANs.

It is also known as WAN. It can be private or may be public leased network. If we want to share the network throughout the country or any larger geographical area then this WAN is used.
Characteristics of WAN are: Generally WAN covers greater distances(like states, countries, continents).

In WAN we use satellite, public telephone networks as communication medium which are linked by routers.

One of the main advantages is instant messaging to anyone in the network.
These messages include images, audio.

A Backbone network (BBN) is a part of a computer network communications that interconnects different pieces of network and providing a path for the exchange of information between different LANs or sub networks.

A global area network (GAN) is a network used for supporting mobile communications across an arbitrary number of wireless LANs, satellite coverage areas, etc.

SAN connects servers directly to devices to store data. SAN moves storage resources off the common user network and reorganizes them into an independent, high-performance network. So, each server is allowed to access shared storage

It is an extended private network which spreads over the internet. Users can send and receive data across shared or public networks. It uses public wires- usually the internet to connect to a private network usually a company private network.

The Internet is a global system of interconnected governmental, academic, corporate, public, and private computer networks. It is based on the networking technologies of the Internet Protocol Suite.

A Network Topology is nothing but the pattern in which the computers or various hardware elements are connected in a computer network(mostly LAN) is called a network topology. It defines how the systems (nodes) in the computer network are connected with each other

Different types of Network topologies are:
Bus Topology
Ring Topology
Star Topology
Mesh Topology
Tree Topology
Hybrid Topology
Internet
Bus Topology:

Bus Topology is one of the network topologies in which a single cable acts as the backbone to link all the computers (nodes) in the network.

If the cable has exactly has two end points then it is known as Linear Bus Topology.

It is known as ring topology because of the arrangement of the nodes in it. In this the nodes or computers are arranged in ring shape.

In this topology, each node is connected to exactly two neighboring devices, which creates a circular network structure.

In Star topology all the computers will be connected to a single hub individually by a cable.

This hub will be the central node and all others nodes will be connected to this central node

In Mesh topology, a node has point-to-point connection with each and every node.

In this, every node is connected to remaining nodes in the network through a particular cable.

If there are n devices in the Mesh topology then the number of physical channels will be

n(n-1)/2.

Hybrid topology is the collection of two or more different topologies which are discussed above.

Tree Topology is nothing but a special type of structure in nodes or computers are arranged like the branches of a tree.

In this tree topology, there will be only one connection is established between any two connected nodes or computers because; any two computers in the network will have only one mutual connection.

The Internet is a worldwide collection of computer networks, cooperating with each other to exchange data using a common software standard.

Through telephone wires and satellite links, Internet users can share information in a variety of forms.

Short Answer Type Questions

What is a Network ?

What is internet

3.What is WWW

Write various advantages of networks.

List various disadvantage of Networks.

List various types of Networks.

Expand LAN, WAN,MAN.

Expand BBN,GAN.

What is Network Topology ? List types of topologies ?

Write any two advantages and two disadvantages of topologies.

Long Answer Type Questions

Explain different types of computer networks.

Explain about Internet.

Explain about network topologies.

Explain advantages and disadvantages of different types of network topologies.

UNIT 3

LAN Components

Learning Objectives

After studying this unit, the student will be able to

Know about a Server &Client.
Know the Client-Server Architecture.
Know different Network Interfaces.
Know different types of Modems
Know the functionalities of Adapters.
Know the functionalities of Multiplexers.

Introduction:

LAN ‘s are made up of software as well as hardware components. A Hardware component consists of interface cards in all the machines and cables that bind them together. The software component consist of drivers for all peripherals and network Operating System that manages the network. In this chapter we will discuss in detail about the LAN components.

Server, Client and File Server

Server

A server is a software program, or it may be a computer, in which that program runs, that provides a specific kind of service (Response) to client software running in the same computer or other computers in a network.

A single computer may have many server software applications running in it. Also, a computer can act as a server and as a client simultaneously, this is possible by connecting to itself in the same way that we will connect to a separate computer.

Several large companies have huge number of dedicated servers. The collection of servers in one location is termed as a “Server farm”. If very heavy traffic is expected on a single server, load balancing will usually distribute the requests among the different servers so that no single machine is overloaded.

Different types of Servers:

Web server : It consists of files related to web sites and servers and sends(If requested by the user) them across the Internet to the clients (i.e., by web browsers).One of the most popular web server program is Apache, which is claimed to host more than 68% of all web sites in the Internet.

File server: It is a software, or the combination of hardware and software, that is dedicated for storing files and it will give access to the clients for reading or for writing across the network.

Print server: Print server is the software or the hardware that manages one or more printers.

Network server: Network server manages network traffic.

Name server: Name server maps user and computer names to machine addresses.

Database server: Database server allows clients to interact with a database in a network.

Application server: Application server runs different applications for clients.

Clients:

It may be an application or system that accesses a service that is made available by a server. The server is often (but not always) on different computer system, in such case the clients access the services by the network. The term "Client" was first applied to those devices which were not capable of running their own stand-alone programs, but can interact with remote computers by a network.

Client-Server Model

The client-server model is a distributed communication framework of network; it is the process between service requestors, clients and service providers, servers. This connection is established through a network or the Internet.

The client-server model is a core network computing model and it is also a building functionality for email exchange and Web/database access. Web technologies and protocols developed around the client-server model are as follows:

Hypertext Transfer Protocol (HTTP)

Domain Name System (DNS)

Simple Mail Transfer Protocol (SMTP)

Telnet

A client consists of Web browsers, chat applications, and email software.

Whereas Servers have Web, database, application, chat and email, etc.

The client–server model has become one of the central ideas of network computing. Many business applications being written today use the client–server model

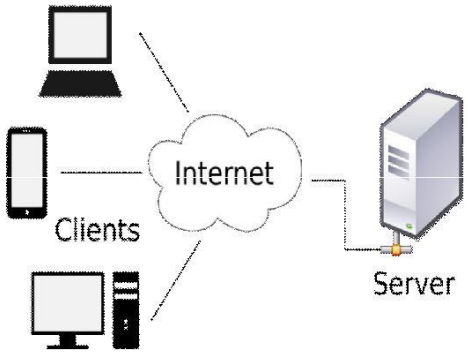


Fig 3.1Model of the Client Server

File Server

A file server is a computer which is responsible for the central storage and management of data files such that other computers in the same network can access the files. A file server will allow the users to share data over a network. Any computer can be designed to be a host and can act as a file server. In simple, a file server can be an ordinary PC that handles requests for files from the clients and sends them over the network (response).

Simply a File Server is a program or mechanism that enables the required processes for sharing the files can also be called as a file server. On the Internet, such programs mainly use the File Transfer Protocol (FTP).

The term server in File Server holds the role of the machine in the client–server model, where the clients are the workstations who are using the storage. A file server won't perform any computational tasks, and it won't run any programs on behalf of its clients. It is primarily designed to enable the storage and accessing of data while the computations are performed by the workstations. We can see file servers in schools and offices, where users uses a LAN cable to connect their computers.

Data Structures:

File Transfer Protocol allows 3 types of data structures :

File Structure – In this the file is considered to be a continuous sequence of data bytes and it does not have any internal structure

Record Structure – In this the file is made up of records which are arranged in sequential manner.

Page Structure – In this the file is made up of independent indexed pages.

File Transfer Protocol commands:

PWD – This command gives the name of the current working directory as the reply.

MKD – This command is used to create a new directory in the specified path.

RMD – This command is used to remove an existing directory from a specified path

ABOR – This command is used to abort the previous FTP service command. QUIT – This command is used to terminate a USER only if file transfer is not in progress, then the server will close the control connection.

LIST – This command will send a request to display the list of all the files present in that particular directory.

A file server can be dedicated or non-dedicated. A dedicated server is specifically designed for using as a file server, with workstations attached for reading and writing files into databases.

Ethernet cards, HUBS, Switches, Routers, Gateways

Ethernet Cards:

An Ethernet card is a kind of network adapter. These adapters support the Ethernet standard for high-speed network connections by cables. Ethernet cards are also known as network interface cards (NICs). Ethernet cards are available in no of different standard packages known as form factors.

In olden days, large ISA cards were the first standard Ethernet cards for PCs, users have to open their computer case for installation

Now-a-days Ethernet cards are installed inside desktop computers which use the PCI standard and are usually installed by the manufacturer.

Smaller PCMCIA Ethernet cards which resembles credit cards are readily available for laptops and for other mobile computers. These can be inserted conveniently into the slots on the side or front of the device.

Though they look like small boxes rather like cards, external USB Ethernet adapters also exist. These are more convenient alternative to PCI cards for desktops and also for commonly used in video game consoles and other consumer devices which are not having PCMCIA slots.

Ethernet cards can operate at different network speeds which depends on the protocol standards they support. Old Ethernet cards were only capable of 10 Mbps maximum speed offered by Ethernet. All modern Ethernet adapters supports 100 Mbps. Now Fast Ethernet standards offer Gigabit Ethernet support at 1 Gbps (1000 Mbps).

An Ethernet card won't support Wi-Fi wireless networking directly, but home network broadband routers contain the necessary technology which allows Ethernet devices to connect by cables and communicate with Wi-Fi devices via router.

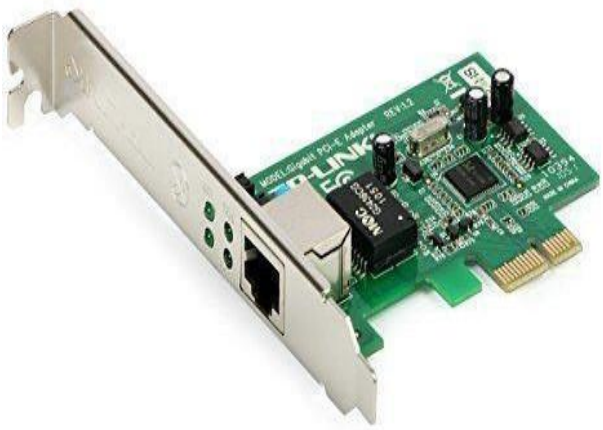


Fig 3.2 Ethernet Card

Hubs

A hub, also known as a network hub, is a common connection point for all the devices in a network. Hubs are devices which are commonly used to connect segments of LAN. The hub consist of multiple ports. When packetarrives at one port, simultaneously it is copied to other ports so that all segments of the LAN can see all the packets.

A passive hub simply serves as a channel for the data, enabling it to go from one device (or segment) to another.

As like the passive hub, intelligent hubs include some additional features that enables an administrator to monitor the traffic passing through the hub and to configure each port in the hub. Intelligent hubs are also known as manageable hubs. A third type of hub, known as switching hub, which reads the destination address of each packet and then it will forward the packet to the correct port.

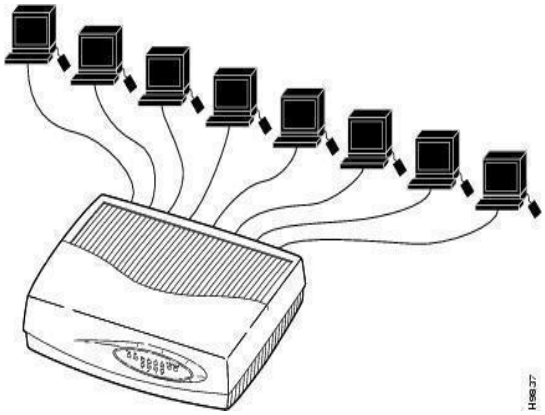


Fig 3.3 HUB

Types of Network Hubs

We have three types of network hubs:

They are

Passive hub

Active hub

Intelligent hub

Passive hubs: Simply these types of hubs are only used to establish the connection between any two devices. They simply receive the signal and then forward it to other devices, without amplifying or regenerating.

Active hubs: These hubs have an advantage as they amplify the incoming signal before sending to the multiple devices. If these hubs receive a weak signal for rebroadcasting they apply resynchronization and retiming techniques. These are also known as Multiport Repeaters

Intelligent hubs: These hubs can act as both passive and active hubs and it also includes remote management capabilities. They can also perform some other tasks like Bridging and routing.

Switches:

A network switch, which is a small hardware device which joins multiple computers together within one LAN. Technically, network switches operate at second layer (Data Link Layer) of the OSI model.

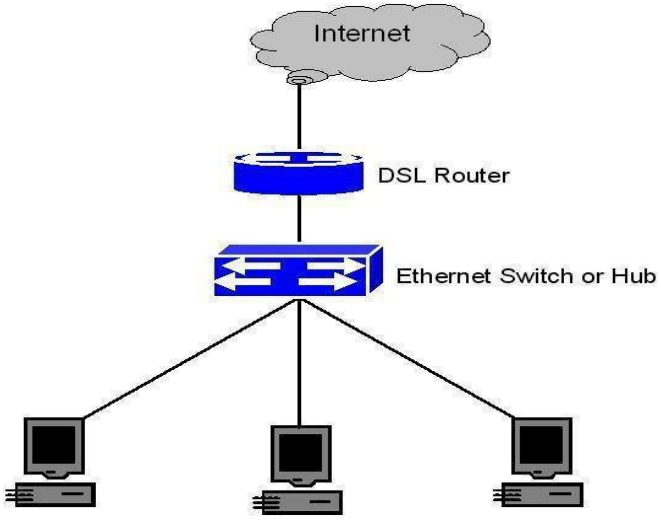


Fig 3.4 Switches

Network switches and network hubs are nearly look alike, but a switch contains more intelligence (and a slightly higher price tag) than a hub. Unlike hubs, network switches are capable of examining data packets as they are received, determining the source and destination device of each and every packet,

and forwarding them to destination appropriately. By delivering messages only to the device which are connected, a network switch conserves network bandwidth and offers better performance than a hub.

Routers

A **Router** is hardware device designed to receive, analyze and transfer incoming packets from one network to another network. It can also be used to convert the packets to another network interface, drop them, and perform other actions relating to a network.

A **router** has many capabilities than other network devices have, such as a hub or a switch which are able to perform basic network functions.

For example, a hub is often used to transfer information between computers or network devices, but it cannot analyze or it cannot do anything with the information (data) it is transferring. Routers can analyze the data being sent over a network, can change how it is packed, and send it to another network or over a different network.

For example, routers are most commonly used in home networks to share the single Internet connection between multiple computers.

Home networks naturally use a wireless or wired Internet Protocol (IP) router

Fig 3.5 Router

3.2.5Gateways:

A gateway is a network element which acts as an entrance point to another network.

For example consider an access gateway which acts as a gateway between telephony network and other network such as internet. LANs consist of component called gateways, which supports in transferring from one LAN to another LAN.

Typically a gateway is a work station or server. It acts as a two-way path between networks. It is used to connect networks of different types. Gateway is a work station by which we can establish connection between external network and internal network. Gateway belongs to transport layer and application layer of the OSI reference model.

Gateways also connect two networks even if the protocols are different. So protocol conversion is also done by gateways only. It simply uncover the old protocol and assigns new protocol to the packets, so it is also known as protocol translator. Because of the time consumption for protocol conversion, it is little bit slower.

Gateways are the junctions between two network architectures. They repackage and convert data transferring from one environment to another environment so that one environment can understand the data of other environment.

A gateway links two systems which consist of different communication protocols, different architecture and different language. Because of this gateways interconnects the heterogeneous networks

For example: Microsoft windows NT server to SNA (IBM's system network architecture).

Gateways passes bad packets too because they won't filter out the data. Gateways are task specific, because of this they are dedicated to a particular type of transfer only. They often eferred to by their particular task name only e.g. Windows NT server to SNA gateway.

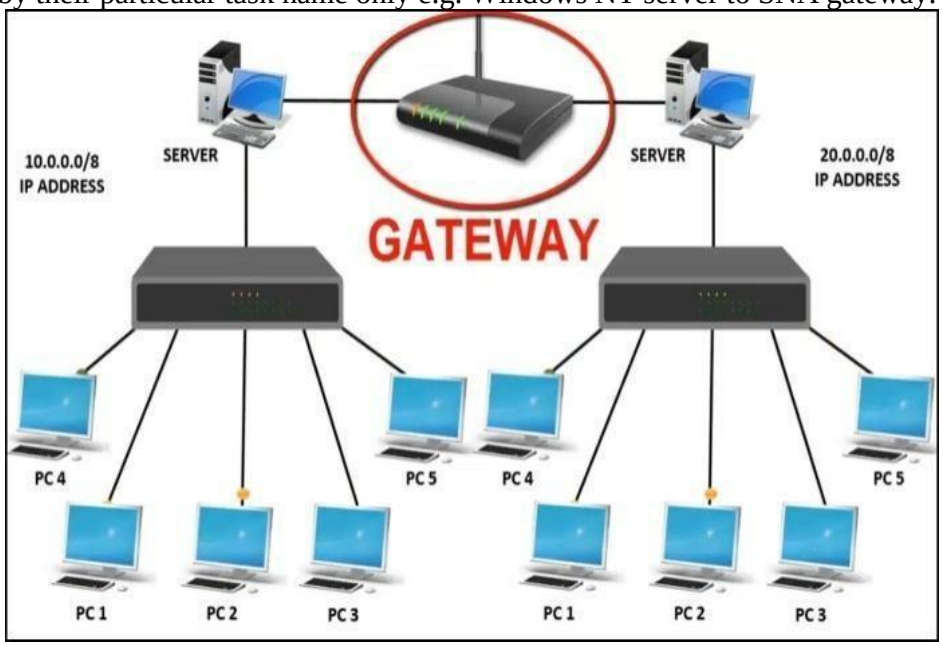


Fig 3.6 Gateways

Modem and Types(V-SAT and ATMS)

The first modem is known as the "Data phone" and it was released by AT&T in year 1960.

