

**IVC Course Code : 319**

**COMPUTER SCIENCE & ENGINEERING  
SECOND YEAR**

**Intermediate Vocational Course**

**Paper I : OOPS & JAVA <sup>TM</sup>**

**NAME:** \_\_\_\_\_

**ROLL No.** \_\_\_\_\_

# **COMPUTER SCIENCE & ENGINEERING**

## **Paper – I**

### **OOPS & JAVA**

#### **INDEX**

<b>Unit-I</b>	<b>Object Oriented Programming And Java</b>	<b>1-11</b>
<b>Unit-II</b>	<b>The Java Programming Language</b>	<b>12-26</b>
<b>Unit-III</b>	<b>Control Statement And Array</b>	<b>27-49</b>
<b>Unit-IV</b>	<b>Implementing oops in Java, Java Methods</b>	<b>50-62</b>
<b>Unit-V</b>	<b>Packages and Interfaces</b>	<b>63-69</b>
<b>Unit-VI</b>	<b>Exception Handling</b>	<b>70-78</b>
<b>Unit-VII</b>	<b>Threads</b>	<b>79-85</b>
<b>Unit-VIII</b>	<b>The Java Applet</b>	<b>86-97</b>
	<b>QUESTION BANK</b>	<b>98-105</b>

# OOPS & JAVA

## UNIT -I

### అబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ & జావా

### Object Oriented Programming and Java

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Oops అనగా నేమి? Define Oops.

జ. Oops: Oops అనగా object oriented programming system అని అర్థము. ఇది క్రొత్త ప్రోగ్రామింగ్ పద్ధతి. Oops ప్రోగ్రామింగ్ లో చాలా అబ్జెక్ట్ లను ఉపయోగిస్తారు. Oops లో అబ్జెక్ట్ ల మధ్య డేటా మరియు ప్రోగ్రామ్ లను, సమాచారాన్ని బదిలీ చేసే అవకాశం కూడా కలదు.

- Object
- Class
- Polymorphism
- Inheritance
- Abstraction
- Encapsulation.

#### 2. జావాను ఎవరు ఎక్కడ అభివృద్ధి చేశారు ? Who invented by Java?

జ. జావాను Sun Micro Systemలోని James Gosling అభివృద్ధి చేశారు. జావాను మొదట 'Oak' అని పిలిచేవారు. దీనిని Electronics లోని Embedded Applicationల కొరకు ఉపయోగించేవారు. దీనిని 1991 లో అభివృద్ధి పరచి జావాను తయారు చేసేవారు.

#### 3. అబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రాం వల్ల ఉపయోగాలు రాయండి. Need of Object Oriented Program?

జ. అబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రాం(Oop) వాళ్ళ ఉపయోగాలు:

Oops లో వ్రాసిన ప్రోగ్రంల ద్వారా modularityని పెంచవచ్చును.

- Oops అనునది Real Time Objectsకు దగ్గరగా ఉంటుంది.
- Oops లో డేటాను పబ్లిక్ మరియు ప్రైవేటుగా విభజించే అవకాశం ఉంది.  
దీనివల్ల ప్రోగ్రంలోని డేటాకు సెక్యూరిటీని పెంచవచ్చును.
- Oops లో Encapsulation, Inheritance, Polymorphism వంటి ఉపయోగకరమైన లక్షణాలు ఉన్నాయి.

#### 4. Class అనగానేమి ? What is Class?

జ. Class:

- Class అనునది Variable, methods మరియు objects ల యొక్క కలయిక. Class అనునది Objectకు blue print వంటింది.

```
ఉదా: class Test
{
    int count ;
    public void display ()
    {
        count++;
    }
}
```

#### 5. Object అనగానేమి ? What is an Object?

జ. Object:

- Class యొక్క instance నే object అంటారు. ఈ object లను వాటి పేర్లతో సూచిస్తారు.
- Object అనునది data మరియు డేటా మీద పని చేసే పంక్షన్ (method) లను కలపి ఒక యూనిట్ గా తయారు చేస్తారు. ప్రతి అబ్జెక్ట్ ఒక ప్రత్యేకమైన classకి చెంది ఉంటుంది.

ఉదా: Public class test

```
{
    Public static void main (string urgs [])
    {
        Demo d1 =new demo ( );
    }
}
పై ఉదాహరణలో d1 అనునది ఒక అబ్జెక్ట్ demo అనునది క్లాస్.
```

#### 6. డేటా అబ్స్ట్రాక్షన్ అంటే ఏమిటి ? What is Data Abstraction?

జ. డేటా నైరూప్యత(Data Abstraction):

- సేకరించిన data లో అవసరమైన data ను మాత్రమే క్రమబద్ధీకరించి సరి అయిన క్రమములో అమర్చడాన్ని Data abstraction అని అంటారు.

## 7. JVM,JRE,JIT,OOP అను విస్తరించండి?

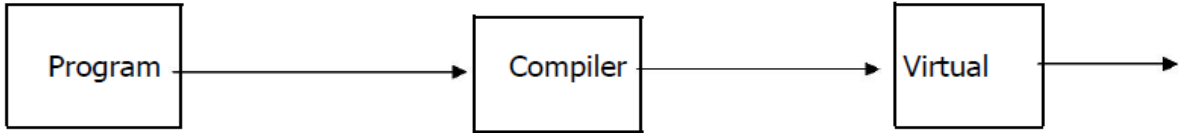
Expand the terms JVM, JRE, JIT, OOP.

- JVM: జావా వర్చువల్ మెషిన్ (Java Virtual Machine)
- JRE: జావా రన్ టైం ఎన్విరాన్మెంట్ (Java Runtime Environment)
- JIT: జస్ట్ ఇన్ టైం (Just In Time)
- OOP: ఆబ్జెక్ట్ ఓరియెంటెడ్ ప్రోగ్రామ్ (object oriented programming System)

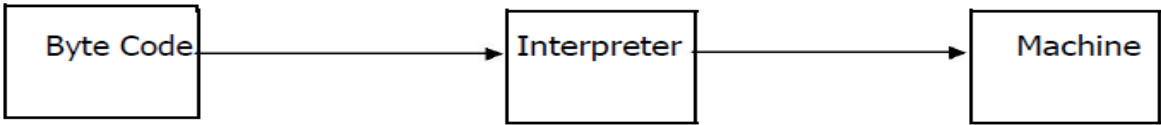
## 8. JVM అనగానేమి? What is JVM?

జ. JVM:

- జావా వర్చువల్ మెషిన్ (Java Virtual Machine) అనేది java byte code ను అమలు చేయగల వర్చువల్ మెషిన్. ఇది java software ప్లాట్ ఫారం యొక్క కోడ్ ఎగ్జిక్యూషన్ కాంపోనెంట్ (Component).
- జావా బైట్ కోడ్ మరియు తర్జుమా చేసుకోవటానికి జావా వర్చువల్ యంత్రం ఉపయోగిస్తారు.



Source Program:



## 9. JRE అనగానేమి? What is JRE?

జ. JRE:

- జావా రన్ టైం ఎన్విరాన్మెంట్ (Java Runtime Environment) ను జావా రన్ టైమ్ అని కూడా అంటారు. ఇది జావా డెవలప్మెంట్ కిట్ (JDK) లో ఒక భాగం.
- జావా అప్లికేషన్ ని అమలు చేయడం కొరకు జావా రన్ టైం ఎన్విరాన్మెంట్ కనీస ఆవశ్యకతలను అందిస్తుంది.

## 10. JIT అనగానేమి? What is JIT?

జ. JIT:

- JIT అనగా Just In Time. Java byte code ను compile చేసేటప్పుడు processing speed ను మెరుగుపరచడానికి ఉపయోగపడుతుంది.

## 11. బైట్ కోడ్ అనగానేమి? What is Java Byte code?

జ. Java Byte Code:

- Java లో మనము తయారు చేసే ఆదేశాలను byte code అని అంటారు. సాధారణముగా ఒక byte code ఒక byte పొడవు కలిగి ఉంటుంది.
- కొన్ని ఆదేశాలు అంతకన్నా ఎక్కువ పొడవు కలిగి ఉంటుంది. జావా బైట్ కోడ్ అనేది JVM ఎగ్జిక్యూట్ యొక్క సూచనల రూపం.

## 12. మెషిన్ న్యూట్రల్ అంటే ఏమిటి?

What is Java Machine neutral or Architectural Neutral?

జ. Java Machine neutral:

- Java లో మనము రాసే ఆదేశాలను source code అని అంటారు. ఈ ఆదేశాలను JIT compiler Java byte code గా మారుస్తుంది.
- ఈ byte code ఏ system లో అయినా execute చేయగలము. ఈ ప్రక్రియను Machine neutral అని అంటారు.

దీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

## 1. Oops యొక్క భావనలు వివరించండి? Explain OOPS Concept.

జ. Oops యొక్క భావనలు:

Oops అనగా object oriented programming System అని అర్థము. Java language ద్వారా OOPS Concept లను అభివృద్ధి చేయవచ్చును. Java programming language ఈ క్రింది భావనలను సపోర్టు చేస్తుంది.

### 1. Object:

- Object అనగా ఒక వ్యక్తి, వస్తువు లేదా ఒక entity గా సూచిస్తారు.
- Object అనునది physical మరియు logical గాను పని చేస్తుంది.
- Object అనునది memory లో ఒక భాగమునకు address గా గుర్తిస్తారు.

## 2. Class:

- కొన్ని object ల సమూహాన్ని class అని అంటారు.
- Class అనునది logical entity.
- Class అనునది Variable, methods మరియు objects ల యొక్క కలయిక.
- Class అనునది objectకు blue print వంటిది.

## 3. Inheritance:

- ఒక object తన లక్షణాలను వారసత్వరూపములో తీసుకొంటే దాన్ని inheritance అని అంటారు.
- ఈ object నుండి కొత్త objects ను create చేయవచ్చు. దీని ద్వారా కొత్త objectలోని Properties అన్ని ప్రతీసారి Create చేయనవసరంలేదు.

## 4. Polymorphism:

- ఒక పనిని అనేక రకాలుగా చేయడాన్ని Polymorphism అని కూడా అంటారు.
- Polymorphism అనగా “Many Forms”.
- దీనిని function overloading, Operator overloading మరియు Dynamic Binding ద్వారా అభివృద్ధి చేయవచ్చును.

## 5. Data abstraction:

- దీని ద్వారా సేకరించిన data లో అవసరమైన data ను మాత్రమే తీసుకొని క్రమబద్ధీకరించి అవసరమైన రీతిలో అమర్చడాన్ని Data abstraction అని అంటారు. అవసరము లేని data ను కనబడ నీయకుండా చేస్తుంది.

## 6. Data Encapsulation:

- Encapsulation అనగా కలపడము (Combining) అని అర్థము.
- ఒక procedure లో ఉన్న object దాని యొక్క లక్షణాలబట్టి వేరొక method లేదా function తో కలిపితే ఒక binding form అవుతుంది. ఈ ప్రక్రియనే Encapsulation అని అంటారు.

2. జావా ప్రోగ్రాం యొక్క లక్షణాలను వివరించండి? Explain the features of Java? (Or)

Write about the main features of Java.

జ. జావా చాలా Advanced లక్షణాలు కలవు. అవి :

**1. Simple, Small & Familiar:**

- జావా అనునది చాలా తక్కువ memory ని కలిగి ఉంటుంది. దీనిలో ప్రోగ్రాంలను తేలికగా వ్రాయవచ్చు. 'C' మరియు 'C++' తెలిస్తే ప్రోగ్రాం కోడ్ లను తేలికగా వ్రాయవచ్చును. అందుకే జావాను "Simplified Version of C++" అని అంటారు.

**2. Object oriented:**

- జావా అనునది ఒక నిజమైన Object Oriented లాంగ్వేజి.
- దీని ద్వారా Object Modelను తేలిక గా అభివృద్ధి పరచవచ్చును. మరియు Extend(పోడిగించవచ్చును) చేయవచ్చును.

**3. Robust and Secure:**

- జావా ప్రోగ్రాం లను user అనుమతి లేకుండా ఉపయోగించకుండా అభివృద్ధి చేసుకొనవచ్చును.
- దీని ద్వారా High level సెక్యూరిటీ ని ప్రోగ్రాం లకు provide చేయవచ్చును.

**4. Distributed:**

- జావా ద్వారా ప్రోగ్రాంలను మరియు డేటాను కూడా ఒక ప్రదేశం నుండి మరొక ప్రదేశాలకు పంపించగలము.
- Networking లో ఉన్న కంప్యూటర్ల మధ్య డేటా మరియు ప్రోగ్రాం లను షేర్ చేయడం ద్వారా Remote Access రిమోట్ యాక్సెస్ ను కూడా చేయగలము.
- Java లో data మరియు programs అన్ని share చేసుకొనే అవకాశం ఉంటుంది.
- ఇది అన్ని రకముల operating systems లోను పని చేస్తాయి మరియు అన్ని Networks లోను ఉపయోగించవచ్చును.