The full form of Modem is Modulator – Demodulator. Modems are mainly used for transferring of data from one computer network to another computer network by telephone lines. Now-a-days the computer network works in digital mode, whereas analog technology is used for carrying massages across the phone lines.

Modulator converts information (data) from **digital mode to analog mode** at the time of transmission from the transmitter end whereas the **demodulator** converts the same from **analog to digital at receiver end**. The process of conversion of analog signals of one computer network into digital signals of another computer network so that they can be processed by a receiver end computer is **referred to as digitizing.**

When an analog transmission is used for data communication between two digital devices known as Data Terminal Equipment (DTE), modems are used at each end. DTE can be a computer or a terminal.

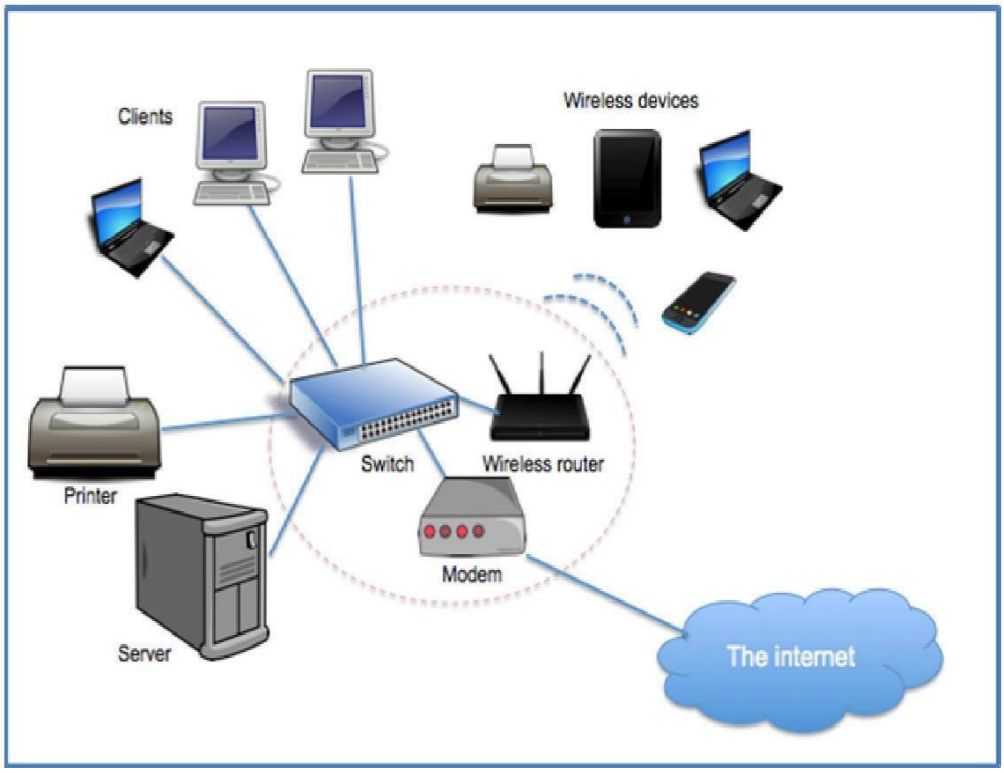


Fig 3.7 Modem

Modems can be categorized in several ways like the direction in which they can transmit data, type of connection to the transmission line, transmission mode, etc.

Depending on the direction of transmission of data, modem can be classified into following types −

Half duplex − In half-duplex modem the transfer of the data can be in both the directions but only one at a time.

Full duplex – A **full duplex modem** allows the transmission of data in both directions simultaneous.

Hence, there will be two carriers on the line, one for outgoing and the other for incoming.

Modems maybe either ethernet modems (Connection Oriented) or wireless modems (Connectionless)

Wireless modems are connected using a wireless LAN (WLAN). Ethernet modems are connected to a computer using a Network Card to which the modem is plug into. There are some modems which offer dual Ethernet and USB connection.

3.3.1.1 Types of Modems

The following are the four versions of computer modems found in computers

1. Onboard modem

In this Onboard modems, modems are integrated or attached to the computer motherboard. They cannot be removed but they can be disabled through a jumper or CMOS Setup.

2. Internal modem

In this Internal modems, modem will be connected to a PCI slot which is present inside of a newer desktop computer or it may be connected to an ISA slot on an older computer.

3. External modem

In this type, modem are present within a box which connects to the computer externally by the serial ports or by the USB port.

4.Removable modem

Modems which are used with older laptops PCMCIA slot and can be removed or added as needed.

3.3.2 V-SAT MODEM:

A satellite modem (also called as Sat Modem) is used to enable the transfer of data using communications satellite as a relay.

The main function of the satellite modem is to transform an input bitstream to a radio signal and vice versa. Modulation and Demodulation are the main functions of satellite modems.

Satellite communication standards also define error correction codes and framing formats.

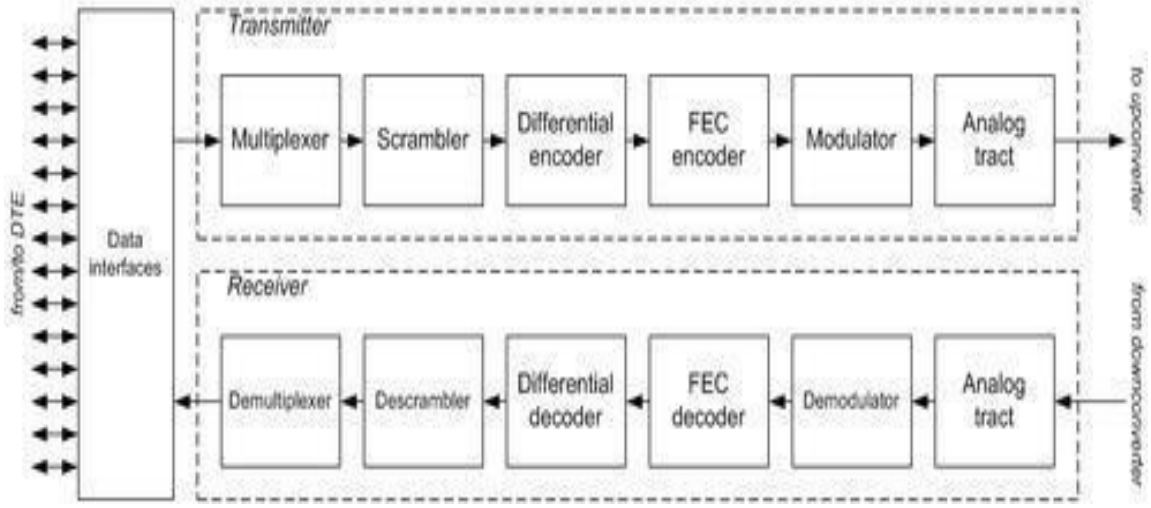


Fig 3.8 Internal structure of satellite modem.

A very small aperture terminal (V-SAT) is a two way satellite ground station. V-SAT are most commonly used to transmit narrowband data or broad band data. V-SAT networks are mostly configured in STAR topology or MESH topology or combination of both STAR and MESH where

the configuration is different for each. V-SAT access satellites in geosynchronous orbit in order to relay data from terminals such as small remote earth stations to other terminals or to master earth station i.e, hubs (varies respectively based on topology)

The utilization of this configuration is done to minimize the overall cost of the network and to reduce the amount of data that has to be relayed.

3.3.4 Asynchronous Transfer Mode (ATM):

ATM is telecommunication concept where it is used to handle entire range of user traffic such as data, video or voice which is conveyed in small fixed size packets called cells which are transmitted asynchronously in a connection oriented network.

Making an ATM call initially needs to send a message to set up a connection. Where all cells follow the same path to the destination. Thus it can carry multiple types of traffic(constant rate and variable rate traffic) with **end-to-end** quality of service. ATM is independent of transmission medium. ATM networks use “Packet” or “cell” Switching with virtual circuits. It’s design helps in the implementation of high performance multimedia networking.

Benefits of ATM:

ATM Cell Format:

It transfers information in fixed-size units knownas cells. Each cell consists of 53 bytes as shown in the below Fig. In first 5 bytes it containsthe information of cell-header, and the remaining 48 bytes consist of the payload (user information). Small and fixed-length cells are well suited to transfer voice and video traffic.

Header	Payload
5 bytes	48bytes

Fig 3.9 ATM Payload Format

Asynchronous Transfer Mode can be of two format types which are as follows:

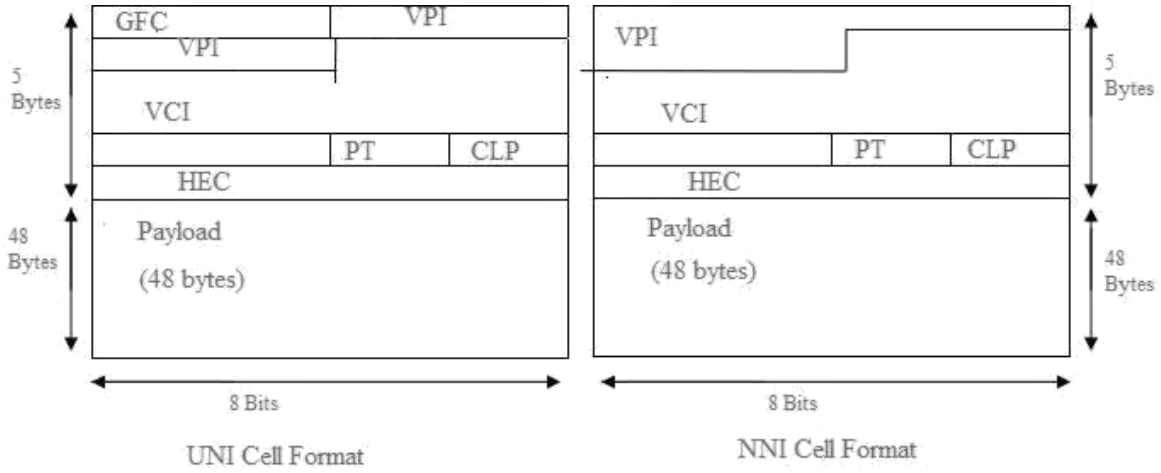


Fig 3.10 ATM Format

UNI Header: This is used within private networks of ATM for communication between ATM endpoints and ATM switches. It includes the Generic Flow Control (GFC) field.

NNI Header: This is used for communication between ATM switches, and it does not include the Generic Flow Control(GFC) instead it includes a Virtual Path Identifier (VPI) which occupies the first 12 bits.

Working of ATM:

ATM standard uses two types of connections. i.e., Virtual path connections (VPCs) which consists of Virtual channel connections (VCCs) bundled together which is a basic unit carrying single stream of cells from user to user. A virtual path can be created end-to-end across an ATM network, as it does not routs the cells to a particular virtual circuit. In case of major failure all cells belonging to a particular virtual path are routed the same way through ATM network, thus helping in faster recovery.

Switches connected to subscribers uses both VPIs and VCIs to switch the cells which are Virtual Path and Virtual Connection switches that can have different virtual channel connections between them, serving the purpose of creating a virtual trunk between the switches which can be handled as a single entity. It's basic operation is straightforward by looking up the connection value in the local translation table determining the outgoing port of the connection and the new VPI/VCI value of connection on that link.

ATM Layers:

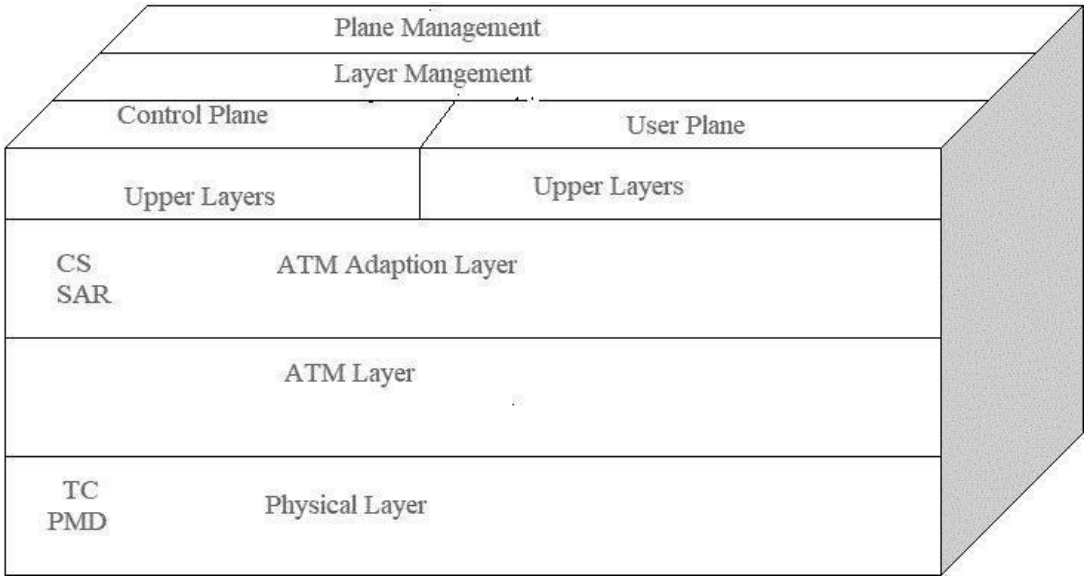


Fig 3.11 ATM Layers

1. **ATM Adaption Layer (AAL)**

Isolating higher layer protocols from details of ATM processes is done here. It prepares for conversion of user data into cells and segments it into 48-byte cell payloads. AAL protocol excepts transmission from upper layer services and assists them in mapping applications, e.g., voice, data to ATM cells.

2. **Physical Layer**

It manages the medium-dependent transmission. physical medium-dependent sub layer and transmission convergence sub layer are the two parts. Main functions are as follows:

•It converts cells into a bit stream.

•It can track the ATM cell boundaries.

•It controls the transmission and receipt of bits in the physical medium.

•Looks for the packaging of cells into appropriate type of frames.

3. **ATM Layer**

Transmission, switching, congestion control, cell header processing, sequential delivery, etc., is done in this layer. It is responsible for simultaneously sharing the virtual circuits over the physical link known as cell multiplexing **ATM**

Summary:

LAN ‘s are made up of software as well as hardware components

A **server** is a software program, or it may be a computer, in which that program runs, that provides a specific kind of service (Response) to client software running in the same computer or other computers in a network.

Web server consists of files related to web sites and servers and sends(If requested by the user) them across the Internet to the clients (i.e., by web browsers)

File server is a software, or the combination of hardware and software, that is dedicated for storing files and it will give access to the clients for reading or for writing across the network.

Print server is the software or the hardware that manages one or more printers.

Network server manages network traffic.

Name server maps user and computer names to machine addresses.

Database server allows clients to interact with a database in a network.

Application server runs different applications for clients

Client may be an application or system that accesses a service that is made available by a server. The server is often (but not always) on different computer system, in such case the clients access the services by the network.

The **client-server model** is a distributed communication framework of network; it is the process between service requestors, clients and service providers, servers. This connection is established through a network or the Internet.

A **client consists** of Web browsers, chat applications, and email software. Whereas Servers have Web, database, application, chat and email, etc.

A **file server** is a computer which is responsible for the central storage and management of data files such that other computers in the same network can access the files. A file server will allow the users to share data over a network.

File Structure – In this the file is considered to be a continuous sequence of data bytes and it does not have any internal structure

Record Structure – In this the file is made up of records which are arranged in sequential manner.

Page Structure – In this the file is made up of independent indexed pages.

An **Ethernet card** is a kind of network adapter. These adapters support the Ethernet standard for high-speed network connections by cables. Ethernet cards are also known as network interface cards (NICs).

A **hub**, also known as a network hub, is a common connection point for all the devices in a network. Hubs are devices which are commonly used to connect segments of LAN. We have three types of network hubs. They are Passive hub, Active hub, Intelligent hub.

Passive hubs: Simply these types of hubs are only used to establish the connection between any two devices. They simply receive the signal and then forward it to other devices, without amplifying or regenerating.

Active hubs: These hubs have an advantage as they amplify the incoming signal before sending to the multiple devices. If these hubs receive a weak signal for rebroadcasting they apply resynchronization and retiming techniques. These are also known as Multiport Repeaters

Intelligent hubs: These hubs can act as both passive and active hubs and it also includes remote management capabilities. They can also perform some other tasks like Bridging and routing.

A **Network switch**, which is a small hardware device which joins multiple computers together within one LAN. Technically, network switches operate at second layer (Data Link Layer) of the OSI model.

Network switches and network hubs are nearly look alike, but a switch contains more intelligence (and a slightly higher price tag) than a hub. Unlike hubs, network switches are capable of examining data packets as they are received, determining the source and destination device of each and every packet, and forwarding them to destination appropriately.

A **router** is hardware device designed to receive, analyze and transfer incoming packets from one network to another network. It can also be used to convert the packets to another network interface, drop them, and perform other actions relating to a network. routers are most commonly used in home networks to share the single Internet connection between multiple computers.

Home networks naturally use a wireless or wired Internet Protocol (IP) router

A **gateway** is a network element which acts as an entrance point to another network. Typically a gateway is a work station or server. It acts as a two-way path between networks. It is used to connect networks of different types. Gateway is a work station by which we can establish connection between external network and internal network. Gateway belongs to transport layer and application layer of the OSI reference model.