## 5. Platform Independent and Portable:

- Java ప్రోగ్రాంను చాలా తేలికగా ఒక computer నుండి మరియొక computer కు ఎప్పుడైనా ఎక్కడికి అయినా (Anywhere and Anytime) మార్చవచ్చును.
- ఒకవేళ ఆపరేటింగ్ సిస్టమ్, ప్రాసెసర్, system యొక్క ఇతర భాగాలూ మార్పు చెందినా జావా ప్రోగ్రాం లో మార్పులు చేయనవసరంలేదు.
- ఈ కారణంగానే java ప్రోగ్రామింగ్ లాంగ్వేజిని ఇంటర్నెట్ ఆధారిత ప్రోగ్రాం లను అభివృద్ధి చేయడంలో విశేషంగా ఉపయోగిస్తారు.

## 6. Multi-threaded and Interactive:

- Multi threaded అనగా ఒకే సమయంలో ఒకటి కన్నా ఎక్కువ ప్రోగ్రాంలను (టాస్క్ లను) ఎగ్జిక్యూట్ చేయటం. Java multi threaded ను కలిగి ఉంటుంది.

## 7. Dynamic Binding:

- వివిధ classes ను, methods ను, libraries, మరియు objects ను link ఏర్పాటు చేయుటకు Binding అని అంటారు.
- జావా అనునది ఒక డైనమిక్ లాంగ్వేజి. దీనిలో C మరియు C++ లలో వ్రాసిన function లను కూడా ఉపయోగించుకొనవచ్చును.

## 8. Compile and Interpreted:

- Java లో మనము తయారు చేసే Byte code ను Machine code instructions గా మార్చి అవి ఏ machine లోను store అయి ఉండవు.

## 9. High Performance:

- ప్రోగ్రామ్ ను రన్ చేసేటప్పుడు ఎటువంటి ఇబ్బందులు లేకుండా జావా ప్రోగ్రాంలను అభివృద్ధి చేయడం జరిగింది. దీని ద్వారా జావా ప్రోగ్రాం ల యొక్క Performance కూడా బాగుంటుంది.

### 3. జావా ప్రోగ్రాం యొక్క స్ట్రక్చర్స్ వ్రాయుము? Explain the Java program, Structure?

#### జ. జావా ప్రోగ్రాం యొక్క స్ట్రక్చర్స్ (Java Programming Structure):

Java program అనునది అనేక classes కలిగి ఉంటుంది. అందులో ఒక్క class main method గా operate చేస్తుంది. మిగిలిన classes అన్నీ దానికి link అయి ఉంటాయి. Java program కొరకు

```
Document Section
Package Statements
Import Statements
Interface Statements
Class definitions
Main Method Class
{   Statements
}
```

#### 1. Documentation Section:

- Documentation Section program పేరు, author పేరు మొదలగు details ఉంటాయి .javaలో comment `/**---*/` ల మధ్య వ్రాస్తారు .

#### 2. Package Statement:

- Program లో రాసే మొదటి line ను package అని అంటారు.
- ఇందులో package name కలిగి ఉంటుంది.
- ఈ పేరు ఆధారంగా compiler సంబంధిత class ఒక చోటుకు చేర్చుతుంది.

Ex: Package Employee;

#### 3. Import statement:

- Java Libraries తో connectivity కొరకు ఈ statementను రాస్తాము.
- ఇది 'C' లంగ్వేజిలో include statement వలె పని చేస్తుంది.

Ex: import student test;

#### 4. Interface statement:

- ఇది కూడా Class వలె ఉంటుంది. కానీ ఇది కొన్ని methods యొక్క కలయిక. దీని ద్వారా Multiple inheritance ను అభివృద్ధి చేయవచ్చును.

## 5. Class Definition:

- Java program multiple Class లను సపోర్ట్ చేస్తుంది.
- Class అనునది జావా ప్రోగ్రాంలో తప్పనిసారిగా ఉండవలసిన element (ఎలిమెంట్స్).
- ఎన్ని Class లతో ప్రోగ్రాం చేయాలి అన్నది మనం అభివృద్ధి చేసే ప్రోగ్రాం మీద ఆధారపడి ఉంటుంది.

## 6. Main method class:

- Java లో main method class అనునది తప్పనిసరిగా ఉండవలెను.
- ప్రోగ్రాం యొక్క ఎగ్జిక్యూషన్ ను main method నుంచే ప్రారంభిస్తుంది. simple java program లో main method ఒకటే ఉంటుంది. main method లోని చివరి end statement కి చేరితే program ఎగ్జిక్యూషన్ పూర్తి అయింది అని అర్థం.

## 4. Java Program ను implement చేయు విధానమును తెలుపండి

### Explain implementation a java program

జ. Java లో program implementation అనేది మూడు భాగాలుగా జరుగుతుంది.

1. Create the Program (Program తయారీ)
2. Compile the Program (Program లో తప్పులు సరిచేయుట)
3. Running the Program (Program result తెలుసుకొనుట)

### 1. Create the Program:

Note pad లో మనము java program ను తయారు చేస్తాము.

Class sample

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

} type చేయడం అయిన తరువాత file ని save చేయాలి. File name sample.java గా save చేయాలి. Save చేసేటప్పుడు class name, file name ఒకటే ఉండాలి.

### 2. Compiling the Program:

- తయారు చేసిన ప్రోగ్రాంను compile చేయుటకు javac అనే command ద్వారా compile ను source code ను byte code గా మరుస్తాము.

- Javac sample.java ప్రోగ్రాంలో తప్పులు లేనట్లైతే compiler automatic గా source file name తో byte code name ని అందిస్తుంది extension class ఇస్తుంది. Sample.class.

### 3. Running the Program:

- Program ను run చేయడానికి java interpreter ను ఉపయోగిస్తారు.
- Program ను execute చేయటంకై java file name లేదా test అని type చేయవలెను.

### 5. Java కు వాడే వెబ్ బ్రౌజర్ల గురించి వివరించండి?

Explain the web browsers used to java.

జ. Web browsers used to Java:

- వెబ్ బ్రౌజర్ అనేది ఇంటర్నెట్ లోని వెబ్ పేజీలను వీక్షించడానికి వ్యక్తులకు సహాయపడే ఒక software అనువర్తనం. ఒక బ్రౌజర్ వెబ్ సైట్ లను సందర్శించడానికి మరియు లాగిన్ చేయడం.
- ఒక సైట్ నుంచి మరోదానికి లింక్ చేయడం, మల్టీమీడియా, మరో పేజీనుండి మరో పేజీని సందర్శించడం.
- Print చేయడం మరియు ఈ-మెయిల్స్ అందుకోవడం మరియు మరెన్నో ఇతర కార్యకలాపాలను చేయడానికి మిమ్మల్ని వీలుచేస్తుంది.
- ఉదాహరణకు ప్రజాదరణ పొందిన వెబ్ బ్రౌజర్లు గూగుల్ క్రోమ్, మొజిల్లా ఫైర్ ఫాక్స్, మైక్రోసాఫ్ట్ ఇంటర్నెట్ ఎక్స్ప్లోరర్, ఆపిల్ సఫారి మరియు ఒపేరా బ్రౌజర్.

➤ Internet Explorer

➤ Google Chrome

➤ Mozilla fire fox

➤ Safari

➤ Opera

### 1. Internet Explorer:

- Software దిగ్గజం మైక్రోసాఫ్ట్ నుండి అత్యంత ప్రజాదరణ పొందిన ఉత్పత్తి లో Internet Explorer(IE) ఒకటి. ఇది 1995లో ప్రవేశ పెట్టబడింది మరియు 1998 లో network పాపులారిటీని క్రాస్ చేసింది.

## 2. Google Chrome:

- గూగుల్ క్రోమ్ ను గూగుల్ అభివృద్ధి చేయగా దాని బీటా వెర్షన్ మైక్రోసాఫ్ట్ విండోస్ ఆపరేటింగ్ system కోసం 2, సెప్టెంబర్, 2008 న విడుదల అయింది.

## 3. Mozilla fire fox:

- మొజిల్లా ఫైర్ ఫాక్స్ భద్రత, సరళీకరణ, పోడిగించబడడం ఇంకా ఆవశ్యకత మరియు శక్తి కోసం రుపొందించిన వెబ్ బ్రౌజర్. ఇది లినక్స్ మరియు ఆండ్రాయిడ్ రెండిటికి అందుబాటులో ఉంది.

## 4. Safari:

- సఫారీ అనేది ఆపిల్ ఇంక్ అభివృద్ధిచేసిన వెబ్ బ్రౌజర్లో ఒకటి. ఇది టాబ్స్ బ్రౌజింగ్, పాప్ అప్ బ్లాకింగ్ మరియు బిల్డ్ ఇన్ సెర్చ్ ఫంక్షనాలిటీ వంటి ఆధునిక వెబ్ బ్రౌజర్ లో కొత్త ఫీచర్లతోపాటు బుక్ మార్క్ లను ఉపయోగించడం సులభం.

## 5. Opera:

- ఒపెరా బ్రౌజర్ లో లభ్యమయ్యే కొన్ని ఫీచర్ లు వేగవంతమయిన, యూజర్ ఫ్రెండ్లీ, multiple windows, జూమ్ ఫంక్షన్, ఇంకా మరెన్నో జావా, నాన్ జావా enable వెర్షన్ లు కూడా అందుబాటులో ఉన్నాయి.

## 6. Java లో JDK installation procedure ని తెలుపండి.

How to install JDK software and set java home on window system.

జ. Java లో JDK installation procedure: Java లో JDK installation చేయుటకై ముందుగా <http://java.sun.com/javase/download/index.jsp> అని type చేసి enter press చేయవలెను లేదా తగిన software link ను చూసుకొని download పై click చేయవలెను. Downloading press పూర్తి అయిన తరువాత java home ని తయారు చేయుటకై

- Mycomputer icon పై right click చేయుము.
- Mycomputer Properties pulldown menu వస్తుంది.
- అందులో advance tab లో environment variables select చేసి edit java home అని press చేసి java home point పై click చేయవలెను.
- Java software ఉన్న ప్రదేశములో develop అవుతుంది.

C:\programfiles/java/jdk1.6.0 గా ఉపయోగించడానికి JDK ఉంటుంది.

## UNIT -II

జావా ప్రోగ్రామింగ్ లాంగ్వేజ్

### THE JAVA PROGRAMMING LANGUAGE

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. వేరియబుల్ ను నిర్వచించండి ? Define Variable.

జ. Variable:

- ఒక డేటా విలువను నిల్వ చేయడానికి వేరియబుల్ అనే డేటా నేమ్ ఉపయోగపడుతుంది. కార్యక్రమాన్ని అమలు చేసే సమయంలో విలువలను మార్చడాన్ని వేరియబుల్ అంటారు.
- అమలు సమయంలో విభిన్న సమయాల్లో వేరియబుల్ విభిన్న విలువలను కలిగి ఉండవచ్చు.

ఉదా:- Avg(సగటు),  
Ht(ఎత్తు),  
Tot(మొత్తం),  
Tot \_ stu(మొత్తం విద్యార్థులు).

#### 2. స్థిరాంకాలు నిర్వచించండి మరియు రకాలను తెల్పండి?

Define Constants. Type of Constants.

జ. Constants:

- స్థిర విలువలు అనేవి ప్రోగ్రామ్ యొక్క అమలు సమయంలో మారవు java అనేక రకాలైన స్థిరాంకాలను సపోర్ట్ చేస్తుంది. అవి:-
  - అంగ స్థిరాంకాలు(Integer)
  - రియల్ స్థిరాంకాలు(Real)
  - సింగిల్ కారెక్టర్ స్థిరాంకాలు (Single character)
  - బ్యాక్ స్లాష్ కారెక్టర్ స్థిరాంకాలు (Backslash character).

#### 3. లిటరల్స్ అనగానేమి? What are Literals?

జ. Literals:

- జావాలోని లిటరల్స్ అక్షరాల యొక్క క్రమం (అంకెలు ,అక్షరాలు మరియు ఇతర అక్షరాలు) వేరియబుల్స్ లో నిల్వ చేయడానికి స్థిరమైన విలువలను సూచించే ఒక క్రమము. అవి:-

- అంగ లిటరల్స్ (Integer)
- రియల్ లిటరల్స్ (Real)
- సింగిల్ కారెక్టర్ లిటరల్స్ (single character)
- బ్యాక్ స్లాష్ కారెక్టర్ లిటరల్స్ (Backslash character).

#### 4. Identifiers అంటే ఏమిటి ? What are Identifiers?

##### జ. Identifiers:

- ఐడెంటిఫైయర్స్ ఆనగా ఇవి ప్రోగ్రామర్- డిజైన్ చెయ్యబడిన టోకెన్లు.
- ఒక కార్యక్రమంలో క్లాసులు, మెథడ్స్, వేరియబుల్, అబ్జెక్ట్ లు, లేబుల్స్, ప్యాకేజీలు మరియు ఇంటర్ఫేస్ ల కొరకు వీటిని ఉపయోగిస్తారు.