Gateways also connect two networks even if the protocols are different. So
protocol conversion is also done by gateways only

The full form of Modem is Modulator – Demodulator. Modems are mainly used for transferring of data from one computer network to another computer network by telephone lines. Now-a-days the computer network works in digital mode, whereas analog technology is used for carrying massages across the phone lines.

Modulator converts information (data) from **digital mode to analog mode** at the time of transmission from the transmitter end whereas the **demodulator** converts the same from **analog to digital at receiver end**.

Half Duplex − In half-duplex modem the transfer of the data can be in both the directions but only one at a time.

Full duplex – A **full duplex modem** allows the transmission of data in both directions simultaneous. Hence, there will be two carriers on the line, one for outgoing and the other for incoming.

Modems maybe either Ethernet modems (Connection Oriented) or wireless modems (Connectionless)

Wireless Modems are connected using a wireless LAN (WLAN). Ethernet modems are connected to a computer using a Network Card to which the modem is plug into. There are some modems which offer dual Ethernet and USB connection.

Following are the four versions of computer modems found in computers

Onboard Modem – In this Onboard modems, modems are integrated or attached to the computer motherboard. They cannot be removed but they can be disabled through a jumper or CMOS Setup.

Internal Modem – In this Internal modems, modem will be connected to a PCI slot which is present inside of a newer desktop computer or it may be connected to an ISA slot on an older computer.

External Modem –In this type, modem are present within a box which connects to the computer externally by the serial ports or by the USB port.

Removable Modem – Modems which are used with older laptops PCMCIA slot and can be removed or added as needed.

A **Satellite Modem** (also called as sat modem) is used to enable the transfer of data using communications satellite as a relay.

The **main function of the satellite modem** is to transform an input bitstream to a radio signal and vice versa. Modulation and Demodulation are the main functions of satellite modems

ATM is telecommunication concept where it is used to handle entire range of user traffic such as data, video or voice which is conveyed in small fixed size packets called cells which are transmitted asynchronously in a connection oriented network.

Short Answer TypeQuestions

1. What are LAN Components ?
2. Write various types of LAN Components.
3. Define Server and Client
4. What is a file server ?
5. What are Ethernet Cards ?
6. What are Hubs and Switches ?
7. What is Router ?
8. What are Gateways ?
9. What is a Modem and list types of Modems ?
10. Write briefly about ATM.
11. Expand V-SAT, ATM, FTP, EDI.

Long Answer Type Questions

1. Explain briefly about any three LAN Components.
2. Explain the function of Modem with a neat diagram.
3. Discuss briefly about: Hubs and Switches.
4. Explain in detail about V-SAT
5. Explain in detail about ATM with its layer architecture.
6. Discuss about (a)Routers (b)Gateways.
7. Explain in detail about hubs and its types

UNIT 4

Computer Networks

Learning Objectives

After studying this unit, the student will be able to
Know an Internet and its usage
Study of various browsers
Know different types of viruses
Know about Anti-virus software.

Computer Network

A Network is a set of devices (often referred to as nodes) connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network. A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.

Performance

Performance can be measured in many ways, including transit time and response time. Transit time is the amount of time required for a message to travel from one device to another. Response time is the elapsed time between an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

Reliability

Network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

Security

Network security issues include protecting data from unauthorized access, protecting data from damage and development, and implementing policies and procedures for recovery from breaches and data losses.

Advantages of Network

Speed

Sharing and transferring files within Networks are very rapid. Thus saving time, while maintaining the integrity of the file.

Cost

Individually licensed copies of many popular software programs can be costly. Networkable versions are available at considerable savings. Shared programs, on a network allows for easier upgrading of the program on one single file server, instead of upgrading individual workstations.

Security

Sensitive files and programs on a network are passwords protected or designated as "copy inhibit," so that you do not have to worry about illegal copying of programs.

Centralized Software Management

Software can be loaded on one computer (the file server) eliminating that need to spend time and energy installing updates and tracking files on independent computers throughout the building.

Resource Sharing

Resources such as, printers, fax machines and modems can be shared.

Electronic Mail

E-mail aids in personal and professional communication.

Flexible Access

Access their files from computers throughout the firm.

Workgroup Computing

Workgroup software (such as Microsoft BackOffice) allows many users to work on a document or project concurrently.

Categories (Types) and connections of networks Types of Network

Networks are discussed in terms of their (i) transmission technology/type of connection and

their (ii) scale / size

Types of Network Based on the Transmission Technology/Type of Connection

1 .Broadcast Networks (Multi point)

2 .Point-to-point Networks

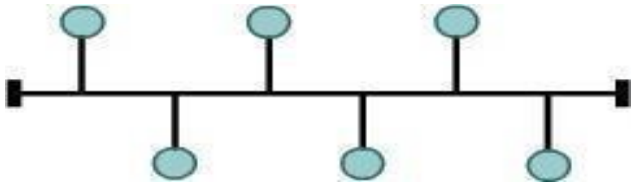
1 .Broadcast Networks (Multi point)

Single communication channel shared by all computers

Packets (short messages) sent by a computer contains an address specifying the destination computer

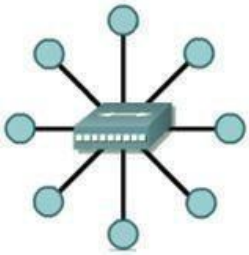
All computers connected to the network receive the packet. The destination computer processes the packet while all other computers discard the packet

In multipoint (multi drop) connection the capacity of channel is shared either temporary or spatially



Point-to-Point Networks

- Consists of many connectionsbetween individual pairsof computers



Sending a packet between two computer may involve the packet being forwarded by intermediate computers

Multiple routes of different lengths possible, therefore routing algorithms are important

CONNECTION-ORIENTED VERSUS CONNECTIONLESS SERVICE

Layers can offer two different types of service to the layers above them: connection-oriented and connectionless.

Connection-oriented service (modeled after the telephone system): to use it, the service user first establishes a connection, uses the connection, and then releases the connection. The essential aspect of a connection is that it acts like a tube: the sender pushes objects (bits) in at one end, and the receiver takes them out in the same order at the other end.

Connectionless service (modeled after the postal system): Each message carries the full destination address, and each one is routed through the system independent of all the others. Such connectionless services are often called datagram services.

Quality of service - some services are reliable in the sense that they never lose data. Reliability is usually implemented by having the receiver acknowledge the receipt of each message. The acknowledgment process is often worth but introduces sometimes undesirable overheads and delays.

Reliable connection-oriented service has two minor variation:

Message sequences - the message boundaries arepreserved.

Byte streams - the connection is simply a stream of bytes, with no messageboundaries.

Applications where delays introduced by acknowledgment are unacceptable:

Digitized voicetraffic,

Video filmtransmission.

The use of connectionless services:

Electronic junk mail (third class mail as advertisements) - this service is moreover unreliable (meaning notacknowledged).

Acknowledged datagram services - connectionless datagram services withacknowledgment.

Request-reply service - the sender transmits a single datagram containing a request. The reply contains the answer. Request-reply is commonly used to implement communication in the client-servermodel.

Overview of Network Models

Protocol Layering Hierarchy

To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network tonetwork. The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actuallyimplemented. Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer nprotocol.

Layering provides two nice features.

It decomposes the problem of building a network into more manageable components. Rather than implementing a monolithic piece of software that does everything implement several layers, each of which solves one part of theproblem

It provides more modular design. To add some new service, it is enough to modify the functionality at one layer, reusing the functions provided at all the otherlayers.

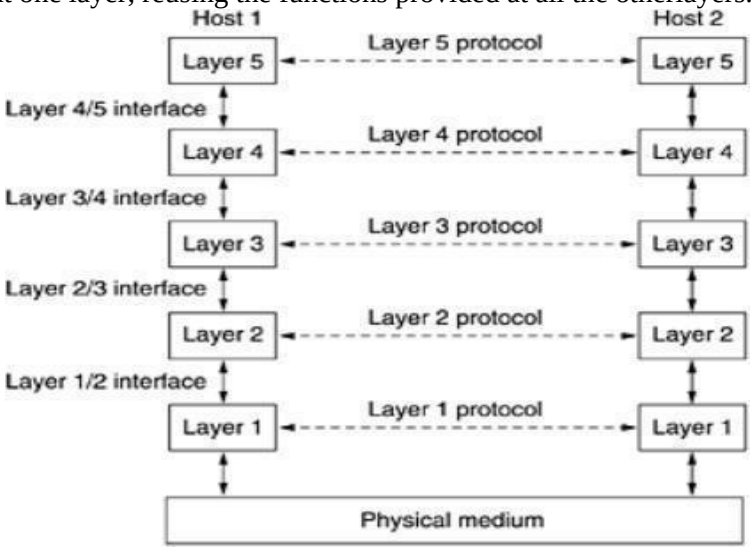


Fig 4.1 Structure of the Layer Hierarchy

A five-layer network is illustrated in figure below. The entities comprising the corresponding layers on different machines are called peers. The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.

Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one.

No data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. A set of layers and protocols is called **network architecture**. A list of protocols used by a certain system, one protocol per layer, is called a **protocolstack**.

Protocols

A protocol is an agreement between the communicating parties on how communication is to proceed. A protocol is a set of rules that governs data communication. It defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics and timing.

Each protocol defines two different interfaces.

Service interface - to the other objects on the same computer that want to use its communication services. This service interface defines the operations that local objects can perform on theprotocol.

Peer interface - to its counterpart (peer) on another machine. It also defines the form and meaning of messages exchanged between protocol peers to implement the communicationservice.

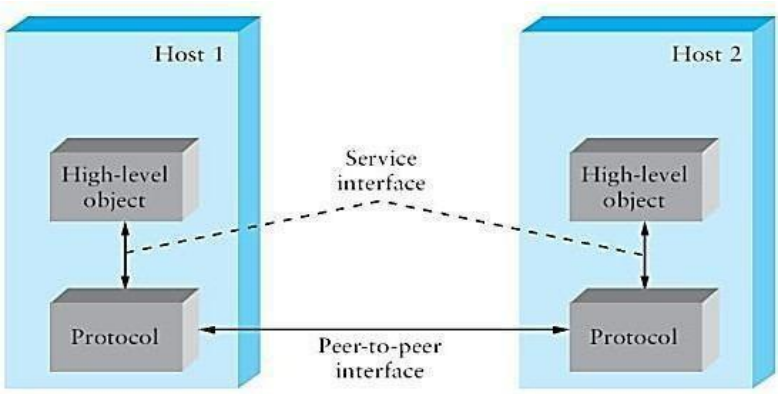


Fig 4.2 Abstract view of Protocol based communication

Encapsulation

Control information must be added with the data to instruct the peer how to handle with the received message. It will be added into the header ortrailer.

Header - Small data structure from few bytes to few kilobytes attached to the front of message.
Trailer – Information will be added at the end of themessage
Payload or message body – Data send by the program In this case data is encapsulated with new message created by protocol at eachlevel.

The Relationship of Services to Protocols

A service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the serviceuser.

A protocol, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled.

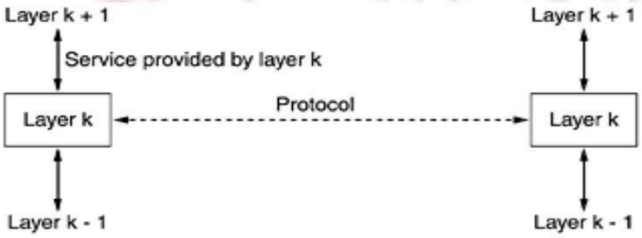


Fig 4.3 Detailed layer functionality

4.1.1 OSIModel

OSI Model

ISO defines a common way to connect computer by the architecture called Open System Interconnection (OSI) architecture. The OSI model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO-OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. Network functionality is divided into seven layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

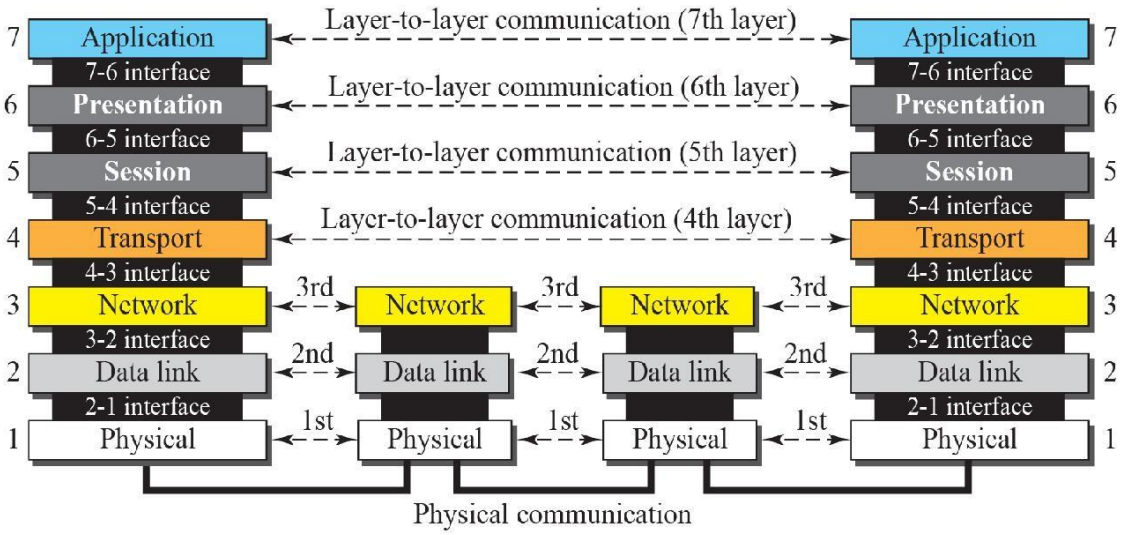
A layer should be created where a different abstraction isneeded.

Each layer should perform a well-definedfunction.

The function of each layer should be chosen with an eye toward defining internationally standardizedprotocols.

The layer boundaries should be chosen to minimize the information flow across the interfaces.

The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.



4.4 OSI Reference Model

Organization of the layers

The 7 layers can be grouped into 3 subgroups

Network Support Layers

Layers 1,2,3 - Physical, Data link and Network are the network support layers. They deal with the physical aspects of moving data from one device to another such as electrical specifications, physical addressing, transport timing andreliability.

TransportLayer

Layer4, transport layer, ensures end-to-end reliable data transmission on a single link.

User SupportLayers

Layers 5,6,7 – Session, presentation and application are the user support layers. They allow interoperability among unrelated softwaresystems

Functionalities of the Layers

PHYSICAL LAYER

The physical layer coordinates the functions required to transmit a bit stream over a physicalmedium.

The physical layer is concerned with the following:

Physical characteristics of interfaces and media - The physical layer defines the characteristics of the interface between the devices and the transmissionmedium.
Representation of bits - To transmit the stream of bits, it must be encoded to signals. The physical layer defines the type ofencoding.

Data Rate or Transmission rate - The number of bits sent each second – is also defined by the physicallayer.
Synchronization of bits - The sender and receiver must be synchronized at the bit level.Their clocks must besynchronized.
Line Configuration - In a point-to-point configuration, two devices are connected together through a dedicated link. In a multipoint configuration, a link is shared between severaldevices.
Physical Topology - The physical topology defines how devices are connected to make a network. Devices can be connected using a mesh, bus, star or ringtopology.
Transmission Mode - The physical layer also defines the direction of transmission between two devices: simplex, half-duplex or full-duplex.

DATA LINKLAYER

It is responsible for transmitting frames from one node to next node. The other responsibilities of this layerare

Framing - Divides the stream of bits received into data units calledframes.

Physical addressing – If frames are to be distributed to different systems on the n/w , data link layer adds a header to the frame to define the sender andreceiver.

Flow control- If the rate at which the data are absorbed by the receiver is less than the rate produced in the sender ,the Data link layer imposes a flow ctrlmechanism.
Error control- Used for detecting and retransmitting damaged or lost frames and to prevent duplication of frames. This is achieved through a trailer added at the end of the frame.
Access control -Used to determine which device has control over the link at any given time.
It is responsible for **Hop to Hop**delivery.

NETWORKLAYER

This layer is responsible for the delivery of packets from source to destination. It is mainly required, when it is necessary to send information from one network to another.

The other responsibilities of this layer are

Logical addressing - If a packet passes the n/w boundary, we need another addressing system for source and destination called logicaladdress.
Routing – The devices which connects various networks called routers are responsible for delivering packets to finaldestination.
It is responsible for **Host** to **Host**delivery.

TRANSPORT LAYER

- It is responsible for **Process to Process**delivery.
- It also ensures whether the message arrives in order or not. The other responsibilities of this layerare

Port addressing - The header in this must therefore include a address called port address. This layer gets the entire message to the correct process on thatcomputer.
Segmentation and reassembly - The message is divided into segments and each segment is assigned a sequence number. These numbers are arranged correctly on the arrival side by thislayer.
Connection control - This can either be **connectionless or connection-oriented.** The connectionless treats each segment as a individual packet and delivers to the destination. The connection-oriented makes connection on the destination side before the delivery. After the delivery the termination will beterminated.
Flow and error control - Similar to data link layer, but process to process takeplace.

SESSIONLAYER

This layer establishes, manages and terminates connections between applications. The other responsibilities of this layerare

Dialog control - This session allows two systems to enter into a dialog either in half duplex or full duplex.
Synchronization-This allows to add checkpoints into a stream ofdata.

PRESENTATIONLAYER

It is concerned with the syntax and semantics of information exchanged between two systems.

The other responsibilities of this layer are

Translation – Different computers use different encoding system, this layer is responsible for interoperability between these different encoding methods. It will change the message into some commonformat.