#### 5. మిశ్రమ రీతి అంటే ఏమిటి ? What is Mixed Mode?

##### జ. Mixed Mode:

- మనము వివిధ రకాల operations చేసేటప్పుడు ఒకే చోట వివిధ రకముల operate ఉపయోగించేటప్పుడు brackets తో ఉన్నవైటిని ముందుగా solve చేసే వాటిని కలుపుటము. ఈ ప్రక్రియయాని mixed mode అని అంటారు.

ఉదా:-int 12.0/5 ఫలితాన్ని ఇస్తుంది 2. ఇక్కడ 12.0 అనేది ఫ్లోట్ ఓపర్ అండ్ 5 అనేది పూర్ణాంకం.

#### 6. కాస్టింగ్ వాల్యూస్ (Casting values) అంటే ఏమిటి? What is Casting values?

##### జ. Casting values:

- ఒక డేటా టైప్ విలువను మరో డేటా టైప్ గా మార్చే ప్రక్రియను కాస్టింగ్ అంటారు. ఇది రెండు విధాలుగా జరుగుతుంది. అవి:-

✓ Implicit type casting

✓ Explicit type casting.

#### 7. యాక్సెస్ నిర్దేశిస్తుంది (Access Specifier) ఏమిటి? What is Access Specifier?

##### జ. Specifier:

- మనము తయారు చేసే class మరియు class members ఒక keyword ద్వారా java users అందరికీ అందుబాటులో ఉండాలా, వద్దా చెప్పవచ్చు. Access specifiers రూపంలో ఉంటుంది.

- పబ్లిక్ (Public)
- రక్షిత (Protect)
- ప్రైవేటు (Private)
- ప్రవేట్ ప్రొటెక్ట్ (Private protect)
- డిఫాల్ట్ (Default)

## 8. నిర్మాణ దారుడు (Constructor) అంటే ఏమిటి ? What is Constructor?

### జ. Constructor:

- అబ్జెక్ట్ క్రియేషన్ సమయంలో instances వేరియబుల్ ప్రారంభించడం కొరకు ఇది ప్రత్యేక విధానం ఉపయోగించబడుతుంది. దీనిలో ఒకే రకమైన గుణాలు ఉంటాయి.
- నిర్మాణ దారు పేరు మరియు తరగతి పేరు ఒకేలా ఉండాలి మరియు దానిపేరు సాధారణ బ్రాన్ జతతో ముగుస్తుంది. ఒక కన్స్ట్రక్టర్ కు పరామితులు ఉండవచ్చు లేదా ఉండకపోవచ్చు.

ఉదా:- Student S1 = new student ();

Student S2 = new student ("Murthy", "Lecturer");

## 9. నిర్వచన పద్ధతి (Method) అంటే ఏమిటి? Define Method ?

### జ. Method:

- ఒక పని జరుగుటకు అవసరమైన ఆదేశాల సమూహమును Method అని అంటారు. Method అనేది ఒక class లో ఉన్న objects మాత్రమే పరిమితము కాదు. వివిధ class లో ఉన్న objects కూడా వర్తిస్తాయి.

ఉదా:- Public static void main (string [] args)

```
{
    Statement ;
    Method 1 ()
        Statement ;
    Method 2 ()
        Statement;
}
```



## 10. తరగతి(Class ) నిర్వచించండి? Define Class.

### జ. Class:

- Variables, objects మరియు method యొక్క కలయికనే class అని అంటారు.
- Class అనునది object blue print వంటిది మరియు లాజికల్ కూడను.
- దీనినుంచి మీరు ఒక ఆబ్జెక్ట్ క్లాస్ ని సృష్టించవచ్చు మరియు ఏ స్థలం కూడా ఉపయోగించదు.

➤ Data member

➤ Method

➤ Constructor

➤ Block

**Syntax :** Class Name [ Extends Super class name ]

{

Declarations of members

Method declarations

}

## 11.ఆబ్జెక్ట్ సు (Object) నిర్వచించండి? Define Object.

### జ. Object:

- స్థితి, ప్రవర్తన ఉన్న అస్తిత్వాన్ని ఒక వస్తువుగా పేర్కొంటారు.
- ఉదాహరణకు ఒక కుర్చీ, పెన్ను, టేబుల్, కీబోర్డ్, బైక్ మొదలైనవి భౌతికంగా లేదా తర్కికంగా ఉండవచ్చు. ఒక వస్తువును ఒక వర్గానికి పర్యాయంగా నిర్వచించవచ్చు. ఆబ్జెక్ట్ అనేది మూడురకాలుగా లక్షణాలు కలిగి ఉంటాయి. అవి:

➤ **State :** Represents the data of an object

➤ **Behaviour :** Represents the behaviour of an object

➤ **Identity :** Identity is typically implemented via a unique ID

## 12. Void నిర్వచించండి. Define Void.

జ. Void:

- శూన్యత కీ పదం ఒక విలువను తిరిగి లేని పద్ధతులను సృష్టించడానికి అనుమతిస్తుంది.
- Void key అనేది return value ఇవ్వని methods తయారు చేయుటకు ఉపయోగించే ఒక keyword. Void లో తయారు చేసే java statements అన్ని semicolon (;) తో end చేయాలి.

## 13. Instance variable అనగా నేమి? What is Instance variable?

జ. Instance variable:

- Java లో నిల్వ చేసే object యొక్క లక్షణాలను తెలిపే దానిని instance variable అంటారు. వీటిని static అనే keyword తో తయారు చేస్తారు.

## 14. This ఉపయోగాన్ని తెలుపండి. What is This method?

జ. This:

- This అనే keyword, class లో object ఉపయోగించే ప్రదేశాన్ని తెలుపుతుంది. ఒక object కు అదే class లో వేరే చోట ఉపయోగించే object ను సూచించడానికి ఉపయోగపడుతుంది.

దీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

### 1. జావాలో యాక్సెస్ క్లాస్ విధానం గురించి వివరించండి?

Explain Accessing method of a java class.

జ. జావాలో యాక్సెస్ క్లాస్ విధానం (Access class method in Java):

- జావా లో class బయట ఉన్న objects, variables మరియు methods ని వెలుపల నుంచి యాక్సెస్ చేసుకోవడానికి అవకాశం లేదు.
- మనం నేరుగా object యొక్క గుణాలను ప్రాప్తి చేయలేము అని అర్థం. దీనిని చెయ్యడం కొరకు, మనం విడిగా సంబంధిత ఆబ్జెక్ట్ మరియు డాట్ ఆపరేటర్ (.Dot) ను ఉపయోగించాలి.

ఉదా:- వస్తువు పేరు. అస్థిర నామము, విలువలు; వస్తువు పేరు పద్ధతిపేరు(పరామితి జాబితా);

ఉదా:- Rectangle re1 = new rectangle();

Rectangle re2 = new rectangle ();

Re1. Length = 20;

```
Re1. Width = 30;
```

```
Re2. Length = 25;
```

```
Re2.width = 40;
```

పైన ఉదాహరణకు విలువలను కేటాయించడం ఒక మార్గం.

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

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. That is :

```
Rectangle re1 = new Rectangle();
```

```
Re1.getdata (20,30)
```

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.out.println("area2="; area2);
}
}

```

2. జావాలో యాక్సెస్ స్పెసిఫైయర్ (Access Specifier) విధానం గురించి వివరించండి?

**Explain the Access Specifier in Java?**

జ. జావాలో యాక్సెస్ స్పెసిఫైయర్ విధానం (Access Specifier in Java):

- ఒక యాక్సెస్ స్పెసిఫైయర్ అనేది ఒక క్లాస్ లేదా క్లాసు యొక్క సభ్యులను యాక్సెస్ చేసుకునే ఒక కీలక పదం.
- ఒక క్లాసు మరియు దానియొక్క సభ్యులముందు మనం యాక్సెస్ స్పేషల్ లను ఉపయోగించవచ్చు. జావాలో నాలుగు యాక్సెస్ స్పెసిఫయర్లు లభ్యం అవుతున్నాయి.
  - ప్రైవేటు (private)
  - పబ్లిక్ (Public)
  - రక్షిత (Protected)
  - ప్రైవేట్ ప్రొటెక్ట్డ్ (Private Protected)
  - డిఫాల్ట్ (Default)

**ప్రైవేటు (private):**

- Private access ఉన్న class member ఆ class లో మాత్రమే ఉపయోగపడతాయి.
- Class members, class బయట లేదా వేరే class లేను ఉపయోగపడవు.

**పబ్లిక్ (Public):**

- Public లో define చేసిన class members ఏ class లో నైనా use అవుతాయి.
- Package లో ఉన్న అన్నీ class కి read మరియు use చేయగలము.

### రక్షిత (Protected):

- Protected members defined class మరియు outside class లో కూడా ఉపయోగపడతాయి కాని ఆ package వరకు మాత్రమే ఉపయోగపడతాయి.
- Derived class లో class member access ఉండదు.

### ప్రైవేట్ ప్రొటెక్ట్ (Private Protected):

- ఇందులో define చేసిన fields వేరే classes లో కనిపిస్తాయి. గాని ఉపయోగపడవు.
- Define చేసిన class లో మాత్రమే ఉపయోగించగలము.

### డిఫాల్ట్ (Default):

- No access specifier programmer వ్రాస్తే అప్పుడు జావా compiler default access ను ఉపయోగిస్తుంది.
- Default members class బయట పని చేస్తుంది , అలాగే same package ను ఉపయోగించుకుంటుంది.

### 3. జావాలో ఉండే ఆపరేటర్స్ ను (Operators in Java) వివరించండి?

Explain the Operators in Java?

జ. జావాలో ఉండే ఆపరేటర్స్ (Operators in Java):

- ✓ Arithmetic Operators
- ✓ Relational Operators
- ✓ Logical operators
- ✓ Bit wise operators
- ✓ Assignment operators
- ✓ Increment & Decrement operators
- ✓ Conditional operators

#### 1. Arithmetic Operators:

- Mathematical calculations కోసము ఈ క్రింది విధముగా table మరియు example రూపంలో ఇవ్వడం జరిగింది.

Operators	Meaning	Example	Result
+	Addition	5+8	13
-	Subtraction	8-5	3
*	Multiplication	8*5	40
/	Division	80/5	16
%	Modules	5%2	01

## 2. Relational Operators:

- రెండు operand ల మధ్య సంబంధాన్ని పోల్చి చెప్పే operator ను Relational operator అని అంటారు. ఈ క్రింది విధముగా table రూపంలో ఇవ్వడం జరిగింది.

OPERATOR	PURPOSE
=	It is equal to
!=	It is not equal to
<	It is less than
>	Greater than
>=	Greater than equal to
<=	Less than equal to

These returns result in the form of 'true' or 'false'.

- ### 3. Logical operators: రెండు expressions మధ్య సంబంధాన్ని తెలియచెప్పేవి logical operator. ఈ క్రింది విధముగా table రూపంలో ఇవ్వడం జరిగింది.

OPERATOR	PURPOSE
AND	&&
OR	
NOT	!

## 4. Bit wise operators :-

- Binary format ఆధారముగా variable యొక్క content ని మార్చడానికి ఉపయోగపడే operator ను Bitwise మరియు Bit shift అని అంటారు.

Symbol	Name of the Operator	Example
~	Unary bitwise complement	~op2
&	Bitwise AND	op1 & op2
	Bitwise inclusive OR	op1   op2
^	Bitwise exclusive OR	op1 ^ op2
<<	Signed left shift	op1 << op2
>>	Signed right shift	op1 >> op2
>>>	Unsigned right shift	op1 >>> op2

#### 5. Assignment operators :-

- Operand లో Operator ద్వారా విలువను మార్చడానికి ఉపయోగించే operator ను Assignment operators అని అంటారు.

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<<=2	x=x << 2
>>=	x>>2	x=x >> 2
&=	x&=2	X = x & 2
^=	x ^= 2	x = x ^ 2
=	x  = 2	x = x   2

## 6. Increment & Decrement operators :-

Operand విలువ కంటే ఒక విలువ ఎక్కువ లేదా తక్కువ చేసే operator ను Increment లేదా Decrement operators అని అంటారు.

- ++increment operator
- --decrement operator

## 7. Conditional operators:

దీనిని ternary operators అని కూడా అంటారు.

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

## 4. Explain logical operators in java?

A) The logical operators are  
&&(and) , ||(or) and !(not).

OPERATOR	PURPOSE
AND	&&
OR	
NOT	!

**&& (and) :** If both operands are true  
then the logical &&(and) is true

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

**|| (or) :** If either operand is true  
then the logical ||(or) is true.

a	b	a    b
0	0	0
0	1	1
1	0	1
1	1	1



**! (not):** It negates the operand.

<b>a</b>	<b>!a</b>
0	1
1	0

**Example:** a, b and c are integer variables and assigned 3, 5 and 10 respectively.

Operator	Purpose	Expression	Result
&&	And	(a>b)&&(c>a)	False
	Or	(a>b)   (c>a)	True
!	Not	!(a>8)	True

## 5. Explain Relational Operators with examples?

### ∞. Relational Operators: Relational Operators:

- There are six relational operators supported by Java language.

These returns result in the form of 'true' or 'false'.

**Ex:** a, b and c are integer variables and assigned 3, 5 and 10 respectively

There are six following relational operation supported by java language.

OPERATOR	PURPOSE
=	It is equal to
!=	It is not equal to
<	It is less than
>	Greater than
>=	Greater than equal to
<=	Less than equal to

These returns result in the form of 'true' or 'false'.

Operator	Purpose	Relational Expression	Result
= =	is Equal to	c = = 10	False
! =	is Not equal to	a != b	True
>	is Greater than	a>b	False
<	is Less than	(a+b)<c	True
<	is Less than	(a+b)<c	True
> =	is Greater than or equal to	a>=3	True
< =	is Less than or equal to	b < = a	False

## 6. Write about separators in Java?

SEPARATORS	PURPOSE
Parentheses ( )	పద్ధతి నిర్వచనాలు మరియు కాలింగ్ లో వాదనలను జతచేస్తుంది ; అంకగణిత వ్యక్తీకరణలలో ప్రాధాన్యతను సర్దుబాటు చేస్తుంది ; విలువల రకాలను తెలుపుతుంది మరియు ఫ్లో కంట్రోల్ స్టేట్ మెంట్ లలో పరీక్ష వ్యక్తీకరణలను డీలిమిట్ చేస్తుంది.
Braces { }	కోడ్ బ్లాక్ లను నిర్వచిస్తుంది మరియు శ్రేణులను స్వయంచాలకంగా ప్రారంభిస్తుంది.
Brackets [ ]	శ్రేణి రకాలు మరియు dereferences శ్రేణి విలువలను ప్రకటిస్తుంది.
Semi colon ;	ప్రకటనలను రద్దు చేస్తుంది.
Comma ,	వేరియబుల్ డిక్లరేషన్ లలో వరుస ఐడెంటిఫైయర్ లను వేరు చేస్తుంది ; పరీక్షలో చైన్స్ స్టేట్ మెంట్ లు, a for loop యొక్క వ్యక్తీకరణ
Period .	వస్తువు నుండి ఫీల్డ్ లేదా పద్ధతిని ఎంచుకుంటుంది ; ప్యాకేజీ పేర్లను ఉప-ప్యాకేజీ మరియు తరగతి పేర్ల నుండి వేరు చేస్తుంది.
:	లూప్ లేబుల్ ల తర్వాత ఉపయోగించబడుతుంది.

## 7. Explain Bitwise and Bitshift operators in Java?

### A) BITWISE and BIT SHIFT OPERATORS:

Bitwise and bit shift operators are used to manipulate the contents of variables at a bit level according to binary format.

Symbol	Name of the Operator	Example
~	Unary bitwise complement	~op2
&	Bitwise AND	op1 & op2
	Bitwise inclusive OR	op1   op2
^	Bitwise exclusive OR	op1 ^ op2
<<	Signed left shift	op1 << op2
>>	Signed right shift	op1 >> op2
>>>	Unsigned right shift	op1 >>> op2

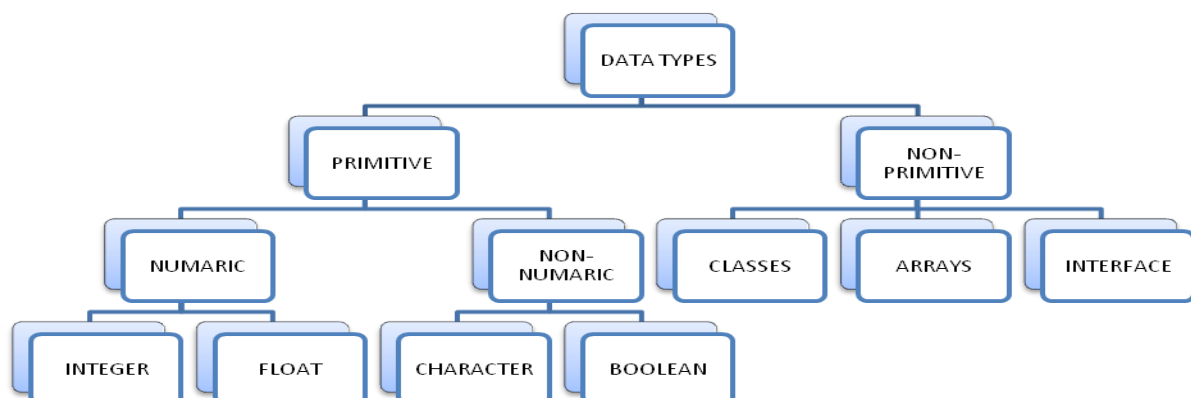
## 8. Explain about java Data Types?

A) 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.



## 9. Write about Method Overloading?

### A) Method Overloading:

- జావాలో ఒకే పేరు ఉన్న పద్ధతులను సృష్టించడం సాధ్యమవుతుంది, కానీ విభిన్న పారామీటర్ జాబితాలు మరియు విభిన్న నిర్వచనాలు. దీన్ని మెథడ్ ఓవర్లోడింగ్ అంటారు. వస్తువులు సారూప్యంగా పని చేయడానికి అవసరమైనప్పుడు ఇది ఉపయోగించబడుతుంది.

### Rules for overloading:

- ✓ అప్పుడు పంపిన పరామితి సంఖ్యలో పద్ధతులు తేడా ఉండవచ్చు.
- ✓ పారామీటర్ల డేటా రకాలు మరియు పారామీటర్ల సీక్వెన్స్లో తేడా ఉండవచ్చు.

### 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) }
    } ఇక్కడ, మనము కస్ట్రక్టర్ మెథడ్ room ()ని ఓవర్లోడ్ చేస్తాము.
```