Encryption and decryption-It means that sender transforms the original information to another form and sends the resulting message over the n/w. and viceversa.

Compression and expansion-Compression reduces the number of bits contained in the information particularly in text, audio andvideo.

APPLICATIONLAYER

This layer enables the user to access the n/w. This allows the user to log on to remote user. The other responsibilities of this layerare

FTAM (file transfer, access, mgmt) - Allows user to access files in a remotehost.

Mail services - Provides email forwarding andstorage.

Directory services - Provides database sources to access information about various sources andobjects.

Network virtual terminal (Remotelog-in)

Accessing the World WideWeb

TCP/IP Reference Model**:** TCP/IP protocols map to a four-layer conceptual model known as the DARPA model , named after the U.S. government agency that initially developed TCP/IP. The four layers of the DARPA model are: Application, Transport, Internet, and Network Interface. Each layer in the DARPA model corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) model.

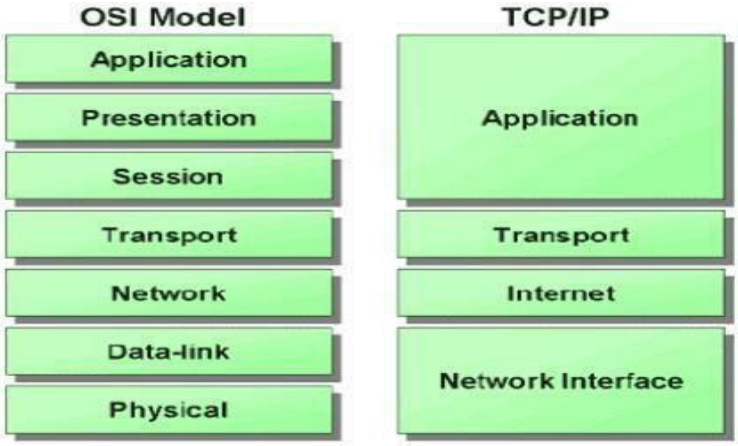


Fig 4.5 Abstract View of TCP/IP Reference Model

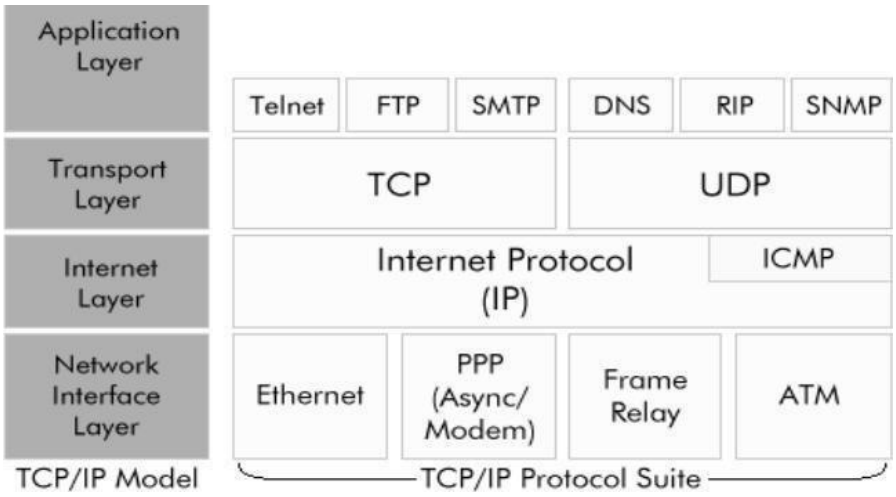


Fig 4.6 Logical view of TCP/IP Reference Model

The Host to NetworkLayer:

Below the internet layer is great void. The TCP/IP reference model does not really say such about what happen here, except to point out that the host has connect to the network using some protocol so it can transmit IP packets over it. This protocol is not specified and varies from host to host and network tonetwork.

Internetlayer:

It is a connectionless internetwork layer forming a base for a packet-switching network. Its job is to allow hosts to insert packets into any network and have them to deliver independently to the destination. They may appear in a different order than they were sent in each case it is job of higher layers to rearrange them in order to deliver them to proper destination.TCP/IP internet layer isvery similar in functionality to the OSI networklayer.

Packet routing is very essential task in order to avoid congestion. For these reason it is say that TCP/IP internet layer perform same function as that of OSI network layer.

The internet layer defines an official packet format and protocol called IP (Internet Protocol) and its provides

Best-effortdelivery

No errorchecking

Notracking

IP is a host-to-hostprotoco**l.**

▪

Transportlayer:

In the TCP/IP model, the layer above the internet layer is known as transport layer. It is developed to permit entities on the source and destination hosts to carry on a conversation. It specifies 2 end-to-endprotocols

TCP (Transmission Control Protocol)

UDP (User DatagramProtocol)

TCP

It is a reliable connection-oriented protocol that permits a byte stream originating on one machine to be transported without error on any machine in the internet. It divides the incoming byte stream into discrete message and passes each one onto the internet layer. At the destination, the receiving TCP process collects the received message into the output stream. TCP deals with flow control to make sure a fast sender cannot swamp a slow receiver with more message than it canhandle.

UDP

It is an unreliable, connectionless protocol for applications that do not want TCP's sequencing on flow control and wish to offer their own. It is also used for client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery such as transmitting speech or video.

ApplicationLayer:

In TCP/IP model, session or presentation layer are not present. Application layer is present on the top of the Transport layer. It includes all the higher-level protocols which are virtual terminal (TELNET), file transfer (FTP) and electronic mail (SMTP). The virtual terminal protocol permits a user on one machine to log into a distant machine and work there. The file transfer protocol offers a way to move data efficiently from one machine to another. Electronic mail was used for file transfer purpose but later a specialized protocol was developed forit.

The Application Layer defines following protocols

File Transfer Protocol(FTP)

It was designed to permit reliable transfer of files over different platforms. At the transport layer to ensure reliability, FTP uses TCP. FTP offers simple commands and makes the differences in storage methods across networks transparent to the user. The FTP client is able to interact with any FTP server; therefore the FTP server must also be able to interact with any FTP client. FTP

does not offer a user interface, but it does offer an application program interface for file transfer. The client part of the protocol is called as FTP and the server part of the protocol is known as FTPd. The suffix "d" means Daemon this is a legacy from Unix computing where a daemon is a piece of software running on a server that offers aservice.

Hyper Text TransferProtocol

HTTP permits applications such as browsers to upload and download web pages. It makes use of TCP at the transport layer again to check reliability. HTTP is a connectionless protocol that sends arequest,receivesaresponseandthendisconnectstheconnection.HTTPdeliversHTML documents plus all of the other components supported within HTML such as JavaScript, Visual script and applets.

Simple Mail TransferProtocol

By using TCP, SMTP sends email to other computers that support the TCP/IP protocol suite. SMTP provides extension to the local mail services that existed in the early years of LANs. It supervises the email sending from the local mail host to a remote mail host. It is not reliable for accepting mail from local users or distributing received mail to recipients this is the responsibility of the local mail system. SMTP makes use of TCP to establish a connection to the remote mail host, the mail is sent, any waiting mail is requested and then the connection is disconnected. It can also return a forwarding address if the intended recipient no longer receives email at that destination. To enable mail to be delivered across differing systems, a mail gateway isused.

Simple Network ManagementProtocol

For the transport of network management information, SNMP is used as standardized protocol. Managed network devices can be cross examined by a computer running to return details about their status and level of activity. Observing software can also trigger alarms if certain performance criteria drop below acceptable restrictions. At the transport layer SNMP protocol uses UDP. The use of UDP results in decreasing network traffic overheads.

Comparison of OSI and TCP/IP reference models

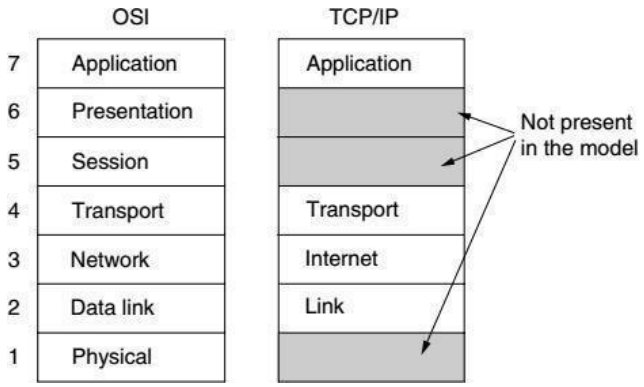


Fig 4.7 Comparison of OSI and TCP/IP Reference Models

An obvious difference between the two models is the number of layers: the OSI model has seven layers and the TCP/IP has four layers. Both have (inter)network, transport, and application layers, but the other layers are different.

Another difference is in the area of connectionless versus connection-oriented communication. The OSI model supports both connectionless andconnection-oriented communication in the network layer, but only connection-oriented communication in the transport layer, where it counts (because the transport service is visible to the users).

The TCP/IP model has only one mode in the network layer (connectionless) but supports both modes in the transport layer, giving the users a choice. This choice is especially important for simple request-response protocols.

The OSI and TCP/IP reference models have much in common. Both are based on the concept of a stack of independent protocols. Also, the functionality of the layers is roughly similar. For example, in both models the layers up through and including the transport layer are there to provide an end-to-end, network-independent transport service to processes wishing to communicate. These layers form the transport provider. Again in both models, the layers above transport are application-oriented users of the transportservice.

Three concepts are central to the OSI model:

Services.

Interfaces.

Protocols.

Probably the biggest contribution of the OSI model is to make the distinction between these three concepts explicit. Each layer performs some services for the layer above it. A layer's interface tells the processes above it how to access it. It specifies what the parameters are and what results to expect. It, too, says nothing about how the layer works inside. Finally, the peer protocols used in a layer are the layer's own business. It can use any protocols it wants to, as long as it gets the job done (i.e., provides the offered services). It can also change them at will without affecting software in higherlayers.

The TCP/IP model did not originally clearly distinguish between service, interface, and protocol. For example, the only real services offered by the internet layer are SEND IP PACKET and RECEIVE IP PACKET. As a consequence, the protocols in the OSI model are better hidden than in the TCP/IP model and can be replaced relatively easily as the technologychanges.

The OSI reference model was devised before the corresponding protocols were invented. This ordering means that the model was not biased toward one particular set of protocols, a fact that made it quitegeneral.

With TCP/IP the reverse was true: the protocols came first, and the model was really just a description of the existing protocols. There was no problem with the protocols fitting the model. They fit perfectly. The only trouble was that the model did not fit any other protocol stacks.

Summary

The internet in simple terms is a network of the interlinked computer networking worldwide, which is accessible to the general public.

There many advantages to using the internet such as Email, Information, Services, Buy or sell products, Online Chat, Downloading Software.

A browser is software that is used to access the internet.

The most common browser software titles on the market are: Microsoft Internet Explorer, Mozilla Firefox, Apple Computer's Safari, and Opera.

A message is a string of bytes that is meaningful to the applications that use it. Messages are used to transfer information from one application program to another.

E-mail is an electronic version of sending a letter. You can send e-mail from your computer at any time of the day to any address around the world and your electronic letter will arrive at its destination seconds after you send it... even if the receiver lives on the other side of the world.

A network protocol defines rules and conventions for communication between network devices.

File Transfer Protocol, or FTP, is a protocol used for transferring files from one computer to another - typically from your computer to a web server.

Internet security is a branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole.

A Virus is a piece of software that can infect other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

A worm is a program that can replicate itself and send copies from computer to computer across network connections.

A Trojan horse, or Trojan, is a malicious application that tricks as a authentic file or helpful program but whose real purpose is, for example, to grant a hacker unauthorized access to a computer.

Phishing is an e-mail fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients.

Antivirus software is a computer program that detects, prevents, and takes action to disarm or remove malicious software programs, such as viruses and worms.

Short Answer Type Questions

1. What is computer Network?
2. What is Protocol?
3. Expand the terms OSI, TCP /IP,FTAM, SMTP, FTP, HTTP, SNMP, ISDN.
4. What is Firewall?
5. List the layers of OSI Reference Model.
6. List the layers of TCP /IP Reference Model.

Long Answer Type Questions

1. Explain OSI Reference Model indetail.
2. Explain TCP /IP Reference Model in detail.

UNIT 5

Internet Connectivity and Services

Introduction to Internet:

In simple terms, Internet is the network of networks. It is the collection of all the computers across the world which can be accessed by public. These interconnected computers work by transmitting the data through a special type ofSwitching technique called packet switching.

Internet is a very huge network of many different interlinked networks related to the business, academic, government, and even smaller domestic networks. This network enables the internet to be used for several important functions such as the several means of communications like the file transfer, the online chatting and even the document share and web sites on the World Wide Web(WWW).

Frequently Used Terms In Internet

World Wide Web (WWW): The World Wide Web ("WWW") or simply the "web" is a collection of several web pages that are linked together like a spider web. These documents are stored on some computers known as servers located around the world.

Web Server: A Web Server is nothing but a computer which stores web pages. It is responsible for accepting request(s) from users and serves them with web pages (as Response).Following are some of the important web server programs: IIS (Internet Information server) and Apache, etc. 24*7 these Web servers are available to us.

Hyperlink: It is simply a link from one document to another document in any location. If we want to migrate from one document to another document then we can simply click on that highlighted word or image. Hyper links are used to play an audio or video file, to download a file, to set up a message to an e-mail address etc.

HTML (Hypertext Mark up Language): It is simply a language used to develop websites. It consists of certain key words called 'Tags' used for developing the documents on the web.

Web Page: A web page is an electronic web document written in a computer language called HTML (Hypertext Mark up Language).Web pages consists of text, graphics, animation , video and sound, as well as some interactive features, such as data entry forms.

URL (Uniform Resource Locator) : Each web page has a unique address known as a URL (Uniform Resource Locator) which identifies the location of web page in the server. Web pages usually consist of hyper links by which we can migrate to other web pages.

Website: A website is collection of one or more web pages, belonging to a specific company, government, institute or an individual. Every website consists of home page, which acts like an index.

Advantages of Internet

1. Buy or sell products

Throughout the world one of the easy and more effective ways to buy and sell products is Internet. Along with online shopping you can also get a huge amount of information about a particular product. We have many online sites and stores that can be used for getting information about the

product as well as we can buy them using our credit card. There is no need to step out of your house, you can do all your shopping in online in any time

2. Information

Before internet came into existence, we have to refer many books if we want some information about a particular topic but now huge amount of information is available on internet about any topic related to any subject. The "search engines" will help you to find the information that you need. A very huge amount of information is available on the internet about every subject ranging from government law and services, market information, trade fairs and conferences, new start up ideas and technical support.

3. Services

We have many services in internet such as online reservation system, online shopping, mainly job seeking which is a very useful thing for all the job seekers etc. These services are not available in absence of internet.

4.Email

One of the main advantages of internet is instant communication which is achieved by Email. It is an online application. With e-mail you can send and receive electronic mails instantly. Your mails are delivered to the people instantly anywhere in the world. Email is one of the essential communication tools in business. It is also an excellent application for keeping in touch with all the family members and with friends. It is freely available on internet (no charge per use) which makes it unique when compared to fax ,telephone and postal services.

5. Downloading Software

One of the most happening and fun things to do via the Internet is Software downloading. You can download many gaming software, development software, videos, movies, music and a host of other entertainment software from the Internet, most of which are free of cost. Along with the advantages it has some disadvantages also which we have to discuss

Disadvantages of the Internet

There are certain cons and dangers relating to the use of Internet that can be summarized as:

1. Personal Information

If you use the Internet, your personal information such as your name, address, etc. can be accessed by other people. If you use a credit card to shop online, then your credit card information can also be 'stolen' which could be asking to giving someone a blank check.

2. Pornography

This is a very serious issue concerning the Internet, especially when it comes to young children. There are thousands of pornographic sites on the Internet that can be easily found and can be a detriment to letting children use the Internet.

3. Spamming

This refers to sending unsolicited e-mails in bulk, which serve no purpose and unnecessarily clog up the entire system. Such illegal activities are frustrating for all Internet users, and so instead of just ignoring it, we should make an effort to try and stop these activities so that using the Internet can become that much safer

Addiction to Internet.

By online chatting you can met bad and violent people

children spend too much time on internet which is a waste of time

viruses can destroyed your computer

Hacking of your computer

Browsers

A web browser is a software application that helps people to view web pages on the internet. It is also used to upload or download files on FTP servers. Internet uses some security methods such as SSL and TLS in order to secure internet traffic. It also uses cookies to store information about a particular website and to make internet surfing more efficiently it caches web pages. A browser will let you to visit websites and do activities within them like login, link from one site to another ,multimedia, visit one page from another, print, sending and receiving emails, and many more other activities. The most common and popular web browsers that are using today are Google Chrome ,Mozilla Firefox, Microsoft Internet Explorer, Apple Safari and the Opera browser. The steps for connecting to a website are explained below.

Types the URL for the website in any browser.

Now the browser attempts to make a connection and sends the request to the Web Server.

The Web Server receives the request and processes it.

The Web Server responds to the request with the home page of that particular website.

The requested webpage will be displayed in the browser as the response.

Internet Explorer:

Internet Explorer (IE) is one of the most popular product from software giant Microsoft. This is one of the most commonly used web browsers. This was introduced in 1995 along with Windows 95 launch and it has crossed Netscape popularity in 1998.

Google Chrome

Google chrome is developed by Google and its beta version was released on September 2 2008 for Microsoft Windows operating system. Today, chrome is known to be the most popular web browser used by maximum number of people in the world with its global share of more than 50%.

Mozilla Firefox

Mozilla Firefox is web browser designed for security, simplicity, extensibility yet flexibility and power. It is available for both Linux and android. It consists of lots of plugins and extensions with some powerful developer tools.

It delivers high performance and the user can customize the user interface. Firefox is an open architecture which can accept the installation of themes and extensions. Themes can give the browser a new customizable look. Developers can write some extensions to block all ads.

Safari

Safari is one of the web browser developed by Apple inc. It is the most popular browser in MAC users.

It is a clean browser, very fast based on the WebKit rendering engine. It comes with all the new features that you want in a modern web browser like tabbed browsing, popup blocking and built-in search functionality. It is easier to keep track of number of websites than using bookmarks.

Opera

Opera is the smaller and the faster browsers than any other browsers. Though it is smaller it is full- featured. Some of the features available in opera are fast, user-friendly, multiple windows, zoom functionality, and many more. Java and non Java-enabled versions are also available.

Slimjet Browser

Slimjet is one of the best browsers and powerful browser we have. It has an in built extensions to ease internet browsing. It is powered by Blink Engine. It supports all Chrome applications and a bundle of more extensive build in applications.

Lynx

Lynx is a full-fledged World Wide Web browser for users inUNIX, VMS, and other platforms running, character-cell terminals or emulators and cursor-addressable.

Netscape Browser

Originally it was published by AOL. Netscape browser has released its latest version based on Mozilla Firefox. It is faster, highly secure and is a choice of many online users.

Messaging E-Mail and FTP

E-mail (Electronic Mail):

E-mail is nothing but an electronic version of sending a letter. You can send an e-mail from your computer to any address around the world at any time of the day and your electronic letter will reach to its destination in seconds after you send it... even if the destination address is on the other side of the world.

Basically e-mail system consists of two parts, one the user agents and the other is message transfer agents. A user agent is nothing but a program that accepts a variety of commands such as composing an email, receiving and replying to an email. The message transfer will do the work of transferring messages from the originator to the recipient.

Advantages of E-mail:

Fast: You can send your message instantly to anywhere in the world.

Inexpensive: It costs nothing to send a mail

Simple: After initial setup it is easy to use.

Efficient: You can send a mail to the group of people in only one click.

Versatile: Images, power-point presentation file and other type of files can be sent too.

Printable: The body of the mail is easy to print.

Message

A message is a Collection of bytes that is meaningful to the applications that use it. They are used to transfer the information from one application program to another application (or in between different parts of the same application). The application may run on the same platform, or on a different platform. Apart from normal text message, voice message is also a message that will be sent to destination using voice media. Voice will itself ‘package’ and sent through the IP (Internet protocol) backbone so that it reaches to its destination ‘address’. In the technical sense, the process of sending voice packets is nothing but it is a semi passive way of communication.

Message lengths:

The default message length is 4 MB, although you can increase its size to a maximum length of 100 MB (where 1 MB = 1 048 576 bytes).

Process of Sending and Receiving an E-mail

To send an e-mail, a user must provide the message, along with the message you must have to enter the destination address and other parameters like subject of that mail and if you want to send any files then you have to attach those files to the mail etc., An attachment files can be an image file, a Word document, or one of many other supported file types.

When your recipients open your mail, they can open the file by double clicking on it with the program assigned to read and write that type of file on their computer. It will be opened only if they have the software to open that type of file.

In order to attach a file to an email, first you have to click on "+" symbol which means "composing a mail" in Mail's toolbar to open a new message window and then click on the Attach icon after entering your message and then entering an appropriate subject line and body text.

An attach icon is the paper clip. When you click on that icon you will be navigated to your hard drive and there you have to search the file(s) you want to attach to your message and click open and that file will be opened and added into attachments.

Steps for sending an e-mail:

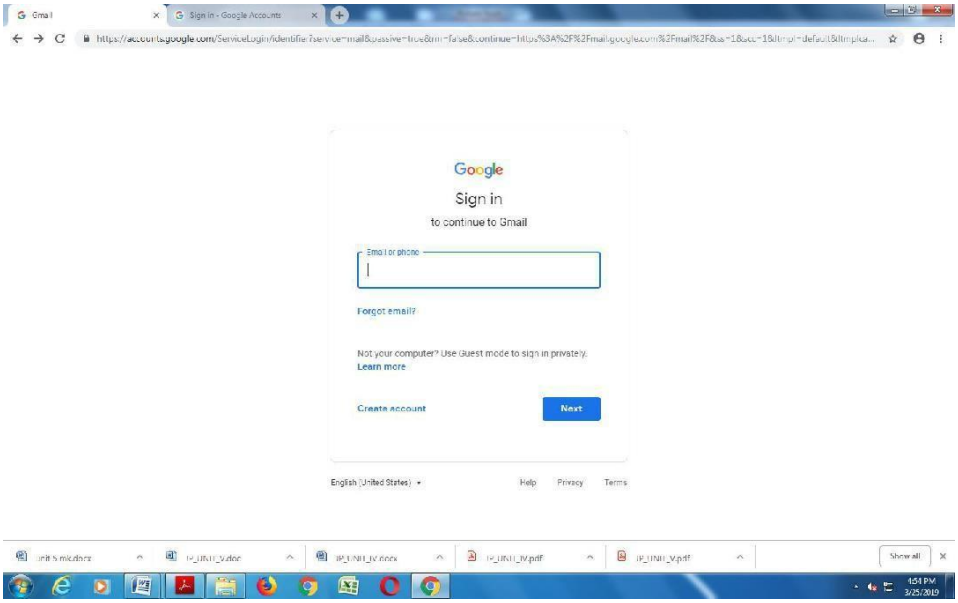
Connect your system to the internet

In web browser URL type Gmail or any other login then you will be navigated to the Gmail login page

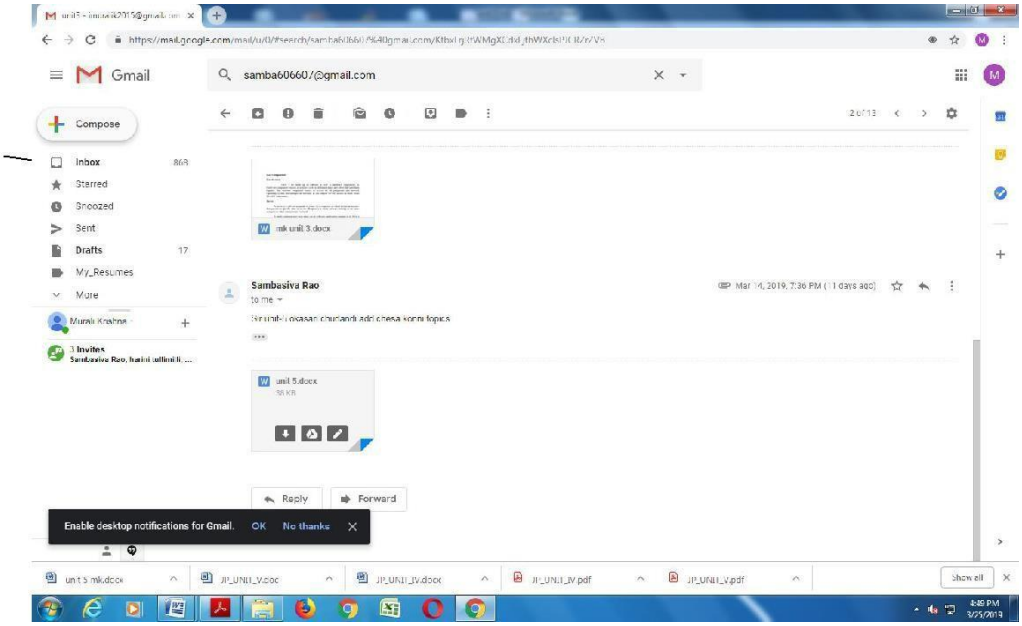
Example:

www.gmail.com

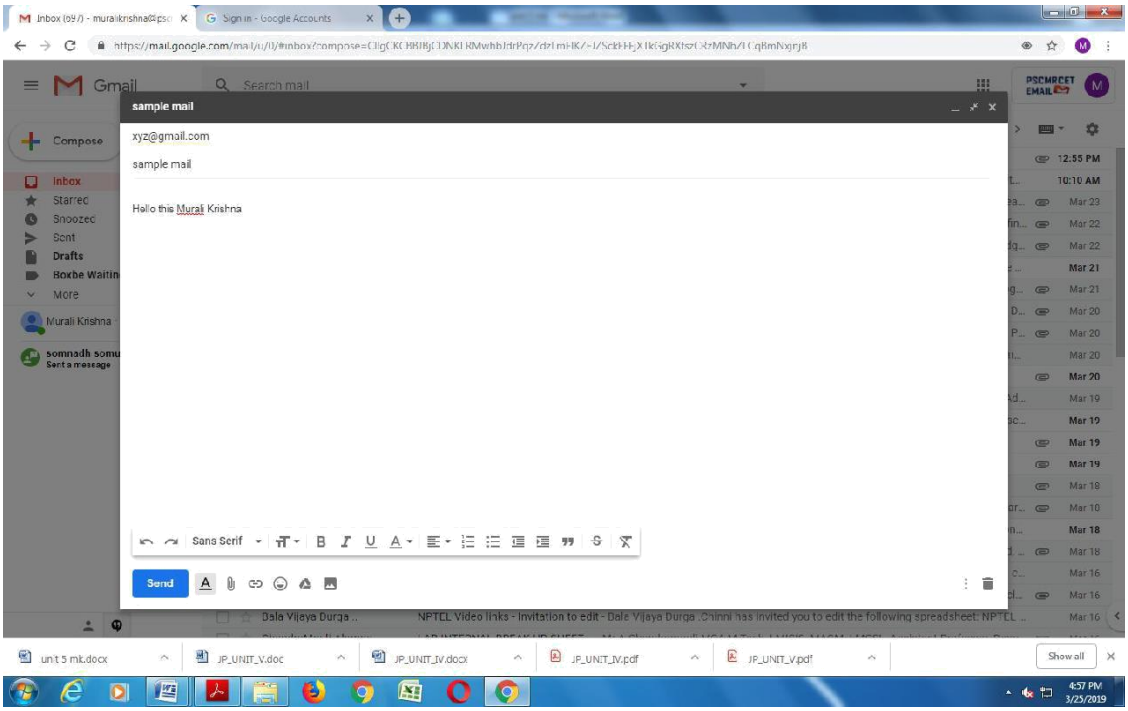
3: There you have to enter your mail address and password and login to your mail account



4: In order to send a mail first you have click on the "+" icon to compose a new mail



In that compose window first you have to enter the recipient's mail address and have to specify the subject of that mail.



Then you have to enter the body of the mail and if you want to send any file with that mail then simply you can attach the file

Then you can click on the send button to send the mail

Steps to receive an E-mail:

First you have to login to your Gmail account

If you have any new mails then those new mails will be arrived at inbox

Simply you can open that inbox page and there you can see your new arrived mails

FTP

It is a network protocol that defines some rules and conventions for communication between any two network devices. Computer networking protocols generally use packet switching technique to send and to receive messages in the form of packets. Network protocols include mechanisms for devices to identify each other and to establish connection with each other, as well as formatting the rules that specify how data is packed into messages sent or received. Some protocols support message acknowledgement also.

Hundreds of different computer networking protocols have been designed and each protocol is developed for a particular purpose and environment. Different types of network protocols are used

for different purposes like HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), TCP (Transmission Control Protocol), IP(Internet Protocol) and UDP(User Datagram Protocol).

File Transfer Protocol, or simply FTP, is a protocol used for transferring files from one computer to another – technically from your computer to a web server. FTP is the preferred method for exchanging files because it is faster than other protocols like POP or HTTP. If you want to exchange large number of files, then you should consider FTP. File Transfer Protocol data is sent and received through computer port number 21 and under the TCP protocol. The file transfer is asynchronous, that means not at the same time, because of that it is faster than other protocols.

To transfer files using FTP, we can use a program called the "client." The FTP client program will establish a connection to a remote computer by running FTP "server" software. Once the connection is established, the client can choose to send or receive copies of files in single or in groups.

In order to connect to an FTP server, a client requires a username and password as set by the admin of the server. Many public FTP records follow a special convention for accepting a username of "anonymous."

Data Structures:

File Transfer Protocol allows 3 types of data structures :

File Structure – In this the file is considered to be a continuous sequence of data

bytes and it does not have any internal structure

Record Structure – In this the file is made up of records which are arranged

in sequential manner.

3. Page Structure – In this the file is made up of independent indexed pages.

Some of the FTP commands are :

PWD – This command gives the name of the current working directory as the reply.

MKD – This command is used to create a new directory in the specified path.

RMD – This command is used to remove an existing directory from a specified path

ABOR – This command is used to abort the previous FTP service command.

QUIT – This command is used to terminate a USER only if file transfer is not in

progress, then the server will close the control connection.

LIST – This command will send a request to display the list of all the files present in

that particular directory.

FTP supports two modes of file transfer: first is the plain text file (ASCII), and second one is the binary file. We can set the mode in FTP Client

Objectives of FTP

Main objective of FTP is to promote file sharing (computer programs or data)

In order to encourage implicit or indirect (via programs) use of remote computers we use this FTP

To protect the user from the variations in the file storage systems among the hosts

To get efficient and reliable data transfer.

The basic steps in an FTP session are as follows:

First you have to start up your FTP client, by typing ftp on your system's command line prompt.
Now specify the FTP client with an address to connect. This is the FTP server address to which the FTP client will get connected automatically.
Next you have to identify yourself to the FTP remote site by entering the Login Name
Now specify the remote site with the password
Remote site will verify your Login Name and Password to allow the FTP client to access the files
Look the directory for the files in FTP server
If required you can change the Directories
Set the transfer mode (optional) andyou can get the file(s) you want and then you can quit

Introduction to Internet Security

Introduction:

Internet security is a branch of computer security which involves browser security and network security. Its objective is to maintain measures and rules to use against attacks over the internet. In fact internet is an insecure channel which allows exchange of information that leads to a high risk of intrusion or frauds such as phishing. Internet security focuses on prevention of threats such as online viruses, Trojans and worms and more.

Viruses

A computer virus is malicious software that can infect other programs by replicating itself and inserting its own code. The main characteristic of viruses is that they are self-replicating programs which modify other software without user involvement. Viruses performs some type of harmful activity on infected host computers such as occupying secondary memory space and access the private information, corrupts the data, spamming the email contacts and often render the computer useless.

There are different types of viruses. Some of them are:

Polymorphic virus: This mutates with any type of infection. This virus cannot be detected using signature of virus.

Memory-Resident virus: This virus loads in main memory which behaves as part of a resident system program. From there it starts infecting every program that executes.

Parasitic virus: It is most common form of virus. It gets attached to the executable files and replicates itself without user involvement. It gets replicated when the infected program is executed, by finding other executable files to infect.

Boot Sector virus: It infects the boot record of the system and spreads when a system is booted from the disk containing virus.

Trojan Or Trojan Horse

It is a malicious program that is developed by hackers to gain access to the users systems. Loading of these Trojans into users systems is done by applying tricks to attract the users by social media adds. Whenever the user is directed to the malicious website Trojans gets loaded and executed in the users systems. Trojans are used to spy on the targeted user where cyber-criminals may get illegal access to the system. They do not attempt to inject themselves into other files like a computer virus.Trojan leads to deletion of data, copies data, edits data, degrades the performance of target computer.

Types Of Trojans:

There are many types of Trojans which steals your credentials, spy on users activities and many other.

Trojan-Downloader: It installs other malware into system.

Trojan-Banker: These programs extracts the account related information such as credit card pins, online banking OTPs etc.,

Ransomware: This Trojan directly encrypts the data on targeted device. The hackers who maintains this Trojan may demand money as ransom for decrypting the device. Criminals provide a decryption key without which recovering the data is very difficult.

Trojan-Rootkits: These are sophisticated malware which prevents the detection of malware and malicious activities on the computer.

Trojan-Backdoor: It is a commonly used Trojan. It allows the cyber criminals to access the computer by creating a backdoor.

Worms

A worm is a malware which sends copies from computer to computer by replicating itself. The worm in the infected machine will definitely searches another machine to replicate itself. These worms use network connections to spread worm over network. The worm which is in active state, will behave as computer virus or bacteria. In order to replicate itself worms uses some network vehicles. Some of them are:

Electronic mails: A worm mails itself to other systems.
Remote login capability: A worm uses a remote system. It logins into it as a user and it applies
commands to replicates itself from one system to another.
Remote execution capability: A worm executes a copy of itself on another system.

PHISHING AND HACKING

Phishing is a cyber-attack that is used to gather personal information using deceptive websites and emails. In this method perpetrator sends some legitimate looking email in order to gather personal or sensitive information from the recipients. In general they trick the email recipients to believe that the message is from well-known or trust worthy websites. The users are directed to visit a website which looks like a genuine website where they are asked for personal information such as passwords, bank account numbers and credit card details etc., where the website is only a setup to steals the users information. The real distinguisher of phishing is that the attackers masquerade as a trusted entity. There are some phishing kits available which makes the cyber criminals work easy.

PREVENTION OF PHISHING

A user can follow a number of steps to avoid phishing. Always check the spelling of URL before checking the links. If an email is received by from a known source which it seems to be suspicious contact that source with a new email rather than clicking the reply. Don’t post personal data on social media publicly. Watching out the URL directions where the user is sent to a different website with identical design.