Room r1= new room(25,20) చేతిలో, room చతురస్రంగా ఉంటే, మనము సంబంధిత వస్తువును Room r2= new room (40).

## UNIT-III

### కంట్రోల్ స్టేట్ మెంట్ & అర్రెస్

### Control Statements and Array

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

1. if statement యొక్క syntax ను వివరించండి . Write the Syntax of If statement

జ. If statement:

- If అనునది ఒక condition statement. దీని ద్వారా Programలో condition ను check చేసి true value వస్తేనే statement ఎగ్జిక్యూట్ (Execute) అవుతుంది. false value వస్తే next statement ఎగ్జిక్యూట్ (Execute) అవుతుంది.

**Syntax:** if (test \_ condition)

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

2. Break statement అనగానేమి? Write the Syntax of Break statement ?

జ. Break statement

- Break అనునది ఒక loop నుండి కంట్రోల్ ను ఎగ్జిక్యూషన్ నుండి బయటకు తీసుకొని రావటానికి Break statement ఉపయోగపడుతుంది. Break అనునది ఒక unconditional control structure
- దీనిని ఎక్కువగా Switch మరియు if, (), for (), while (), do- while () లో ఉపయోగిస్తారు.

**Syntax :-** break;

Example: 

```
int i;
for(i=0;i<=10;i++)
{
    if(i===5)
        Break;
    System.out.println(i);
}
```

### 3. Continue Statement ను గురించి వ్రాయండి?

Write the Syntax of Continue statement ?

జ. Continue Statement:

- Continue statement ను ఉపయోగిస్తే కండిషన్ వచ్చిన తర్వాత loop continue చేసి తరువాతా iteration తో ఎగ్జిక్యూట్ చేస్తుంది.
- దీనిని for, while, do-while లలో ఉపయోగించవచ్చును. దీని ద్వారా loop ను terminate చేయకుండా next iteration లో loop ను continue చేస్తుంది.

**Syntax :-** continue;

```
Example:  int i;
          for(i=0;i<=10;i++)
          {
              if(i==5)
                  Continue;
              System.out.println(" "+i);
          }
```

### 4 . Array అనగానేమి ? Define an array ?

జ. Array:

- ఒకే రకమైన మూలకాల సముదాయాన్నే Array అంటారు. మూలకాలాన్ని memory లో ప్రక్కప్రక్క లొకేషన్స్ లో load అవుతాయి.
- ఒకే data type కలిగిన వివిధ data elements ఒకే పేరు తో save చేస్తే వాటిని array అని అంటారు. ఈ data elements అన్ని memory లో ఒకే చోట నిల్వ చేయబడతాయి.

**Declaration of Array :- Syntax :-** Data type Array-name [size]

### 5. Array యొక్క రకాలు? Write different the Types of arrays?

జ. Array లు ముఖ్యంగా మూడు రకాలుగా విభజించారు. అవి

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

## 6. String array అనగా నేమి? What is a String Array?

### జ. String array:

- String అనగా అక్షర సమూహము character array ద్వారా strings ను store చేస్తాము.

Example:

- char a[]= new char[3];
- a[0]='c', a[1]='s', a[2]='e' i.e. "cse"

## 7. Buffer classes అనగా నేమి? What about Buffer classes?

### జ. Buffer classes:

- Buffer class అనునది string class ను define చేసినప్పుడు మనము ఇచ్చే fixed length ను flexible చేయుటకు గాను ఉపయోగపడుతుంది. మన string లో కొత్తగా character ను add చేయుట(append) లేదా string మధ్యలో కొత్త అక్షరాలను అమర్చుట (insert) వంటి పనులను buffer class సమర్థిస్తుంది.

**String Buffer append(Val):** Val may be Boolean int, short, long etc.

## 8. String method అనగా నేమి? What is String method?

### జ. String Methods:

- స్ట్రింగ్ క్లాస్ అనేక రకాల స్ట్రింగ్ మానిప్యులేషన్ టాస్క్స్ ని పూర్తి చేయడానికి మనకు అనుమతించే అనేక పద్ధతులను నిర్వచిస్తుంది.

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.
5) S1.equals (S2)	Returns a true if S1 is true to S2.

## 9. Nested loops అనగా నేమి? Write about Nested Loops ?

### జ. Nested loops:

- ఒక looping statement లో వేరొక looping statement వ్రాయడాన్ని Nested loop అని అంటారు. (లేదా) ఒక loop లో మరియొక loop వచ్చే ప్రక్రియను Nested loop అంటారు.

```
Syntax: if(test _conditin _ 1)
    {
        if(test _conditin _ 1)
        { statement _ 1;
        }
        else
        { statement _ 2
        } statement _ n;
```

## 10. Switch statement ను గూర్చి వ్రాయండి. Explain switch statement with example?

### జ. Switch statement:

- ఒకేసారి multiple conditions ని check చేయవలసి వస్తే switch statement ని ఉపయోగిస్తారు. ఒక్కొక్క condition ఒక case లో వ్రాయబడి condition true అయితే ఆ case statements ను execute చేస్తుంది.

**Syntax:** switch(Expression)

```
{ Case Constant-value _ 1: Statement-block _ 1;
                                Break;
  Case Constant-value _ 2: Statement-block _ 1;
  -----                    Break;
  Case Constant-value _ n: Statement-block _ n;
  Default:                    default-statement block;
                                Break;
} Statement-X;
```



1. Java లో ఉపయోగించే Conditional Control statement ను వివరించుము?

Explain the conditional statement in java?

జ. Java లో ఉపయోగించే conditionalControl structures:

Java లో condition ఆధారంగా statement ఎగ్జిక్యూట్ (Execute) కావాలా, వద్దా తెలియజేసే Control structure condition, Control structure అంటారు.

1. if statement
2. if-else statement
3. nested-if statement
4. if- else ladder
5. switch statement

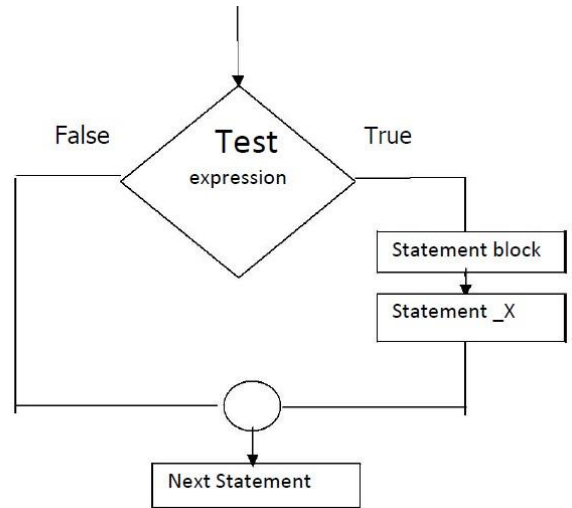
1. If Statement:

- if స్టేట్ మెంట్ (statement) లో condition true అయితే stament ఎగ్జిక్యూట్ అవుతుంది false అయితే next statement ఎగ్జిక్యూట్ అవుతుంది

Syntax:-

```
if (condition)
{
    Statement;
}
next statement;
```

If Statement flow chart:



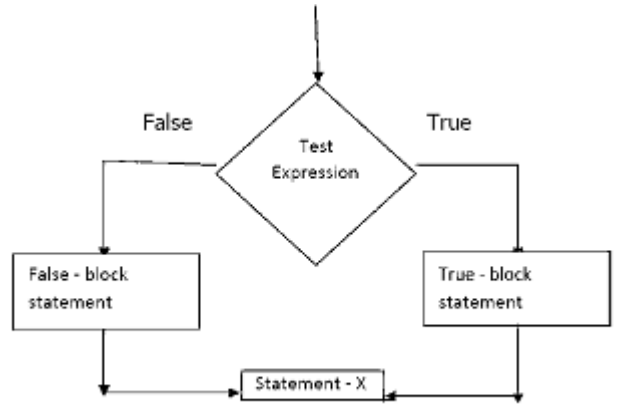
## 2. if else statement:

- if else statements condition true అయితే statement 1 ఎగ్జిక్యూట్ (Execute) అవుతుంది false అయితే statement 2 ఎగ్జిక్యూట్ అవుతుంది

**Syntax:-**

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

**If else Statement flow chart:**



## 3. Nested if Statement:-

- ఒక if లోపల మరియొక if statement ఉంటే దానిని Nested if అంటారు .

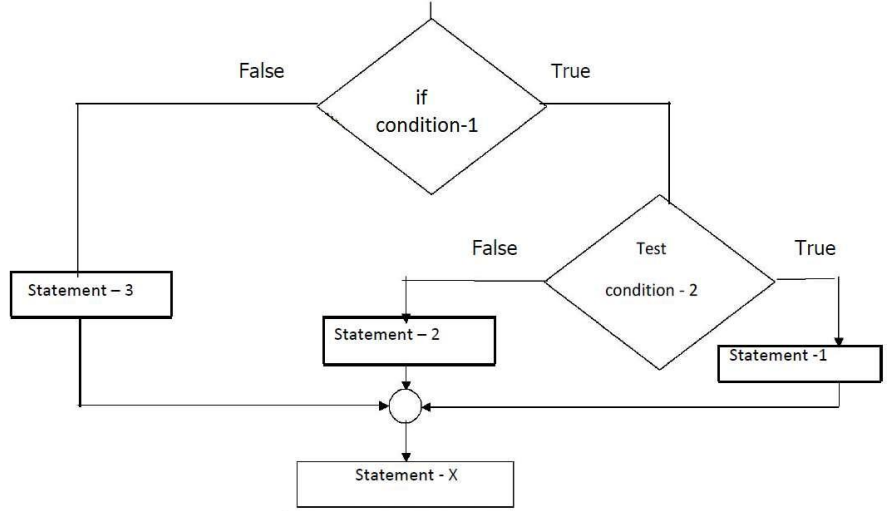
**Syntax:-**

```
if ( test _ condition _ 1)
{
  if (test _ condition _ 2)
  {
    statement _ 1;
  }
}
else
{
  statement _ 2;
}
```

else

```
{ statement _ 3;  
} statement _ n;
```

### Nested if Statement flow chart:



#### 4. ELSE.....IF ladder:

- Multipath ప్రమేయం ఉన్నప్పుడు if's కలిసి ఉపయోగించడం మరొక మార్గం. Multipath నిర్ణయం అనేది ifs యొక్క గొలుసు , దీనిలో ప్రతి ఇతరతో అనుబంధించబడిన స్టేట్‌మెంట్ if ఉంటుంది.

Syntax:-

```
if (condition _ 1)  
statement _ 1;  
else if ( condition _ 2)  
statement _ 2;  
else if ( condition _ 3)  
statement _ 3;  
-----  
else  
default _ statement;  
statement _ X;
```

2. Java లో switch statement యొక్క ఉపయోగము తెలిపి switch statement ను Syntax లో వివరించండి. Explain switch statement with example?

జ . Java లో switch statement యొక్క ఉపయోగము

- Java లో రెండుగాని అంతకన్నా ఎక్కువ ఆప్షన్ ల నుండి ఒకటి మాత్రమే ఎగ్జిక్యూట్ చేయవలసిన సందర్భాలలో switch statement ఉపయోగించవచ్చు.

**Syntax :-**

```
switch (Expression)
{
case constant -value _ 1:  statemnet-block _ 1 ;
break ;
case constant-value _ 2 :  statment -block _ 2 ;
break ;
.....
case constant-value _ n :  statemnet-block _ n;
default : statemnet n;
break;
default : statements-block _ n;
break;
}
```

**Sample Program with Switch.. Case is as follows:**

```
import java.util.Scanner;
public class SwitchExample
{
public static void main(String args[])
{
System.out.println("enter a code");
Scanner in = new Scanner(System.in);
int code = in.nextInt();
```

```

switch(code)
{
    case 1 :
        System.out.println("first class"); break;
    case 2:
        System.out.println("second class"); break;
    case 3 :
        System.out.println("third class"); break;
    default :
        System.out.println("fail");
}
}
}

```

### 3. Java loop గురించి వివరించండి ? Explain the loops in java?

జ. జావా లో loops ఆనేవి మూడు రకాలు. అవి

1. while loop
2. do-while loop
3. for loop

#### 1. while loop:

ఎంట్రీ కంట్రోల్ లూప్ statemnet లో ఉన్నప్పుడు కండిషన్ మొదట మదింపు చేయబడుతుంది మరియు ఒకవేళ కండిషన్ సత్యం అయితే, అప్పుడు లూప్ యొక్క బాడీ అమలు చేయబడుతుంది.

**Syntax:-** while (Condition)

```

{
    Body of the loop;
}

```

Example :-

```
S= 0;
i=0;
While (i<_ 10)
{
    S=s+i ;
    i=i+1;
}
System.out.Println (“sum=5d\n”, s);
```

## 2.do-while loop:

do statement అనేది ఎగ్జిట్ కంట్రోల్ లూప్ statement. పని ప్రకటన , ముందుగా లూప్ యొక్క బాడీని మదింపు చేయడం కొరకు ప్రోగ్రాం ప్రోసిడింగ్. లూప్ యొక్క చివరల్లో, condition సత్యమా అని condition లో ఉంది. ఈ కార్యక్రమం లూప్ యొక్క బాడీని మదింపు చేయడం కొరకు కొనసాగుతుంది.ఈ ప్రక్రియ ఎంతవరకు నిజమో పరిస్థితి అలాగే కొనసాగుతుంది. Condition అసత్యం అయినప్పుడు, లూప్ రద్దు అవుతుంది మరియు తరువాత వెంటనే కనిపించే statement కు కంట్రోల్ వెళ్ళుతుంది.

**Syntax:-** do  
{  
Body of the loop;  
} While (condition);

### Example:-

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

### 3. for loop:

The for loop control structure ని కూడా entry controlled structure అని అంటారు. ఇందులో ముఖ్యముగా మూడు భాగాలుగా ఉంటాయి. అవి initialisation, condition మరియు increment / decrement.

#### The Syntax of the for loop is:

For loop is more concise than while loop and do while loop.

```
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);
```

### 4. జావా లో string మెథడ్ గురించి వివరించండి .

What are the string method Explain.

#### జ. స్ట్రింగ్(String ):

- స్ట్రింగ్ అక్షరాల క్రమాన్ని సూచిస్తుంది.

#### స్ట్రింగ్ పద్ధతులు(String methods):

- స్ట్రింగ్ క్లాస్ అనేక రకాల స్ట్రింగ్ మానిప్యులేషన్ టాస్క్ని పూర్తి చేయడానికి మాకు అనుమతించే అనేక పద్ధతులను నిర్వచిస్తుంది.

Method call	Task performed
1) S2 = S1.toLowerCase();	Convert the string S1 to all lower case.
2) S2=S1.toUpperCase();	Convert the string S1 to all upper case.
3) S2=S1.replace('a','b')	Replace all appearances of a with b
4) S2=S1.equalsIgnoreCase(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 nth character of S1
9) S1.Comparator (S2)	Returns negative 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 nth character
12) S1.substring (n, m)	Gives substring starting from nth character up to in the character (not including mth)
13) String.valueOf (P)	Creates a string object of the parameter P(simple) type of object
14) P. toString ()	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 (variable)	Converts the parameter value to string representation



5. జావాలో స్ట్రింగ్ బఫర్ క్లాసు గురించి వివరించండి

Explain the string Buffer class in Java?

జ. Buffer classes:

- Buffer class అనునది string class ను define చేసినప్పుడు మనము ఇచ్చే fixed length ను flexible చేయుటకు గాను ఉపయోగపడుతుంది. మన string లో కొత్తగా character ను add చేయుట(append) లేదా string మధ్యలో కొత్త అక్షరాలను అమర్చుట (insert) వంటి పనులను buffer class సమర్థిస్తుంది.

**String Buffer append(Val):** Val may be Boolean int, short, long etc.

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 ith to J-1th postion In the sting 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 nth character to X
* Int set length (n)	sets the length of the string S1 to n. if n>S1. Length() zeros one added to S1.

6. String buffer మరియు builder class ల మధ్య తేడాలు ఏమిటి?

What are the differences between string buffer and string builder classes.

జ. String buffer మరియు builder class ల మధ్య తేడాలు

Java మూడు తరగతులను, అక్షరాల వరుసక్రమాన్ని సూచిస్తుంది. sttring, string buffer మరియు string builder. string class అనేది ఒక

BASIC FOR COMPARISON	STRING BUFFER	STRING BUILDER CLASSES
ప్రాథమిక	స్ట్రింగ్ వస్తువు యొక్క పొడవు స్థిరంగా ఉంటుంది.	స్ట్రింగ్ బఫర్ యొక్క పొడవును పెంచవచ్చు
సవరణ	స్ట్రింగ్ వస్తువు మార్పులేనిది	స్ట్రింగ్ బఫర్ ఆబ్జెక్ట్ మార్చదగినది
ప్రదర్శన	సంయోగం సమయంలో ఇది నెమ్మదిగా ఉంటుంది.	సంయోగం సమయంలో ఇది వేగంగా ఉంటుంది
జ్ఞాపకశక్తి	ఎక్కువ మెమరీ వినియోగిస్తుంది	తక్కువ మెమరీని వినియోగిస్తుంది.
నిల్వ	స్ట్రింగ్ స్థిరమైన పూల్.	హీప్ మెమరీ.

7. జావాలో ప్రోగ్రాం లో 1 నుండి n వరకు ప్రింట్ చెయ్యడాన్ని వివరించండి?

Write a java program to print 1 to n?

జ. Java program to print 1 to n:

```
import java.util.Scanner;
public class Print_1_To_N_UsingWhile
{
    public static void main(String[] args)
    {
        int i =1 ;    //loop counter initialisation
        //create object of scanner class
        Scanner Sc = new Scanner(System.in);
        // enter the value of " n "
        System.out.print("Enter the value n : ");
        int n = Sc.nextInt(); // read the value.
        System.out.println("Numbers are : ");
        while(i<=n)
        {
            System.out.println(i);
            i++;
        }
    }
}
```

Output: java -cp /tmp/z98rheCILD Print\_1\_To\_N\_UsingWhile

Enter the value n : 3

Numbers are :

1

2

3

8. ಜಾವಾ ಪ್ರೋಗ್ರಾಂಮ್ ಲ್ಲ addition of two matrices ವ್ರಾಯಂಡಿ ?

Write a java program addition of two matrices?

ಃ. Java program addition of two matrices:

```
import java.util.Scanner;
class AddTwoMatrix
{
    public static void main(String args[])
    {
        int m, n, c, d;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows and columns of matrix");
        m = in.nextInt();
        n = in.nextInt();
        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];
        System.out.println("Enter the elements of first matrix");
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                first[c][d] = in.nextInt();
        System.out.println("Enter the elements of second matrix");
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                second[c][d] = in.nextInt();
        for ( c = 0 ; c < m ; c++ )
            for ( d = 0 ; d < n ; d++ )
                sum[c][d] = first[c][d]+ second[c][d];
        System.out.println("First Matrix :-");
```

```

    for ( c = 0 ; c < m ; c++ ) {
        for ( d = 0 ; d < n ; d++ )
            System.out.print(first[c][d]+"\\t");
        System.out.println();
    }
    System.out.println("Second Matrix:-");
    for ( c = 0 ; c < m ; c++ ) {
        for ( d = 0 ; d < n ; d++ )
            System.out.print(second[c][d]+"\\t");
        System.out.println();
    }
    System.out.println("Sum of entered matrices:-");
    for ( c = 0 ; c < m ; c++ ) {
        for ( d = 0 ; d < n ; d++ )
            System.out.print(sum[c][d]+"\\t");
        System.out.println();
    }
} }

```

**Output:**

Enter the number of rows and columns of matrix

2

2

Enter the elements of first matrix

1 2

3 4

Enter the elements of second matrix

5 6

7 8

First Matrix :-

1    2

3    4

Second Matrix:-

5    6

7    8

Sum of entered matrices:-

6    8

10   12

**9. Write a java program to sort "N" numbers in an array?**

**æ. Java program to srot "N" numbers in an array:**

```
import java.io.*;
import java.lang.*;
public class SortAsc {
public static void main(String[] args)
{
    //Initialize array
    int [] arr = new int [] {5, 2, 8, 7, 1};
    int temp = 0;
    //Displaying elements of original array
    System.out.println("Elements of original array: ");
    for (int i = 0; i < arr.length; i++)
    {
        System.out.print(arr[i] + " ");
    }

    //Sort the array in ascending order
    for (int i = 0; i < arr.length; i++) {
        for (int j = i+1; j < arr.length; j++) {
```

```

        if(arr[i] > arr[j]) {
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
}

System.out.println();
//Displaying elements of array after sorting
System.out.println("Elements of array sorted in ascending order: ");
for (int i = 0; i < arr.length; i++)
{
    System.out.print(arr[i] + " ");
} } }