HACKING

Hacking is a method that is used to compromise the security of digital devices which are over the network such as computers, smart phones, tablets etc. Hacking is an unlawful activity done by the cyber criminals named as hackers for various types of reasons. They are motivated by information gathering that is spying, financial gain, protest, even just for fun. Hacking involves creation of

malvertising that deposits malware in a user's system which does not require a user interaction. Hackers use the system to work as a server.

HACKING PREVENTION:

Use anti malware product which both detects and block malicious phishing websites. Download only legitimate mobile apps which police themselves for malware. Always make sure your operating system is updated. Avoid visiting unsafe websites and downloading unverified attachments or clicking on links in unfamiliar emails.

ANTI VIRUS and Anti Virus Software

In order to get protected from the viruses ANTIVIRUS software is used. An antivirus is a set of programs which are designed to prevent, detect, and to remove software viruses and other malicious software like worms, Trojans etc.

Computer viruses are set of programs that are designed to interfere with computer operations such as recording, corrupting or deleting data or replication of themselves to other systems using network.

To help preventing these viruses anti-virus software must be updated regularly.

List of Anti Virus Software

McAfee

McAfee's advanced retail desktop solutions include premier anti-virus, security, encryption, and desktop optimization software. McAfee's managed Web security services employ a patented system and process of delivering software through an Internet browser to provide these services to users online through its Web site http://us.mcafee.com, one of the largest paid subscription sites on the Internet with over two million active paid subscribers.

McAfee gives you two ways to protect your PC against nasty viruses and worms like Sasser, Mydoom and Lovsan/MSBlaster. You can purchase McAfee VirusScan for real-time virus protection or McAfee Web Essentials for 2-in-1 protection from viruses and hackers.

Fig McAfee Anti Version Outlook

McAfee VirusScan	**McAfee Web Essentials**
• Protects your PC from over 87,000 viruses.	• Protects against viruses, worms, Trojans, and hackers.
• Detects spyware and adware protecting loss of data and privacy.	• Detects and quarantines spyware, web dialers, adware and more.
• Scans e-mail, downloads, CD-ROMs, & Instant Message attachments.	• Free updates and upgrades during your subscription.
• WormStopper detects and stops mass-mailing worms.	
• Detects and attempts to clean malicious code.	
• Free updates and upgrades during your subscription.	

Norton Antivirus

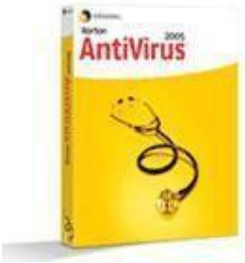


Fig Outlook of Norton Antivirus

Symantec's Norton AntiVirus™ 2005

Symantec's Norton AntiVirus 2005 is the latest version and it is the world's most trusted antivirus solution. It removes viruses, worms, and Trojan horses automatically without interrupting your work. New Norton™ Internet Worm Protection blocks certain more sophisticated worms (such as Blaster and Sasser) before they enter your computer. Norton AntiVirus can also detect spyware and other non-virus threats.

Norton™ Internet Worm Protection stops certain damaging Internet worms at their attempted point of entry. QuickScan tool automatically searches for and removes viruses whenever new virus protection updates are downloaded. PreInstall Scan quickly detects and removes infections that can interfere with installing and launching Norton AntiVirus. It will scans and cleans both incoming and outgoing email messages. It can blocks viruses in instant message attachments. It will detect spyware and certain non-virus threats such as adware and keystroke logging programs. On the other hand, it will download new virus protection updates automatically to protect against new threats.

Email precautions

Like the telephone, email is one of the quickest and most convenient forms of communication today. We send them through the office, to our children, and to our clients. We use it because it is quick and easy. We have electronic address books instead of little black books. Email frauds commit their crimes in a way similar to con artists who commit their crimes over the telephone. Creators of email viruses prey on you by claiming to be from somebody that they're not. The good news is, you really don't have to be a victim, if you're armed with the right information.

The easiest way to prevent a virus is to not engage in the activity that causes it to spread. If you're not willing or able to give up your email entirely, there are a few guidelines to make your email use a lot safer. Don't download files from strangers. If you simply must read the attachment, however, download it to a floppy disk to be on the safe side, and then scan it with your anti-virus software. This is the safest way to handle downloads because the file is not accessible from a network drive or your hard disk. If you get a virus warning sent to you, make sure to check your software provider's website to make sure it's accurate.

Identifying Potential Viruses in E-mail Attachments

Whenever you receive an attachment in MS Outlook / Outlook Express there is generally a picture of a paper clip. Single clicking the paper clip reveals that there are 2 files attached. Notice that the attachment above appears to be a Word Document because of the **.doc** extension. However, the icon does not appear to be a typical MS Word icon. The typical MS Word icon is a picture of a **blue 'W'**, as shown below (at left):

Fig: "Good Attachment" vs. "Bad Attachment"

Notice how the attached file in the e-mail above is called "**The Pillarbanquet wine.doc**" but has a different icon (depicted above right). In this case, the file is really a **batch file**, which contains potentially dangerous commands. The entire file name of the above file is "**The Pillarbanquet wine.doc.bat".** If you clicked on this icon, any commands that are contained within would be executed, including **deleting all your files** or **reformatting your hard drive**. Depending on how your computer is configured, it may not be set to show file extensions or extensions of registered files or system files. This means that you may never see certain types of extensions (.bat, .com, .exe, .pif, etc.).

Good Files

Typically, "good" attachments will have a picture of their associated program, such as **MS Word, Excel, Acrobat Reader**, etc.

Fig: Good File Attachments

Even "good" files can contain viruses. There are some macro viruses that are embedded in MS Office documents (Word, Excel, etc.). The best bet is to always make sure that your virus definitions are up-to-date.

Bad Files

There are a number of files that can cause problems. Anything that ends in **.bat, .com, .exe, .scr, .vbs, .lnk, .pif** can cause external commands to be run. Below is a sample of what the icons might look like:

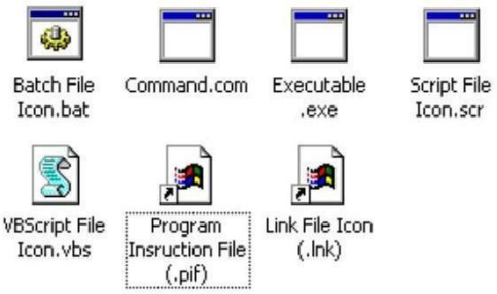


Fig : List of Bad Files

Firewall

A Firewall is software that is used to provide the security of a private network. It prevents unauthorized access to private networks connected to the network. It provides better safety where data can be encrypted. Firewall may be implemented using hardware, software or both. It serves as a blockade between trusted internal network and untrusted external network.

Following are the different types of firewall

Packet filtering:

On the Internet, packet filtering is the process of passing or blocking packets at a network interface based on source and destination addresses, ports, or protocols. The process is used in conjunction with packet mangling and Network Address Translation (NAT). Packet filtering is often part of a firewall program for protecting a local network from unwanted intrusion.

In a software firewall, packet filtering is done by a program called a packet filter. The packet filter examines the header of each packet based on a specific set of rules, and on that basis, decides to prevent it from passing (called DROP) or allow it to pass (called ACCEPT). There are three ways in which a packet filter can be configured, once the set of filtering rules has been defined. In the first method, the filter accepts only those packets that it is certain are safe, dropping all others. This is the most secure mode, but it can cause inconvenience if legitimate packets are inadvertently dropped. In the second method, the filter drops only the packets that it is certain are unsafe, accepting all others. This mode is the least secure, but is causes less inconvenience, particularly in casual Web browsing. In the third method, if the filter encounters a packet for which its rules do not provide instructions, that packet can be quarantined, or the user can be specifically queried concerning what should be done with it. This can be inconvenient if it causes numerous dialog boxes to appear, for example, during Web browsing.

Proxy servers

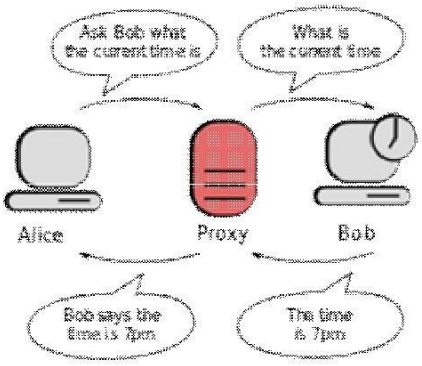


Fig: Proxy Server Architecture

Communication between two computers (shown in grey) connected through a third computer (shown in red) acting as a proxy. Bob does not know to whom the information is going, which is why proxies can be used to protect privacy.

In computer networks, a **proxy server** is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems.

Application gateway:

Application gateways provide high-level secure network system communication. For example, when a client requests access to server resources such as files, Web pages and databases, the client first connects with the proxy server, which then establishes a connection with the main server.

The application gateway resides on the client and server firewall. The proxy server hides Internet Protocol (IP) addresses and other secure information on the client's behalf. A computer's internal system may communicate with an external computer using firewall protection. The application gateway and external computer function without client information or knowledge of the proxy server IP address.

Dynamic packet filtering:

A dynamic packet filter is a firewall facility that can monitor the state of active connections and use this information to determine which network packets to allow through the firewall. By recording session information such as IP addresses and port numbers, a dynamic packet filter can implement a much tighter security posture than a static packet filter.

For example, assume that you wish to configure your firewall so that all users in your company are allowed out to the Internet, but only replies to users' data requests are let back in. With a static packet filter, you would need to permanently allow in replies from all external addresses, assuming that users were free to visit any site on the Internet. This kind of filter would allow an attacker to sneak information past the filter by making the packet look like a reply (which can be done by indicating "reply" in the packet header).

By tracking and matching requests and replies, a dynamic packet filter can screen for replies that don't match a request. When a request is recorded, the dynamic packet filter opens up a small inbound hole so only the expected data reply is let back through. Once the reply is received, the hole is closed. This dramatically increases the security capabilities of the firewall.

Circuit level gateway:

A proxy server is a security barrier between internal and external computers, while a circuit-level gateway is a virtual circuit between the proxy server and internal client.

For example, when a user Web page access request passes through the circuit gateway, basic internal user information, such as IP address, is exchanged for proper feedback. Then, the proxy server forwards the request to the Web server. Upon receiving the request, the external server sees the proxy server's IP address but does not receive any internal user information. The Web or real server sends the proxy server a proper response, which is forwarded to the client or end user via the circuit-level gateway.

Summary

The **internet** in simple terms is a network of the interlinked computer networking worldwide, which is accessible to the general public.

There many advantages to using the internet such as Email, Information, Services, Buy or sell products, Online Chat, Downloading Software.

A **browser** is software that is used to access the internet. • The most common browser software titles on the market are: Microsoft Internet Explorer, Mozilla Firefox, Apple Computer's Safari, and Opera.

A **message** is a string of bytes that is meaningful to the applications that use it.

Messages are used to transfer information from one application program to another.

E-mail is an electronic version of sending a letter. You can send e-mail from your computer at any time of the day to any address around the world and your electronic letter will arrive at its destination seconds after you send it... even if the receiver lives on the other side of the world.

A **network protocol** defines rules and conventions for communication between network devices.

File Transfer Protocol, or FTP, is a protocol used for transferring files from one computer to another - typically from your computer to a web server.

Internet security is a branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole.

A **Virus** is a piece of software that can infect other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

A **worm** is a program that can replicate itself and send copies from computer to computer across network connections.

A **Trojan horse,** or Trojan, is a malicious application that tricks as a authentic file or helpful program but whose real purpose is, for example, to grant a hacker unauthorized access to a computer.

Phishing is an e-mail fraud method in which the perpetrator sends out legitimate - looking email in an attempt to gather personal and financial information from recipients.

Antivirus software is a computer program that detects, prevents, and takes action to disarm or remove malicious software programs, such as viruses and worms.

A **Firewall** is software that is used to provide the security of a private network. It prevents unauthorized access to private networks connected to the network.

On the Internet, packet filtering is the process of passing or blocking packets at a network interface based on source and destination addresses, ports, or protocols.

A proxy server is a security barrier between internal and external computers, while a circuit-level gateway is a virtual circuit between the proxy server and internal client.

In computer networks, a **proxy server** is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.

A dynamic packet filter is a firewall facility that can monitor the state of active connections and use this information to determine which network packets to allow through the firewall.

Application gateways provide high-level secure network system communication. For example, when a client requests access to server resources such as files, Web pages and databases, the client first connects with the proxy server, which then establishes a connection with the main server.

Short Answer Type Questions

1. What is an Internet ?
2. Write advantages of Internet.
3. What is a Browser ? List types of browsers.
4. What is a Message ?
5. What is an E-mail ?
6. Write any four uses of E-mail.
7. What is an attachment ?
8. What is voice messaging ?
9. What is Internet Explorer ?
10. What is FTP ?
11. What is a Protocol and Write types of Network Protocols ?
12. What is Internet Security ?
13. What is a virus ?
14. What is a Trojan ?
15. What is Hacking ?
16. What is a Worm ?
17. Expand FTP, E-Mail, WWW, TCP/IP
18. Expand ISDN, HTTP. FTP, NIC.
19. What is Firewall ?

Long Answer Type Questions

1. What is an Internet ? Explain any three advantages and three disadvantages of Internet.
2. Explain various Web Browsers.
3. Write various advantages and disadvantages of e-mail.
4. How do you send and receive an E-mail with attachment.
5. Write about Internet Security.
6. Explain different type of Firewall

UNIT 6

TROUBLE SHOOTING

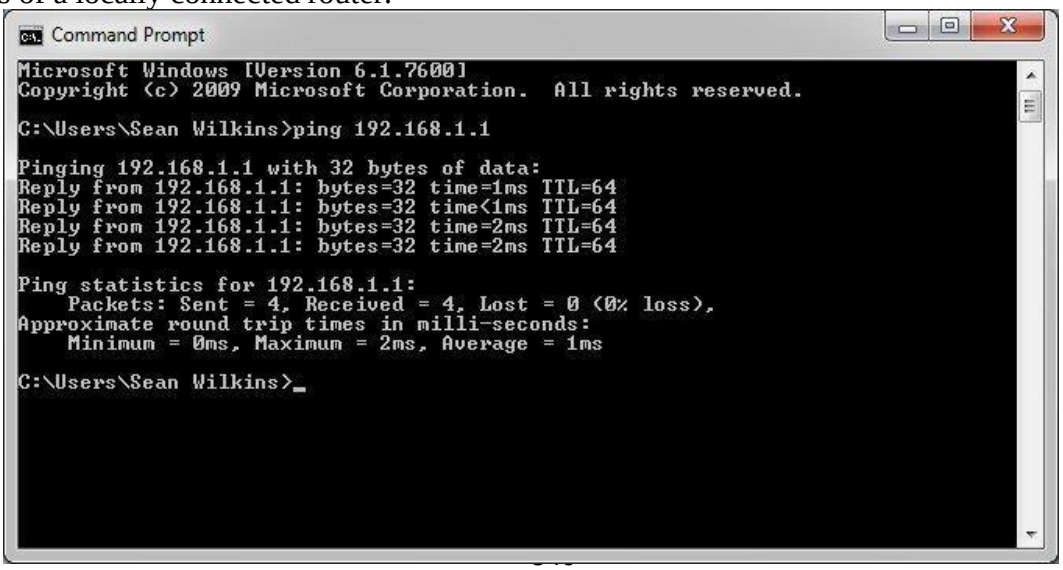
Introduction To Trouble Shooting:

Trouble shooting is a process in which certain measures and processes are collectively used to identify and fix problems which are encountered with in a computer network. It is a systematic process which aims to resolve problems or to optimise the network. Some of the processes used in network trouble shooting are:

- Identifying and resolving problems and establishing network connections of a device
- Configuration of network management devices such as router, switch etc.,
- Installing cables or Wi-Fi devices
- Removing viruses
- Adding, configuring and reinstalling a network printer

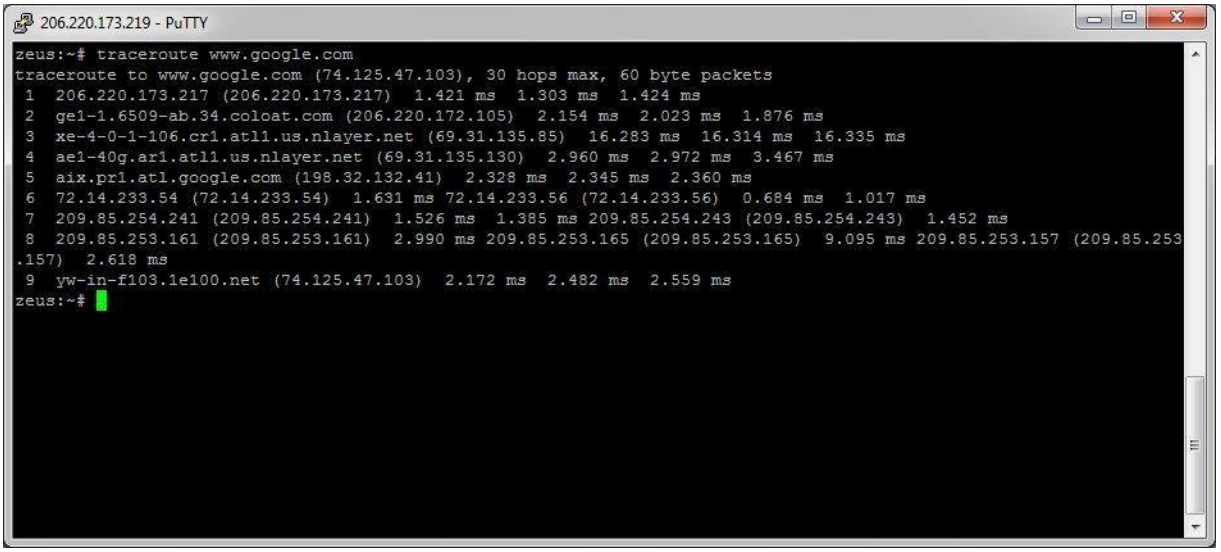
Trouble shooting processes are not only limited to above. They can be manual or automatic tasks. Network diagnostic software can be used for network management while using automated tools. There are some tools which helps in trouble shooting most networking issues.