```

**Output:**

Elements of original array:

5 2 8 7 1

Elements of array sorted in ascending order:

1 2 5 7 8

**10. Write a java program multiplication of two matrices?**

**∞. Java program multiplication of two matrices**

```

import java.util.Scanner;

class Mulmat {

    public static void main(String args[]) {

        int m,n,i,j,k;

        Scanner in=new Scanner(System.in);

System.out.println("Enter the number of rows and columns of matrix:");

```

```

m=in.nextInt();
n=in.nextInt();
int a[][]=new int[m][n];
int b[][]=new int[m][n];
int c[][]=new int[m][n];
System.out.println("Enter the elements of first matrix:");
for(i=0;i<m;i++)
    for(j=0;j<n;j++)
        a[i][j]=in.nextInt();
    System.out.println("Enter the elements of b matrix:");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            b[i][j]=in.nextInt();
    for(i=0;i<m;i++){
        for(j=0;j<n;j++) {
            c[i][j]=0;
            for(k=0;k<2;k++) {
                c[i][j]=c[i][j]+(a[i][k]*b[k][j]); / Multiplication of two matrices
            } } }
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            System.out.print(c[i][j]+"\\t");
        System.out.println();
    } } }

```



**Output:**

Enter the number of rows and columns of matrices 2 2

Enter the elements of first matrix: 1 2 3 4

Enter the elements of second matrix: 1 2 3 4

7 10

15 22

**11. Write a java program to find the factorial of given number?**

æ. The factorial of given number:

```
import java.io.*;
class FactorialExample
{
public static void main(String args[])
{
int i,fact=1;
int number=5;//It is the number to calculate factorial
for(i=1;i<=number;i++)
{
fact=fact*i;
}
System.out.println("Factorial of "+number+" is: "+fact);
}
}
```

Output:

Factorial of 5 is: 120

12. Write a java program to find whether the given number is perfect or not?

∞. The given number is perfect or not:

```
import java.util.Scanner;
class Perfect {
    public static void main(String arg[]) {
        long n,sum=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number");
        n=sc.nextLong();
        int i=1;
        while(i<=n/2)
        {
            if(n%i==0)
            {
                sum+=i;
            }
            i++;
        }
        if(sum==n) {
            System.out.println(n+" is a perfect number");
        }
        else
            System.out.println(n+" is not a perfect number");
    }
}
```

Output:

Enter a number

6

6 is a perfect number

### 13. Write a java program to transpose of given matrices?

æ. To transpose of given matrices:

```
import java.io.*;

public class Transpose {

    public static void main(String[] args) {

        int row = 2, column = 3;

        int[][] matrix = { {2, 3, 4}, {5, 6, 4} };

        display(matrix);    // Display current matrix

        // Transpose the matrix

        int[][] transpose = new int[column][row];

        for(int i = 0; i < row; i++) {

            for (int j = 0; j < column; j++) {

                transpose[j][i] = matrix[i][j];

            } } display(transpose); // Display transposed matrix

    } public static void display(int[][] matrix) {

        System.out.println("The matrix is: ");

        for(int[] row : matrix) {

            for (int column : row) {

                System.out.print(column + " ");

            }

            System.out.println();

        } } } }
```

Output: The matrix is:

2 3 4

5 6 4

The Resultant matrix is:

2 5

3 6

4 4

-----

## Unit-IV

### Implementing oops in Java, Java Methods

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1.Method అనగానేమి? What is a Method?

##### జ. Method:

- class లో ఉండే ఫంక్షన్ లనే methods అంటారు.ఇది ఆబ్జెక్ట్ లో ఉన్న డేటా మీద ఏవైనా ఆపరేషన్స్ చేయడానికి ఇవి ఉపయోగపడతాయి.వీటికి ఒక పేరు parameters (ఆప్షన్) return type body మొదలైనవి ఉంటాయి.

For example: public double sum (double a double b)  
{  
    Return (a+b)  
}

#### 2. Java లో call method ఎలా? How to call method?

##### జ. To call method:

- This statement calls the my method ( ) method that was declared earlier

```
Class Main {  
    Public static void main(String[] args)  
    { myFunction();  
    }  
    Private static void myFunction()  
    {  
        // function body  
    }  
}
```

(1)

(2)

(3)

### 3. పద్ధతుల(method) ద్వారా లాభాలు ?

What are the advantages of using methods?

జ. Advantages of using methods:

- ప్రధాన ప్రయోజనాలు code reusability.
- ఒక సారి method ను రాయవచ్చు మరియు పలు సార్లు ఉపయోగించవచ్చు.
- ప్రతిసారి మొత్తం code ని తిరిగి రాయవలసిన పని లేదు
- Write once, reuse multiple times

### 4. constructor అనగానేమి ? What is a Constructor?

జ. Constructor:

- constructor అనునది ఒక మెంబెర్ ఫంక్షన్, constructor పేరు కూడా క్లాస్ పేరు వలె ఉంటుంది.
- ఇది ఆబ్జెక్ట్ ను create చెయ్యగానే ఆటోమేటిక్ గా కాల్ అవుతుంది.

### 5. Overloading అనగానేమి ? What is a Over loading?

జ. Over loading:

- Method overloading రెండు లేదా అంతకన్నా ఎక్కువ methods ఒకే పేరుతో ఉండే వాటిని overloading అంటారు.

### 6. inheritance అనగానేమి? What is a Inheritance?

జ. Inheritance:

- జావాలో inheritance అనునది subclass మరియు super class కు మధ్య సంబంధాన్ని తెలియజేస్తుంది.
- inheritance ద్వారా ఇది వరకే ఉన్న classes ల నుండి కొత్త class లను సృష్టించవచ్చు.
- దీనిలో ఈ క్రింది విధముగా inheritance రకాలు కలవు.
  - 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)

## 7. Polymorphism అనగానేమి? What is a Polymorphism?

### జ. Polymorphism:

- Polymorphism అనగా ఒకే object వివిధ రూపాలలో కనిపించడము.
- Polymorphism ను oop లో child class object లో parent class refernce ను ఉపయోగిస్తారు. దీనిలో operator overloading , function overloading వంటివి ఉంటాయి.

## 8. Overriding అనగానేమి? What is a Overriding?

### జ. Overriding:

- దీనిలో ఒకేరకమయిన function signature తో parent class లోనూ ఉంటుంది. Function యొక్క signature ఒకే విధంగా ఉంటుంది.
- ఒకటి parent class లోను మరొకటి child class లోను ఉంటుంది. దీనిలో function యొక్క ప్రవర్తన తీరు వేరుగా ఉంటుంది. redefinition అని కూడా పిలుస్తారు.

దీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

### 1. Java methods with arguments and return value గురించి వివరించండి?

Explain the Java methods with arguments and return value?

### జ. Java methods with arguments and return value:

1. పద్ధతి నుండి విలువను తిరిగి ఇవ్వండి
2. వాదనలను అంగీకరించడం మరియు విలువను తిరిగి ఇచ్చే విధానం
3. కామాలను ఉపయోగించడం ద్వారా జావా పద్ధతికి ఒకటి కంటే ఎక్కువ ఆర్గ్యుమెంట్లను పంపడం

➤ జావా పద్ధతి సున్నా లేదా అంతకంటే ఎక్కువ పారామితులను కలిగి ఉంటుంది. మరియు, వారు విలువను తిరిగి ఇవ్వవచ్చు. ఉదా: పద్ధతి నుండి తిరిగి విలువ

### 1. 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 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.

## 2. 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);
        Syatem.out.println ("Square of 4 is : "+ result);
    }
    Statc 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

### 3. Pass more than one argument to the java method by using commas.

```
Public class Arithmetic 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

### 2. తుది వేరియబుల్స్ మరియు methods గురించి వివరించండి?

**Explain the final variables and methods?**

#### జ. 1. Java final variable:-

ఒకవేళ ఏదయినా వేరియబుల్ ని ఫైనల్ చేసినట్లయితే, final variable యొక్క విలువను మార్చలేము(స్థిరం గా ఉంటుంది). తుది variable యొక్క ఉదాహరణ :- ఫైనల్ వేరియబుల్ స్పీడ్ లిమిట్ ఉంది, ఈ వేరియబుల్ మార్చాలనుకున్నాము ,అయితే దీనిని మార్చలేం , ఎందుకంటే ఒక్కసారి కేటాయించబడ్డ ఫైనల్ వేరియబుల్ ని ఎన్నటికి మార్చలేము.

ఉదాహరణ: Class bike 100 {  
 Final speed limit = 100; // తుది వేరియబుల్ చెల్లుబాటు కాని రన్నింగ్  
 {  
 speed limit =100; }  
}



```

Public static void main (string args [ ])
{
    Bike 100 ఓబజి= new bike ( ); ఓబజి running ( );
}
} // క్లాసు అవుట్ పుట్ ముగింపు

```

## 2. Java final method :-

class లో ఉండే ఫంక్షన్ లనే methods అంటారు. ఇది ఆబ్జెక్ట్ లో ఉన్న డేటా మీద ఏవైనా ఆపరేషన్స్ చేయడానికి ఇవి ఉపయోగపడతాయి. వీటికి ఒక పేరు para meters (ఆప్షన్) return type body మొదలైనవి ఉంటాయి.

ఫైనల్ గా ఏదయినా పద్ధతి ప్రకారం చేస్తే దాన్ని override చెయ్యలేరు

Final method కి ఉదాహరణ :-

```

Class bike 1000 {
    Final void running ( )
    {
        System.out.println( "bike నడుపుతున్నాడు");
    }
}
Class Yamaha extends bike running( )
{
    System.out.println("bike 90 kmph తో రన్ అవుతుంది సురక్షితంగా");
}
public static void main ( string args [ ])
{
    Yamaha y = new Yamaha();
}
Running();
}
}

```

### 3. Inheritance విధానాన్ని example లో వివరించండి.

Explain the inheritance with example in java?

జ. Inheritance:

- వారసత్వం(Inheritance) అనేది ఒక తరగతికి చెందిన వస్తువులు మరొక తరగతి వస్తువుల లక్షణాలను పొందే ప్రక్రియ.
- వారసత్వం క్రమానుగత వర్గీకరణ భావనకు మద్దతు ఇస్తుంది.
- వారసత్వం పునర్వినియోగ ఆలోచనను అందిస్తుంది.
  - వారసత్వం ద్వారా మేము ఇప్పటికే ఉన్న తరగతులపై నిర్మించబడిన కొత్త తరగతులను సృష్టించవచ్చు.
  - మనము ఇప్పటికే ఉన్న తరగతి నుండి వారసత్వంగా పొందినప్పుడు, మేము పద్ధతులు మరియు ఫీల్డ్ లను తిరిగి ఉపయోగించుకోవచ్చు మరియు కొత్త పరిస్థితులకు మా కొత్త తరగతిని స్వీకరించడానికి మేము కొత్త పద్ధతులు మరియు ఫీల్డ్లను జోడిస్తాము.

**Syntax:**

```
class <subclass _name> extends <superclass _name>
{
    //methods and fields
}
```

మనము ఇప్పటికే ఉన్న తరగతి నుండి ఉద్భవించిన కొత్త తరగతిని తయారు చేస్తున్నామని కీవర్డ్ విస్తరింపజేస్తుంది. అంటే , వారసత్వంగా వచ్చిన తరగతిని సూపర్క్లాస్ అంటారు. కొత్త తరగతిని సబ్క్లాస్ అంటారు.

**Example:**

```
class Employee{
    int salary=60000;
}
class Programmer extends Employee{
    int bonus= 10000;
}
```

```

class InterfaceExample{
    public static void main(String args[ ]){
        programmer p=new Programmer( );
        System.out.println("Programmer salary is: " +p.salary);
        System.out.println("Programmer bonus is: " +p.bonus); } }

```

Output:

Programmer salary is: 60000 Programmer bonus is: 10000

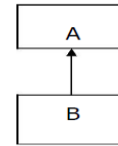
#### 4. వివిధ రకాల inheritance ను తెలుపండి? How many Types of inheritance in Java?

జ. Inheritance అనునది ఐదు రకాలుగా జరుగుతుంది.

- Single inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Multiple inheritance
- Hybrid inheritance

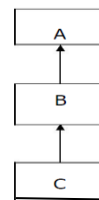
##### 1) Single Inheritance:

ఒక class నుండి వేరొక class ను తయారు చేయుటను single inheritance అని అంటారు. class A ను parental class గాను, B ని child class గాను గుర్తిస్తారు. Class A యొక్క లక్షణాలు class B లో ఉన్న object పై ప్రభావితం చేస్తాయి.



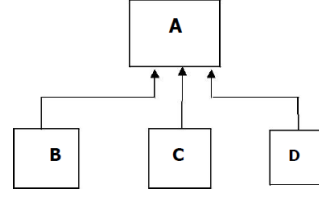
##### 2) Multilevel inheritance:

ఒక class నుండి వేరొక class ను తయారు అయితే దాని నుండి మరల వేరే class ను తయారు చేయబడితే ఈ ప్రక్రియని Multilevel inheritance అని అంటారు.



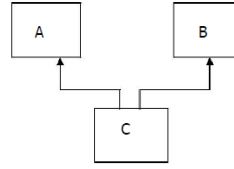
### 3) Hierarchical inheritance:

ఒక class నుండి ఒకటి కన్నా ఎక్కువ class లు తయారు చేయబడితే ఆ ప్రక్రియను Hierarchical inheritance అని అంటారు.



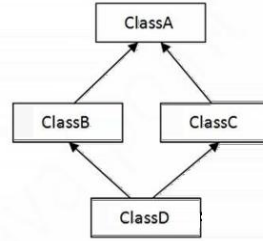
### 4) Multiple inheritance:

ఒకటి కన్నా ఎక్కువ class లు కలయిక తో కొత్త class ను సృష్టించబడే ప్రక్రియను Multiple inheritance అని అంటారు.



### 5) Hybrid inheritance:

Multilevel మరియు multiple inheritance యొక్క కలయికనే Hybrid inheritance అని అంటారు



5. Method overloading పద్ధతిని ఉదాహరణలతో వివరించండి?

Explain the method overloading with example in Java?

జ. Method overloading:

వివిధ input parameter లు కలిగి ఒకే పేరు తో అనేక method లను ఒక class లో వ్రాస్తే over loading అని అంటారు. Overloading అనునది compile టైమ్ polymorphism అని అంటారు.

Overloading అనునది ఒకే రకమైన code ను ఎక్కువ సార్లు రాయుటను నియంత్రిస్తుంది. Compilation టైమ్ లో method ని select చేసుకొనే వివిధ input మరియు output parameter లతో execute చేస్తుంది.

**Example :-**

Class parent-class

```
{ Parent class();  
  {  
    System.out.println("construct of parent");  
  }  
  Void disp()  
  { System.out.println("parent method");  
  }  
}
```

Class java example extends parent-class

```
{ Java example ()  
  {  
    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

## 6. Operator overloading ఆనగా నేమి? ఉదాహరణలతో వివరించండి.

Explain the operator overloading in java.

### జ. Operator overloading:

User defined data types తో overloading జరుగుటకు java support చేయదు. Overloading అనునది built –in data type కి వర్తించదు. Java ఒకటి కన్నా ఎక్కువ operator స ఒకేసారి ఉపయోగించదు. అది కేవలము string concationation వద్దనే జరుగుతుంది.

### 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 );
}
```

- ఇది తప్పుగా ముద్రిస్తుంది మరియు మరేమీ లేదు.
- + ఆపరేటర్, అదనంగా లేదా స్పింగ్ సంయోగం కోసం ఉపయోగించబడినా == ఆపరేటర్ కంటే ఎక్కువ ప్రాధాన్యతను కలిగి ఉంటుంది.
- అందువల్ల println పద్ధతి యొక్క పరామితి ఇలా మూల్యాంకనం చేయబడుతుంది.  
System.out.println("first and second are equal: "+ first == second);

## 7. Polymorphism ఆనగా నేమి? ఉదాహరణలతో వివరించండి.

Explain the polymorphism with example?

### జ. Polymorphism:

Polymorphism ఆనగా అనేక functions ను ఒకే form లోకి తీసుకురావాడం. ఇది oops యొక్క ముఖ్య లక్షణం. Extended class లో ఉండే అనేక object లను parent class యొక్క variable గా తీసుకొనవచ్చు కానీ వీటి యొక్క reference variable గా declare చేసిన తరువాత మరలా మార్చలేము.

Reference variable లో వేరే object కి reference గా లేదా reference subtype గా declare చేయవచ్చును లేదా class లేదా interface type గా parental class లో declare చేయవచ్చు.

**Example:**

```
class Box
{
    int w,h;
    void info( )
    {
        System.out.println("This is a simple box");
        System.out.println("width = "+w+"height="+h);
    }
}

class woodenBox extends Box
{
    int life;
    void info( )
    {
        System.out.println("This is a wooden box");
    }
}

class SteelBox extends Box
{
    int wg;
    void info( ) {
```

```

        System.out.println("This is a steel box");
    } }
class LargewoodenBox extends woodenBox {
    void info( ) {
        System.out.println("This is a Huge wooden Box"); } }
class BoxDemo {
    public static void main(String ary[ ]) {
        Box x;
        Box b1= new Box( );
        woodenBox wb=new woodenBox( );
        SteelBox s1=new SteelBox( );
        LargewoodenBox p1=new LargewoodenBox( );
        b1.info( );
        wb.info( );
        s1.info( );
        p1.info( );
    } }

```

Output:

This is a simplebox

Width=0 hieght=0

This is a wooden box

This is a steel box

This is a Huge wooden Box

-----



## UNIT-V

### ప్యాకేజీస్ & ఇంటర్ఫేస్

## PACKAGE AND INTERFACES

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

1. ప్యాకేజీ (Package) అనగానేమి ? Define package.

జ. ప్యాకేజీ (Package):

- User defined packageలను User తన అవసరాలకు అనుగుణంగా తయారుచేస్తారు
- సాధారణంగా వీటిని Reusability కి పనికివచ్చే విధంగా వ్రాస్తారు.
- దీనికోసం source file పైన 'package' అనే keyword తో మొదలు పెడతారు.

**Syntax :** 1. To create a package:

```
package package_name;
```

2. To import a package:

```
import package_name.*;
```

2. ప్యాకేజీ (Package) రూపొందించండి ? How to Create a package?

జ. Create a package:

- ప్యాకేజీ యొక్క పేరును ఉపయోగించి ప్యాకేజీ పేరు తరువాత ప్యాకేజీ పేరును మేం ప్రకటిస్తారు . జావా సోర్స్ file లో ఇది మొదటి statement గా ఉండాలి.

**Syntax:**

1. To create a package:

```
package package_name;
```

Ex :- package first package;

```
Public class first-class
```

```
{
```

```
-----;
```

```
}
```

### 3. ప్యాకేజీ(Package) ని యాక్సెస్ చేసుకోవడం? How to Accessing a package?

#### జ. Accessing a package:

- క్లాసుల ప్యాకేజీలను ఇంపోర్ట్ చేసుకోవడం కొరకు ఇంపోర్ట్ statement ను ఉపయోగిస్తాం. ఇంపోర్ట్ statement ని నిర్దిష్ట క్లాసు కొరకు ప్యాకేజీల జాబితాకు ఉపయోగించవచ్చు.

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

### 4. ఇంటర్ఫేస్ అనగా నేమి? What is an Interface?

#### జ. ఇంటర్ఫేస్ (interface):

- ఇంటర్ఫేస్ (interface) అనగా వివిధ abstract methods యొక్క కలయిక.
- ఒక class ద్వారా interface ను తయారు చేయాలంటే abstract methods గా define చేయాలి.
- Interface ను తయారు చేయుటకు interface అనే keyword ను ఉపయోగించాలి.

**Syntax:** interface interface \_name  
{  
Variable declaration;  
Methods declaration;  
}

### 5. Abstract class అనగా నేమి? What is Abstract class?

#### జ. Abstract class:

- Abstract class అనగా ఒకటి లేదా అంతకన్నా ఎక్కువ Abstract methods declare చేసిన వివిధ class యొక్క కలయిక.
- వీటిని మనము direct గా implement చేయలేము. దీని కొరకు Sub-class అవసరము పడుతుంది.

An example abstract class in Java