1. **Ping**: This utility is used to perform basic connectivity tests between requesting and destination hosts. Internet control message protocols (ICMP) is used for this which sends an echo packet to the destination and it listens response from the host. If the requesting host receives the response from the destination the host is reachable this utility is generally used to check where specific networking problem exists that is whether the problem exists at the receiving host or at the destination address. This utility is used to obtain reachability status of a locally connected router.



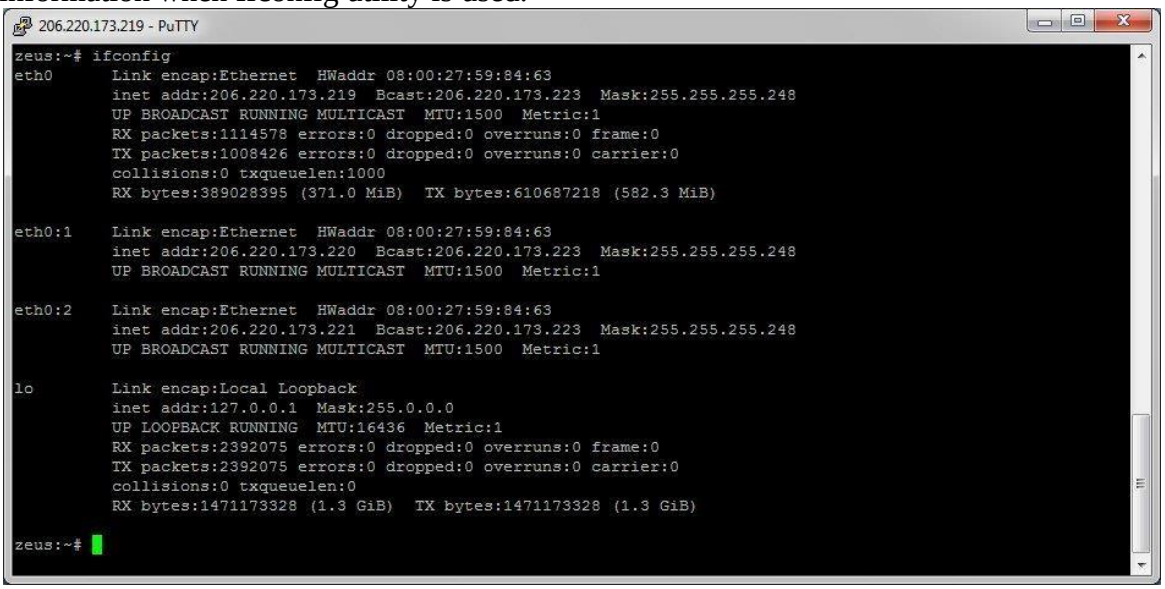
1: Ping utility

2. **Tracert/traceroute:** Once the basic connectivity test has been done using ping utility, the tracert/traceroute is used to obtain more specific information about the path to the destination host which includes the route that packet follows, response time of these intermediate hosts. Both tracert/traceroute utilities have same functionalities but they differ in operating systems which they operate in. Tracert is used for windows and traceroute is for Linux based machines.



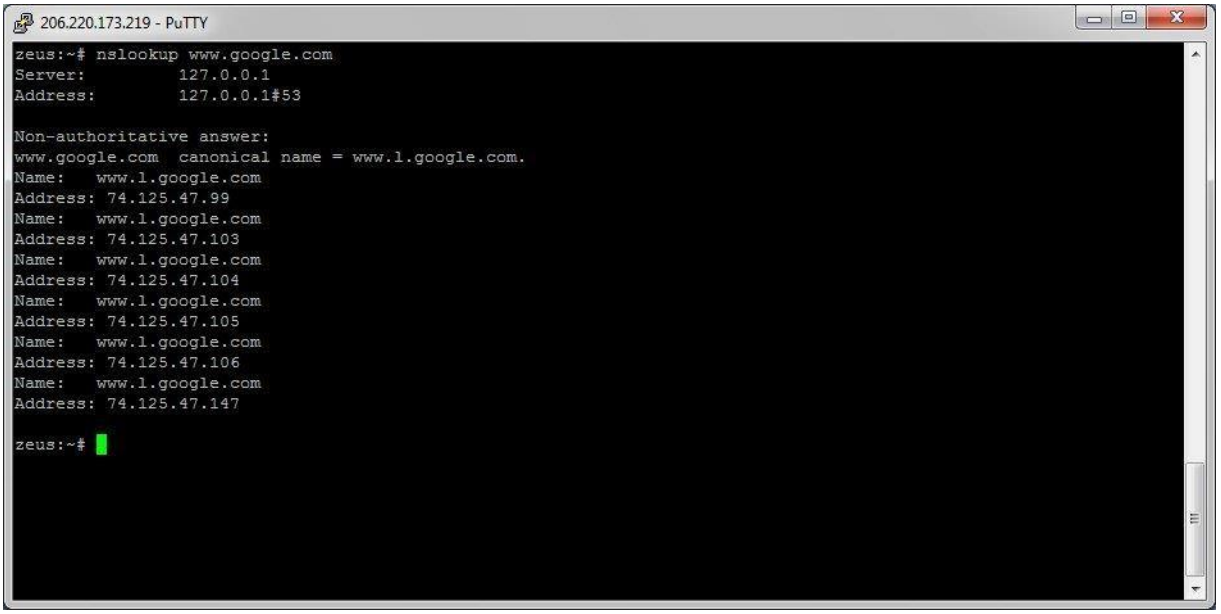
2: Tracert/traceroute utility

3. **Ipconfig/ifconfig:** Identifying the specific ipconfiuration of variously affected hosts is the major thing that should be done when trouble shooting a networking issue. When address configuration is done statically. This information is already known but in case of dynamic addressing ip address of each host will change often. Thus these utilities can be used to find out this information. Ipconfig is used on windows and Ifconfig is used on Linux based machines. Below is the figure showing Ipconguration information when Ifconfig utility is used.



3: Ifconfig utility

4. **Nslookup**: Many of the networking issues involves in issues with dynamic name system(DNS) address resolution issues. The Nslookup system utility is used to lookup the specific ip address associated with the domain name. If this utility fails to resolve this information there is a DNS issue.



Nslookup utility

Similarly there are many types of utilities such as netstat, putty, pathping, route which helps in trouble shooting issues in a network connection.

Trouble Shooting Of Motherboard:

Motherboard and processor are most important hardware components of a computer. Entire communication between the hardware components inside the pc is done through the circuits in the motherboard. Thus replacing the motherboard in case of failure is very expensive but diagnosing of hardware on failure will reduce the repair costs.

The motherboard itself is a computer, thus a failed motherboard exhibit a usual symptom of completely dead system. When the motherboard is dead fans, drives and other peripherals may spin up, but nothing at all happens when you turn on the power. The common symptoms of a motherboard issues are very similar to CPU problems:

- The system does not display anything
- One or more beeps occur
- Appearing of error code
- System locks
- The system reboots
- Windows BSOD (blue screen of death) appears
- Failure of memory module

It is most difficult to troubleshoot the motherboard problems and power problems.
Below are the some of the steps that helps in troubleshooting motherboard.

- Check whether the motherboard is receiving power or not.
- Check the BIOS settings.

- Power off the computer and let it cool down. Check for over heating.
- Boot the computers and remove unnecessary adapters and devices.
- Check whether there is any short in motherboard.

Refer to the online mother board documents to resolve the problems and find a solution.

FDD, HDD, CDROM

FDD (FLOPPY DISK DRIVE):

A FDD is a hardware device which reads data and stores information. They are used for both reading and writing on removable floppy discs.Now floppy disks are replaced by other storage devices.The floppy disk helps to store huge amounts of data outside of a computer's hard drive for personal use because they were very low cost and easy to carry.

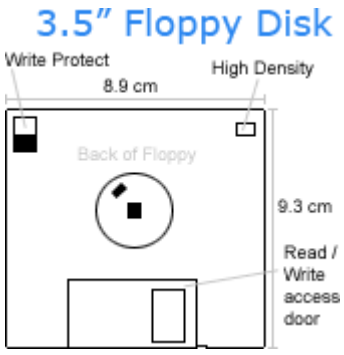
FDD's has four basic components:

- There will be one or two magnetic heads for both reading and writing.
- It spins at a rate of 300 to 360 rotations per minute.
- A spindle, which helps to hold the disk in a certain place.
- A frame with levers that opened and closed the device
- A circuit board which contains all of the electronics.

The same head is used for both reading and writing. Heads can read both sides of the disk. An additional head is used for erasing the data. This is a wider head which make sure that no interference took place with the adjacent track while data on one track is getting erased.Now a days floppy drives have become out dated. These were the devices which were used in the past. Now these are replaced by many other such as ZIP drives, CDs, USB etc.

TROUBLE SHOOTING IN FDD:

Bad floppy diskette



- Check whether the floppy disk we are attempting to read is not bad. It should not write protected.

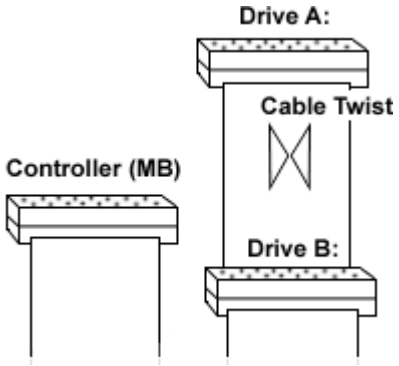
- Checking whether it is write protected can be done by sliding the tab into opposite direction thus preventing the light from shining through it.
- If there is no tab, place tape over this hole.
- Because of the functionality of floppy disk drives, it is likely for a floppy disk to become bad.
- Verify that other floppy disks are not exhibiting the same issue.
- If other floppies work well, it is likely that you may have a bad floppy diskette.

Not setup in CMOS

- Verify the floppy drive is properly set up in CMOS setup.
- If the floppy drive is not set up properly, you may experience read/write errors or the floppy may not work at all. Most computers need to have the floppy setup as a 3.5, 1.44 MB.

Not connected properly

- Power down, unplug, and open the computer, being aware of ESD and its potential dangers.
- Verify the floppy connection is connected to the motherboard FDD connector. If it appears to be connected, disconnect and then reconnect the cable and check the cable is seated properly.
- Verify the floppy cable coming from the motherboard is connected to the back of the floppy drive. If connected, disconnect and reconnect the floppy drive cable to verify it is seated properly.



- Verify a power cable is connected to the back of the floppy drive.
- If your floppy cable has more than one connection, verify that you have connected the floppy to the appropriate connection. The above picture illustrates what drive should be connected where.

Bad drivers

If you are not able to read or write to a floppy diskette from Windows,

- Verify the computer is not exhibiting floppy driver issues by testing the floppy drive from MS-DOS.

- If you are running Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000, or Windows 2000, click Start, Shutdown, and Restart the computer to an MS-DOS prompt. If you are running latest version windows, click Start, type cmd to get the prompt.
- Once at the prompt, place a diskette into the floppy disk drive and type **format a**
- If you receive any of the following errors when trying to format the floppy disk:
  - **Invalid media or Track 0 Bad** - Try formatting another floppy disk.
  - **Write Protect Error** - Ensure the disk is not write protected by removing the floppy disk and verifying you cannot see through both holes. If you can see light, move the tab on the left side (looking from the back of the floppy) to the down position and try again. If the disk is not write protected, try another floppy disk.
  - **Invalid drive specification** - Verify the floppy drive is set up properly in CMOS setup.
- If the floppy disk drive formats properly in MS-DOS but does not format in Windows, it is likely that Windows or a program within Windows is preventing the floppy drive from working. End task all running background programs. If you continue to experience the same issues, we recommend that Windows be reinstalled.

Bad hardware

If you continue to experience issues after following the above steps, it is likely that some of the hardware in the computer is bad. Replace the following hardware in the computer in the below order.

1. Replace the floppy data cable that connects the computer floppy drive to the motherboard.
2. Replace the floppy driveif the floppy data cable did not resolve your issues.
3. Replace or request that the motherboard be replaced.

HDD (HARD DISK DRIVE):

HDD is a computer storage device which has magnetic disks rotating at high speeds. In a system it acts as a Secondary storage device where data is stored permanently (where RAM works as primary storage device). In general it is non-volatile in nature where data is retained when the computer is turned off.

The disk/platter rotates at a rate of 5,400 to 15,000 PMwhich allows data to be accessed immediately. When the platter rotates an arm with read/write head moves across it in order to read or write data form or to it respectively. As volatile in nature the data gets saved when the power is turned off.

In order to control read/write heads when, where and how to move across disks, a ROM (Read Only Memory) controller board is needed. Similar to FDD, both sides of the disk are used to store data. Each side of every disk is said as head, which is divided into several sectors and tracks. Every track will have the same distance from the centre. Collection of tracks is known as cylinder. While writing the data, read/write heads move inward from one cylinder to another once it gets filled.

A hard drive is divided into one of more partitions, which can be further divided into logical drives or volumes. Usually a master boot record (MBR) is found at the beginning of the hard drive which contains a table of partition information. Each logical drive contains a boot record, a file allocation table (FAT) and a root directory for the FAT file system.

TROUBLE SHOOTING HDD:

A bad hard drive may cause various issues on your computer. Here are some of the possible issues. (These issues can also be caused not only by a bad disk drive but also for various other reasons)

1. Errors when reading, copying, moving, or deleting data on the computer.
2. Extremely slow.
3. Operating system unable to boot.
4. Other random errors or computer reboots.

Software solutions

Below is a listing of a software programs available that are designed to test your computer's hard drive for errors.

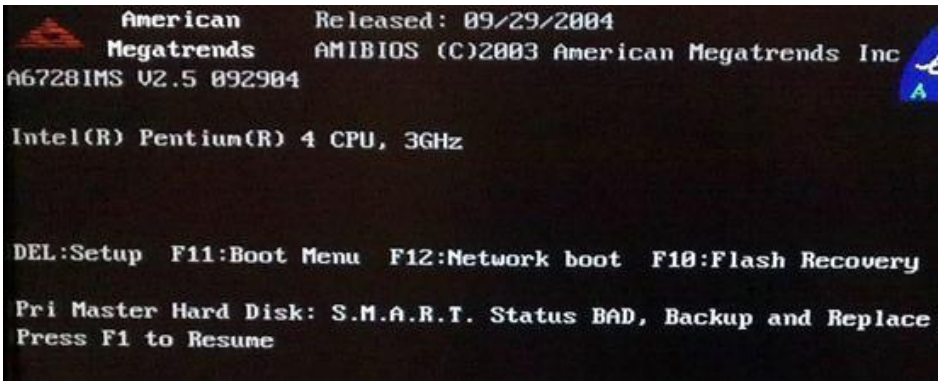
- **ScanDisk** - Users running Microsoft Windows who're still able to get into Windows can utilize the already installed disk checking tool ScanDisk to find and repair any errors on their hard drive.
- **Chkdsk** - Another Windows command line utility to test the hard drive. If you're unable to boot into Windows, boot from the Windows CD, enter the recovery console, and run **chkdsk /f** to fix errors.
- **TestDisk** - Fantastic free and open-source utility to test and fix different hard drive errors.
- **Seagate SeaTools** - Fantastic and free program that is used to test all computer hard drives.
- **HDD health** - Another great program that utilizes S.M.A.R.T technology to display a hard drive's statistics, such as the temperature of the drive, it's overall health, and every other SMART attribute.
  S M A R T is the short form of Self-Monitoring Analysis and Reporting Technology. S.M.A.R.Tanalyzes many of the mechanical attributes, over time, some failures can be predicted. Although S.M.A.R.T technology can detect upcoming issues, not all hard drive failures are predictable.

Another option is to download the files to make an Ultimate Boot CD. The Ultimate Boot CD contains multiple tools for testing computer hardware, including hard drives, and help fix

Replace the hard drive

If the hard drive appears to be bad or is generating SMART errors indicating its bad even after trying the above suggestions, replacing is the better option.

The picture below is an example of a S.M.A.R.T error indicating that the status of the *Primary*(master) hard drive is being detected as bad and that it should be replaced.



In general, hard drives have a several year warranty and if the drive or the computer may not be new, it may still be under warranty.

CD ROM:

CD ROM Stands for "Compact Disc Read-Only Memory." A CD-ROM is a CD that can be read by a computer with an optical drive. The term "ROM" means the data on the disc is "read-only," or cannot be altered or erased. CD-ROMs share the same technology as audio CDs, but they are formatted differently, which allows them to store many types of data.

TROUBLE SHOOTING CD-ROM

A bad disk drive can cause many issues. Such as

- Error when reading CD or DVD.
- CDs or DVDs may not play audio or video properly.
- CD or DVD programs may not install or encounter errors after being installed.

There are different ways to test CD drive to determine if it is bad.

Below are some of the solutions to resolve the issues.

CD and DVD drives can sometimes get dusty and dirty inside, causing problems with reading discs. Using a CD/DVD Drive Cleaner kit helps to clean the disc drive.

Some of the software programs available to test your computer's CD and DVD drives and

- **CDRoller** - Great program that is used to test and, if needed, recover data from CDs and DVDs.
- **CDCheck** - Another great program used to help check CD drives (no DVD compatibility) and can also be used to help recover data from damaged discs.

There are some other commercial products which helps to resolve. Below is one of these products.

- **CD/DVD Diagnostic** - Diagnostic program from InfinaDyne that is not only capable of testing a drive and disc, but also can recover some or all data from discs that may be bad.

Check the power supply

If the power supply is not providing enough power to the disc drive, it results in improper functionality. In case of insufficient power disc drive may not be able to read disc properly. You can check the power supply to determine if it is not working correctly.
If the disc tray will not open, that may also be a result of the power supply not providing any power, or enough power, to the disc drive.

Replace the disc drive

If the disc drive is still unable to read any CD or DVD after trying the above suggestions, the disc drive is most likely bad. It is recommended that you replace the disc drive to fix the problem.

PRINTERS,MODEMS, MONITORS AND SMPs

PRINTERS:

A **printer** is an external hardware output devicethat takes the electronic data stored on a computer or other device and generates a hard copy of it.

Trouble Shooting A Printer

When a printer is unable to print, there may be a possibility that either the printer is experiencing a hardware or software problem. Before testing the software, it is always recommended that the printer is tested for hardware issues first.

Testing Hardware

1. Check that paper is loaded and that there are no paper jams.
2. Check the cable connections. Both the power cable and data cables (e.g., USB cable) going from printer and to your computer.
3. Verify that the printer power indication light is turned.
4. Verify that the printer has no flashing lights or Red or Orange lights. Having red, orange, or any other flashing lights indicates malfunctioning of printer.
5. Run a self-test on the printer. Running a self-test should print a basic page of information indicating the printer is physically working. Your users manual should have the instructions for printing a self-test. If your printer does not print a self-test, it's a good possibility that there is a defect or misconfiguration with your printer, and you should contact the printermanufacturer.

Testing Software

Windows users

1. If possible, make sure your printer can print using the above hardware tests.
2. Install the printer software provided with your printer. If no software was provided with your printer, see the printer drivers page for a listing of printer drivers. Drivers are required for your printer to properly work with the operating system.
3. Click Start, Settings, Printers. Within the printers window, verify that your printer manufacturer and model is listed.
4. Print a test page by right-clicking the printer icon and then click the Properties option and click print test page. If the test page does not print, download the latest drivers from your printer manufacturer. For a list of of printer drivers, see our printerdriversindex.
5. If the page prints successfully, get back into the printers window, right-click the printer and ensure that there is a check next to Set as default. Newer versions of Windows also show a check next to the printer icon once it has been set as the default printer.
6. Once the printer is set as default, click Start, Run, type notepad, and press Enter. In Notepad, type a test message and click File and Print. If the printer also prints successfully from this program, but you are still unable to print from another program, it is likely that program has an issue and not your printer.

MODEMS

Modem RESET

If your Internet connection isn't working or you've mucked around with your modem settings, you might be wondering if hitting the modem reset button will solve your problems. Maybe you're wondering what happens if you simply unplug your modem. Before you do anything, read on.

The reset button is on the back of your modem. It's probably red and labeled RESET. Since it's not meant to be used often, it's recessed to prevent it from being accidentally pressed. You will need to use the tip of a pen or the end of a paperclip to reach it. You will likely feel a slight click when you press it.

MONITORS

Monitor is one of the most important parts of your pc which displays the output from the computer. Text and graphics video modes are the most basic ones displayed by a monitor.

Monitor acts as an intermediate between user and computer which shows graphical interface, otherwise it will be difficult to know what is going on inside the computer

TROUBLE SHOOTING A MONITOR: Unlike other computer parts, monitor troubleshooting requires more technical knowledge about repairing electronics. This is because some parts of a monitor are to be replaced or repaired. This requires tools and knowledge.

Below are some of the basic repairing steps which can be carried out.

Common problems of computer monitor

If computer screen fails to work, communication with PC cannot be done.
Below are some of the reasons for screen problems that can be easily traced and fixed at user level. Other than these kinds of computer monitor troubleshooting, repairing the inside part demands knowledge and involves risks as well. So, it is advisable not to try to repair it by own

The most common computer monitor problems are as follows.

Display problem

Distorted, shaking, wavy, dithering screen is the result of power interference and related problem. If your computer is near to other equipments such as TV, radio, refrigerator or other electric powered equipments, changing the place or power line will resolve these kinds of problems. In addition, turning on and off the monitor or if it is built in with screen menu, degaussing would also resolve it.

For unreadable screen, this is the problem related with **improper configuration of display properties**. It is also caused by incorrect display driver. Remove the driver and install it again with new and updated version.

Power cable related problem

Checking the power cable is the simplest method you can do at first when doing computer monitor troubleshooting. Always make sure that the power cord is properly fitted and tight. If it is defective you should change it before it brings serious damage to your computer monitor.

Display graphics card problem

Display graphics card (VGA card) is the other reason for monitor problem. In modern systems graphics card is integrated with the motherboard. This card may fail to work. If this happens, you can still buy a separate video card and hook it on the available **PCI express slot**.

However, since there are several types of graphics card, make sure it is compatible and fit on your computer motherboard PCI slot before purchasing.

Screen menu and control related problems Screen control related problems happen as result of poor color balance, brightness, resolution, and adjustability setting. These computer monitor troubleshooting methods are easy to configure and correct using your system control panel.

Defective parts inside computer monitor

This is a problem related with the monitor itself. If all the other computer parts are working fine, but nothing is displaying, the monitor has either defective or burned part that should be replaced.

As I have already said, repairing and replacing the inside part is not easy. Before purchasing another one, take to maintenance shop and make sure that the part is easily replaceable and cheap.

Steps for dead or black screen

1. Ensure that the power cord and VGA cable is tightly fitted in at both ends and work properly. Power on, and see something comes on the screen.

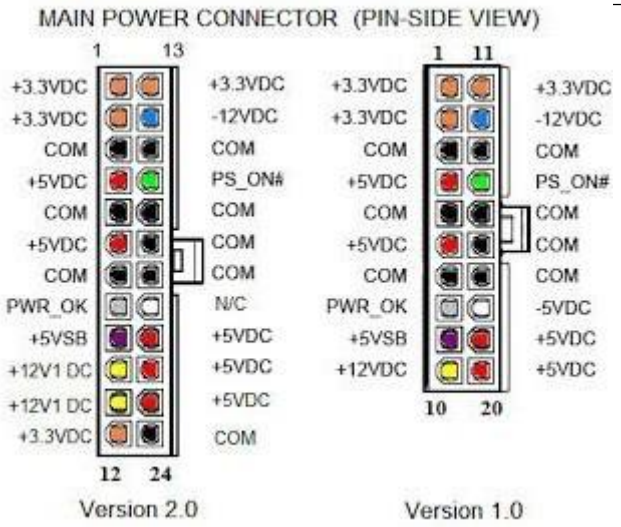
2. If you still get a black screen, unplug the monitor hook it on another computer. Power on the computer and check whether it works or not.

3. If nothing comes again, your monitor has some defective parts and should be repaired. If the monitor works on the second computer, probably the VGA card on the first computer is not working. You should buy a separate graphics card and install on PCI express slot.

If your graphics card was on the PCI slot, the problem could also be the slot. In this case moving the card to another slot may solve the problem.

SMPS (Switched Mode Power Supply)

SMPS - Switched Mode Power Supply. An SMPS is a Device to efficiently provide a regulated output voltage, from different level of the input voltage. SMPS transfers power from a source like the electrical power grid to a load (Eg: Computer).



SMPS Connection Diagram

The SMPS in a conventional desktop system is designed to convert the 220V AC, 50HZ into +5V, -5V, +12V and +3.3 V DC

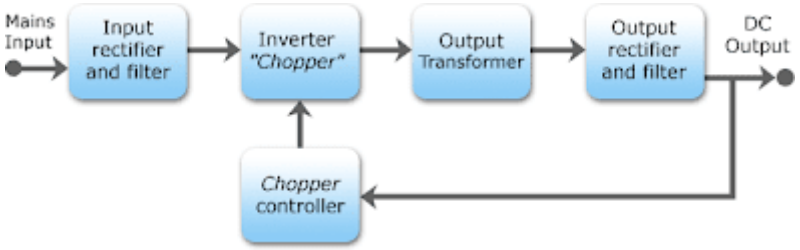


Diagram -SMPS

Troubleshooting SMPS

Problem 1: The Power doesn't come on.

Solutions

1. Check the Power from the wall socket
2. Check the Voltage Setting On the CPU
3. Check the Power switch of the Cabinet and Front Panel of Mother Board
4. Check the Power Supply Connections to the Motherboard
5. Check the SMPS without connecting it to the motherboard, you could see the two wire green and black which you have to short them (using any piece of wire/paper clip) in the 24 pin motherboard connector of the SMPS.

Problem 2: The PC Powers on after the second or third try

Solutions

1. Check the Power switch of the Cabinet
2. Replace SMPS (Get a Better Quality SMPS)

Problem 3: The PC Powers on but nothing happens after that (no beep)

Solutions

1. Remove the last hardware component installed and check again
2. Replace SMPS (Get a Better Quality SMPS)
3. Check the power cables to the Devices (Hard disk, DVD Drive etc)

Problem 4: The PC Powers on beeps and stops. NO Power OnSelf Test (POST) messages

Solutions

1. Check with another SMPS
2. This may be a Motherboard Problem.

Problem 5: The PC Powers on runs POST but there is no display

Solutions

1. Check the Monitor and the VGA Cable Connections
2. Check with another SMPS
3. This may be a Display Card Problem

Problem 6: There is a squealing/whistling/whining noise from SMPS when the PC starts

Solutions

1. Check the SMPS Fan
2. Component problem with SMPS. Replace it

Problem7: The PC freezes or reboots suddenly

Solutions

1. Check the SMPS Fan (May be overheating Problem)
2. Replace the SMPS

SUMMARY:

- **Trouble shooting** is a process in which certain measures and processes are collectively used to identify and fix problems which are encountered with in a NETWORK or any Hardware device.
- Trouble shooting can be both **manual**and **automatic**.
- Many tools such as PING, Traceroute, Nslookup, netstat, putty, pathping, etc which helps in trouble shooting network connection issues.
- **Mother Board** is the major component which handles entire communication between hardware components inside the PC.
- Trouble shooting a mother board is most difficult task than any.
- **FDD** is a old modelled storage device which are used in the past which can be used for both reading and writing.
- Some of the problems faced in FDD are **bad floppy diskette, Not setup in CMOS, not connected properly, bad drivers, bad hardware**.
- **HDD** is a Secondary storage device which stores data **permanently**.
- Basically there **are software solutions** for trouble shooting a HDD
- A **CD-ROM** is used to store data which can be read by a computer with an optical drive.
- Software Programs such as **CD Roller, CD Check, CD Diagnostic** helps to test the CD Drive
- A **printer** is an external hardware output devicethat takes the electronic data stored on a computer or other device and generates a hardcopy of it.
- Printer should be tested for both hardware and software
- Trouble shooting a MONITOR is a risky process which should not be done without knowledge.

Short Answer Type Questions:

1. What is Trouble Shooting?
2. What are the tools used in the network trouble shooting?
3. How to face the problem of The Power doesn't come on SMPS?
4. Explain different types of software solution to trouble shoot of HDD?
5. Explain the concept of ping?
6. What is Mother Board.
7. What is HDD?
8. What is printer?
9. Give the possible reasons for the printer not printing

Essay type Answer questions:

1. What is Trouble Shooting and write the processes of Trouble shooting.
2. Explain how will you trouble shoot when system is not functioning.
3. Explain the concept of various trouble shoots in Printer.
4. Explain the handling mechanism to handle troubles in Mother Board.
5. Explain the various trouble shoots in Modem and SMPs.
6. Explain the concept of various trouble shoots in HDD.
7. Explain the concept of various trouble shoots in FDD.
8. Explain the concept of various trouble shoots in CDROM.

COMPUTER SCIENCE & ENGINEERING

II YEAR

PART-B, VOCATIONAL COURSE

PAPER-ITHEORY

OOPS & JAVA

Periods/Week :04 **Periods/Year:110**

Time Schedule Weightage And Blue Print

S.No	Name of Unit	No.Of Periods	Weightage In Marks	Short Answer Questions	No.of Essay Questions
1	Object-Oriented Programming and Java ,	10	8	1	1
2	The Java programming language	25	8	1	1
3	Control Statements and Arrays	25	18	3	2
4	Implementing OOPS in Java	25	8	1	1
5	Packages and Interfaces	10	8	1	1
6	Exception Handling	5	2	1	1
7	Threads	5	8	1	1
8	Java Applets	5	8	1	---
	Total	**110**	**68**	**10**	**8**

COMPUTER SCIENCE & ENGINEERING

II YEAR

PART-B, VOCATIONAL COURSE

PAPER-II THEORY

RELATIONAL DATABASE & MANAGEMENT SYSTEM

Periods/Week : 04 **Periods/Year:110**

Time Schedule Weightage And Blue Print

S.No	Name of Unit	No.Of Periods	Weightage In Marks	Short Answer Questions	No.of Essay Questions
1	Concepts of DBMS	20	16	2	2
2	Entity and Relationship	20	16	2	2
3	Relational Model	20	16	2	2
4	SQL	30	10	2	1
5	Software Development Life Cycl	20	10	2	1
	Total	**110**	**68**	**10**	**8**

COMPUTER SCIENCE & ENGINEERING

II YEAR

PART-B, VOCATIONAL COURSE

PAPER-III THEORY

DATA COMMUNICATIONS AND COMPUTER NETWORKS

Periods/Week :04 **Periods/Year:110**

Time Schedule Weightage And Blue Print

S.No	Name of Unit	No.Of Periods	Weightage In Marks	Short Answer Questions	No.of Essay Questions
1	Data communications	10	8	1	1
2	Network Topologies	15	16	2	2
3	LAN components and Communication Hardware	20	8	1	1
4	Computer Networks .	20	10	2	1
5	Internet Connectivity and Services	20	16	2	2
6	Trouble shooting	25	10	2	1
	Total	**110**	**68**	**10**	**8**

COMPUTER SCIENCE & ENGINEERING

II YEAR

PAPER I OOPS & JAVA

Time:- 3 Hrs Max.Marks:- 50

SECTION-A

Note:- 1. Answer ALL Questions: **10 x 2=20**
Each Question carries 2Marks.

1. Define oops.
2. What is mixed mode?
3. What is the syntax of if statement?
4. What is goto statement?
5. Define an Array.
6. Define Polymorphism.
7. Define an interface.
8. What is an exception?
9. What is debugging?
10. Define an Applet.

SECTION - B **5 x 6 = 30**

Note:- 1. Answer ANY FIVE Questions:
Each Question carries 6 Marks.

11. Describe the main features of OOPS.
12. Describe the main data types in Java.
13. Explain the loop statements in java.
14. Write a program for matrix multiplication.
15. Explain Inheritance with an example.
16. Explain Packages in Java with an example.
17. Write about the types of errors?
18. What are the differences between Multitasking and multithreading.

COMPUTER SCIENCE & ENGINEERING

II YEAR

PAPER II RELATIONAL DATABASE & MANAGEMENT SYSTEM

TIME: 3 Hours **Max. Marks:50**

SECTION-A

Note:- 1. Answer ALL Questions: **10 x 2=20**
2. Each Question carries 2Marks.

1. Define DBMS.
2. Expand DDL and DML.
3. What is Entity and Relationship?
4. What are different types symbols used in E-R Diagram?
5. What are the different types of attributes in relational model?
6. What are the unary operations in relational model?
7. Write the structure of SQL.
8. What are basic operations in SQL?
9. Define a system.
10. What are the testing methods?

SECTION – B

5 x 6 = 30

Note:- 1. Answer any FIVE Questions:
Each Question carries 2Marks.

11. Explain the advantages of DBMS over File processing system.
12. Write the functions of Database Administrator.
13. Draw an ER diagram to show relation between a Student and Bank.
14. Explain basic structure of Relational Model.
15. Explain Binary operations in Relational data base.
16. Write about CODD rules.
17. Explain any five DDL commands with an example.
18. Explain various stages of System Development Life Cycle.

COMPUTER SCIENCE & ENGINEERING

II YEAR

PAPER III DATA COMMUNICATIONS AND COMPUTER NETWORKS

TIME:3 HoursMax. Marks:50

SECTION-A

Note:- 1. Answer ALL Questions: **10 x 2=20**
Each Question carries 2Marks.

1. Define Data Communication.
2. What is Computer Network?
3. What is Network Topology
4. What is HUB? Write types in it.
5. What are layers in OSI model?
6. Expand TCP/IP.
7. What is an E-Mail?
8. What is search Engine?
9. What is Trouble shooting?
10. Give the possible reasons for the printer not printing.

SECTION - B

5 x 6 = 30

Note:- 1. Answer ALL Questions:
Each Question carries 2Marks.

11. Explain basic of modes of Data Transmissions
12. Explain about LAN,WAN and MAN
13. Explain any three network topologies.
14. Explain about ROUTER and Bridge
15. Explain briefly the OSI model.
16. Explain any three web browsers.
17. Explain different types of firewalls.
18. Explain how will you trouble shoot when system is not functioning.