```
abstract class Shape {  
    int colour;  
    // An abstract function (like a pure virtual function in C++)  
    abstract void draw(); } }
```

## 6. Sub-class నిర్వచించండి. Define Sub-class.

### జ. Sub-class:

- Super class నుండి properties ను వారసత్వం లో తీసుకొను దానిని Sub-class అని అంటారు.
- Super class ఆధారముగా Sub-class తన objects ను తయారు చేయవచ్చును.

**Syntax:** interface <interface name>  
{  
    Variable declaration;  
    Methods declaration;  
}

దీర్ఘ సమాధాన ప్రశ్నలు మరియు జవాబులు

### 1. ప్యాకేజీ ల యొక్క రకాలను వివరించండి? Explain the types of packages.

#### జ. ప్యాకేజీలు(package) రెండు రకాలు

- API package, java
- User defined package.

#### Java API package:

- ఇది దీర్ఘకాలం functionality అనుగుణంగా విభిన్న ప్యాకేజీలకు వర్గీకరించబడ్డ గ్రూపును పెద్ద సంఖ్యలో అందిస్తాయి. యుటిల్ ,అవుట్ , అప్లెట్ మొదలైనవి.

Package	Purpose
Java Long	దీనిలో సాధారణంగా class data types మరియు string math function లు ఉంటాయి.
Java .util	వివిధ languages, utility classes మరియు వివిధ సంఖ్యా స్థానాల ఉంటాయి.
Java . Applet	దీనిలో విండోస్ బటన్ లిస్టు ,మైనస్ మరియు abstract window toolkit కోసం తరగతులు ఉంటాయి.
Java. IO	ఈ package లో input మరియు output class లు కలిగి ఉంటాయి.
Java . net	Net ఆనగా Network అని అర్థం. ఈ package ముఖ్యంగా client, server program కి ఉపయోగిస్తారు.

<b>Java.text</b>	ఈ ప్యాకేజీ సంఖ్యా విలువలను ఫార్మాట్ చేయడానికి ఉపయోగాకరముగా ఉంటుంది(తేదీల కోసం తేదీ ఫార్మాట్ class)(విలువల కోసం నెంబర్ ఫార్మాట్ తరగతి).
<b>Java.sql</b>	Sql అనేది Structured Query language ఈ package database ను తయారు చేయుటకు data ను నిల్వచేయుటకు మరియు data ను తిరిగి వాడుకొనుటకు ఉపయోగిస్తారు.

### Java User defined package:

- User తన అవసరార్థం కొన్ని class లను నిర్దేశిస్తూ తయారు చేసే package ను user defined package అని అంటారు.

**Syntax:** package <package name >

Ex : package student

Example :-

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

2. Package ను తయారు చేయు విధానాన్ని తెలుపండి.

Explain to create own package with example.

జ. Package ను తయారు చేయు విధానాన్ని

- Package ను తయారు చేయుటకు ముందుగా package అనే keyword ను ఉపయోగిస్తాము.
- Java source file లో ఇది మొదటి statement అవుతుంది. ఇందులోని మనము వివిధ class లను తయారు చేస్తాము.

Ex : package first package;

Public class first-class

{

-----

}

ముందుగా ఒక sub directory ని తయారు చేసి అందులో source file ను నిల్వ చేస్తాము. తరువాత class definition లన్ని notepad లో తయారు చేస్తాము. అయితే వీటిన్నంటిని public mode లో declare చేస్తాము.

**For Example:-** Public class class A

{

Pubic void display ()

{

System.out.println (“class A”);

}

}

3. ఒక తరగతి అమలు ద్వారా ఇంటర్ఫేస్ కు అమలు ఎలా అందించాలి ?

**How to provide implementation to the interface by the implementation a class**

**జ. Implementation to the interface by the implementation a class:**

జావాలో, పద్ధతులను అమలు చేయడానికి మనము implement keyword ని ఉపయోగిస్తున్నాము, అవి సంబంధిత interfaceలో ప్రకటించబడింది. అంటే:

Class < class name > implement <interface name >

{

Body of the class name

}

1. Interface అనేది మెథడ్ ప్రోటోటైప్ ల సెసిఫికేషన్. యొక్క అన్ని పద్ధతులు
2. ఇంటర్ఫేస్ పబ్లిక్ మరియు abstractమైనది.

3. Interface పద్ధతులు పబ్లిక్ గా ఉంటాయి, ఎందుకంటే అవి మూడవ పక్షానికి అందుబాటులో ఉండాలి
4. అమలును అందించడానికి vendors . ఎందుకంటే అవి abstract మైనవి
5. అమలు మూడవ పక్షంగా మిగిలిపోయింది.
6. interface లో వేరియబుల్స్ ఉండవచ్చు, అవి పబ్లిక్ స్టాటిక్ మరియు డిఫాల్ట్ గా ఫైనల్.
7. దీని అర్థం interface యొక్క అన్ని వేరియబుల్స్ స్థిరాంకాలు.
8. interface లోని పద్ధతులు ఏవీ ఫ్రైవేట్, రక్షిత లేదా స్థిరమైనవి కావు.
9. ఇంటర్ఫేస్ యొక్క అన్ని పద్ధతులు దాని అమలు తరగతులలో అమలు చేయబడాలి
10. ఏదైనా పద్ధతి అమలు చేయకపోతే , అమలు class ఉండాలి
11. "abstract" గా ప్రకటించబడింది.
12. Interface సూచన దాని అమలు తరగతుల వస్తువులను సూచించవచ్చు.
13. Interface వ్రాతపూర్వకంగా ఉన్నప్పుడు , ఏదైనా మూడవ పక్ష విక్రేత అమలును అందించగలరు దాని తరగతులు.
14. ఒక interface మరొక interface ను అమలు చేయదు, విస్తరించగలదు.
15. ఒక class బహుళ interface లను అమలు చేయగలదు (విస్తరించడం కాదు).

#### 4. Abstract class మరియు Interface మధ్య తేడాలను తెలుపండి?

What are the differences between Abstract class and Interface ?

Abstract class	Interface
1. Abstract class లో అన్ని objects ఒకే లక్షణాన్ని కలిగి ఉంటాయి.	1. Interface లో ఉన్న అన్ని objects ఒకే లక్షణాన్ని కలిగి ఉంటాయి.
2. Abstract class లో తయారు చేసిన program వివిధ sub class ను కలిగి ఉంటుంది.	2. Interface లో తయారు చేసిన program తన యొక్క పనిని వేరే వాళ్ళకు అప్పజెప్పుతుంది.
3. Abstract method కలిగి ఉండి వివిధ కొత్త method లను తయారు చేస్తుంది.	3. ముందుగా తయారుచేసిన Abstract method ను ఉపయోగిస్తుంది.
4. constants మరియు variables ను కలిగి ఉంటుంది.	4. కేవలం constants ను మాత్రమే ఉపయోగిస్తుంది.
5. Abstract class ను తయారు చేయుటకు Abstract అనే keyword ని ఉపయోగిస్తుంది.	5. Interface తయారు చేయుటకు interface అనే keyword ని ఉపయోగిస్తుంది.

## 5. ఇంటర్ఫేస్ వల్ల కలిగే లాభాలు,నష్టాలు ఏమిటి ?

### What are the advantages and disadvantages of interface

జ. Interface అనేది ఒక రకమైన తరగతి, ఇది ఒక వ్యత్యాసంతో పద్ధతులు మరియు వేరియంట్లను కలిగి ఉంటుంది. తేడా ఏమిటి అంటే, Interface కేవలం అమూర్త పద్ధతులు మరియు తుది ఖాళీలను మాత్రమే నిర్వచిస్తుంది అంటే ఇంటర్ఫేస్లు ఈ పద్ధతులను అమలు చేయడానికి ఎటువంటి సంకేతాన్ని పేర్కొనవు మరియు డేటా ఫీల్డ్ లు కేవలం స్థిరాంకాలు మాత్రమే కలిగి ఉంటాయి. అందువలన, ఈ పద్ధతుల అమలుకు నియమావళిని నిర్వహించడానికి ఒక ఇంటర్ఫేస్ ను అమలు చేసే తరగతి యొక్క బాధ్యత.

#### లాభాలు:

1. Interface ప్రధానంగా బహుస్వరూప ప్రవర్తనను అందించడానికి ఉపయోగించబడతాయి .
2. సంక్లిష్టమైన డిజైన్లు చేదించడానికి మరియు మధ్య ఆధారితాలు క్లియర్ చేయడానికి Interface ల విధి
3. Interface ద్వారా మనం జావలో బహుళ సొమ్మును అమలు చేయవచ్చు.
4. సంక్లిష్టమైన డిజైన్లను చేదించడానికి మరియు వస్తువు మధ్య ఆధారిత క్లియర్ చేయడానికి ఇంటర్ఫేస్ ల విధి .
5. Interface లు మీ అప్లికేషన్ ని వదులుగా మారుస్తుంది.

#### నష్టాలు:

1. జావా Interface నెమ్మదిగా మరియు ఇతర వాటి కంటే ఎక్కువ పరమితంగా ఉంటాయి.
2. Interface ని అనేకసార్లు ఉపయోగించాలి, వాటిని కలిగి ఉండటం వల్ల ఎలాంటి ఉపయోగం ఉండదు .
3. జావా Interface నెమ్మదిగా మరియు ఇతర వాటి కంటే ఎక్కువ పరిమితంగా ఉంటాయి.
4. Interface ను అనేకసార్లు ఉపయోగించాలి, వాటిని కలిగి ఉండటం వల్ల ఉపయోగం ఉండదు.

## UNIT-VI

### ఎక్స్ ప్షనల్ హ్యాండింగ్

### EXCEPTION HANDLING

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

1. Debugging అనగానేమి ? What is Debugging?

జ. Debugging:

- Computer ప్రోగ్రాంలో ఉన్న తప్పులను గుర్తించి, వాటికీ సరి చేసే ప్రక్రియనే debugging అంటారు.

2. Errors యొక్క రకాలు ? Write the Types of errors.

జ. Error: Java program లో ముఖ్యముగా మూడు రకాల errors ను కలిగి ఉంటుంది. అవి

1. Syntax errors (Compile errors)
2. Runtime errors
3. Logical errors

3 . Exception అనగానేమి ? Define exception.

జ. Exception:

- జావా ప్రోగ్రాంను రన్ చేస్తున్నప్పుడు వచ్చే errors ను తొలగించటానికి మనం వ్రాసే Conditions ను exception లు అంటారు.

4. Throws clause గురించి వివరించండి? Write about throws clause.

జ. Exception:

- Exception handling చేయునప్పుడు “throw” clause ను కనుక ఉపయోగిస్తే try మరియు catch block లను ఉపయోగించవలసిన అవసరం లేదు.

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

5. Throw clause అనగా నేమి? What is Throw clause ?

జ. Throw clause:

- Throw clause ని ఉపయోగించడం ద్వారా, మేము మినహాయింపు object ను సంబంధిత Catch block తీసివేయవచ్చు.
- కాబట్టి throw statement సాధారణంగా try block లలో మాత్రమే ఉంచబడతాయి.

Example :- Throw new IO Exception (“simple”);



6. దోషం మరియు మినహాయింపు మధ్య తేడాలు ఏమిటి?

Write difference between error and exception.

Error	Exception
1. Error అనేది solve చేయుటకు వీలు కానీ ఒక సమస్య. దానిని solve చేయలేము.	1. Exception అనునది ఒక runtime error అనగా solve చేయుటకు వీలు పడే సమస్య. దీనిని మనము catch చేయుటకు try చేయగలము.
2. Error లో ఉండే runtime exceptions ని మరియు వాటి యొక్క subclass ని Unchecked exceptions అని అంటారు.	2. Exception classes లో ఉండేవన్నీ checked exceptions అని అంటారు.

7. Unchecked exceptions అనగానేమి? Write about Unchecked exceptions.

జ. Unchecked exception:

- Compile time అప్పుడు verify చేయని exceptions ని Unchecked exceptions అని అంటారు. ఈ exceptions అనునవి program తప్పుగా వ్రాయడం వలన ఇవి కనపడతాయి.
- వీటిని compiler ఎటు వంటి తప్పులు చూపలేదు కానీ solution తప్పుగా చూపుతుంది. వీటిని jav a.long.Exception class నుండి వీటిని inherite చేస్తారు. ఇవి సాధారణంగా ప్రోగ్రాం ను రన్ చేసేటప్పుడు వచ్చే డేటా errors. arithmetic overflow, devide by zero మొదలైనవి.  
ఈ క్రింది కొన్ని unchecked exception లు ఇవ్వబడినది.
  - ✓ index out of bounds exception
  - ✓ array index out of bounds exception
  - ✓ class cast exception
  - ✓ arithmetic exception
  - ✓ null pointer exception
  - ✓ Illegal state exception
  - ✓ security exception

1. వివిధ రకాల ఎర్రర్స్ గురించి వ్రాయండి ? Explain the types of errors in java.

జ. Errors వల్ల ప్రోగ్రాం తప్ప results ఇచ్చే అవకాశం ఉంది. కంప్యూటర్ భాషలో ఎర్రర్స్ ను bugs (బగ్స్) అని పిలుస్తారు. ప్రోగ్రాంను అభివృద్ధి చేసే క్రమంలోనూ, టైప్ చేసే క్రమంలోనూ ఎర్రర్స్ ఎక్కువగా వస్తాయి. ఎర్రర్స్ వల్ల ప్రోగ్రాం పూర్తిగా ఎగ్జిక్యూట్ కాకపోవచ్చు. తప్పగా టెట్టుట్ ఇవ్వవచ్చును.

ఎర్రర్స్ ను ముఖ్యంగా మూడు రకాలుగా విభజించవచ్చును . అవి

- Compile –time errors
- Run-time error
- Logical error

**Compile–time errors:**

- Java language లో ఉన్న syntax errors అన్ని కంపైల్ టైంలో చూపబడతాయి. ఈ ఎర్రర్స్ ను కంపైల్ టైం ఎర్రర్స్ అంటారు. Compile error ఉండగా class file తయారు చేయదు. ఈ errors మొత్తమును clear చేసిన తరువాత java virtual file అనునది తయారు అవుతుంది.

ఉదాహరణ:-

- ✓ సెమికోలన్ లను మరిచిపోవటం.
- ✓ బ్రేకేట్లను సరిగా పెట్టకపోవటం.
- ✓ డబుల్ కోట్ లు , సింగిల్ కోట్ లను ముగించక పోవటం .
- ✓ object లను రిఫరెన్స్ చేసునప్పుడు మొదలైనవి.

**Run–time errors:**

- Compile time error clear అయిపోయినా class file అవ్వదు , ప్రోగ్రాం రన్ కాదు. మరియు result రాదు. Logical గా కనుక తప్పులు ఉంటే stack over flow అనే error ను చూపిస్తుంది.

ఉదాహరణ

- ✓ ఒక integar విలువను సున్నాతో భాగించటం
- ✓ array యొక్క boundary ని ధాటి విలువలను యాక్సెస్ చేయడం
- ✓ సరైన పారామీటర్ లను పంపక పోవటం

- ✓ array లకు బుణాత్మక విలవను సైజుగా ఇవ్వటం
- ✓ సరికాని string లను నెంబర్లు గా మార్చాలని ప్రయత్నించడం
- ✓ ఉదా :- int x = 5,y =0 ;
- ✓ system.out.ప్రిన్ట్(x/y);
- ✓ పై ఉదహరణ divide by zero ను చూపిస్తుంది.

### Logical error:

- logical errors అనునది program ను run చేసినప్పుడు మాత్రమే గుర్తించగలము. Logical errors అనునవి ఇచ్చిన task లో formula లో తప్పు ఉంటే JVM గుర్తిస్తుంది. Logical errors యొక్క పూర్తి భాద్యత programmer పై ఉంటుంది.

## 2. Exception handling అనగానేమి? java లో దీనిని ఎలా ఉపయోగిస్తారు?

### How to handle the exception in java.

#### జ. Exception handling:

- Java లో exceptions వచ్చినప్పుడు వాటిని handle చేయాటానికి వ్రాసే ప్రోగ్రాంలనే exception handling అంటారు.

Exception handling ద్వారా ప్రోగ్రాంలోని ఎర్రర్ లను runtime లో గుర్తించి exceptional circumstance కు తెలియ చేస్తుంది ఇది error కు సంబంధించిన codeను run చేస్తుంది.

- ✓ error లను గుర్తించటం
- ✓ error వచ్చిందని చెప్పటం (throw the exception )
- ✓ error యొక్క సమాచారాన్ని తెలుసుకోవటం (catch the exception )
- ✓ సరియైన సమాచారం ద్వారా తెల్పటం (handling the exception)

Error handling అనునది ముఖ్యంగా రెండు రకాలు అవి

- 1) Error గుర్తించు throw exception చేయటం
- 2) సరియైన సమాచారం గుర్తించి catch exception చేయటం

A try block looks like as Follows

```
Try {
    ----
}
```

Catch(Exception class ref)

```
{ --  
}
```

Finally()

```
{ ---  
}
```

3. Exception యొక్క రకములను తెలుపండి. Explain types of exceptions in Java.

జ. Exception రెండు రకములు

1. Checked Exceptions

2. Unchecked Exceptions

1. **Checked Exceptions:** ప్రోగ్రాం ను compile చేస్తున్నప్పుడు వచ్చే exception లను checked exception అంటారు. ప్రోగ్రాంను compile చేస్తున్నప్పుడు compiler ప్రోగ్రాంలో ఏవైన exception handlers ఉన్నాయో లేవో చెక్ చేస్తుంది. ఈ క్రింద కొన్ని సాధారణముగా వచ్చే checked exception లు ఇవ్వబడ్డాయి. అవి

- Class Not Found Exception.
- IOException
- EOF Exception
- File Not Found Exception
- No Such Method Exception

2. **Unchecked Exceptions:** ప్రోగ్రాం ను Run చేస్తున్నప్పుడు వచ్చే exception లనే unchecked exception అంటారు. వీటిని Java.Lang.Exception class నుండి inherit చేస్తారు. ఇవి సాధారణముగా ప్రోగ్రాంను run చేస్తున్నప్పుడు వచ్చే errors arithmetic overflow, divide by zero మొదలైనవి. ఈ క్రింద కొన్ని Unchecked exception లు ఇవ్వబడ్డాయి. అవి

- Arithmetic Exception
- Illegal Argument Exception
- Number Format Exception
- Index Out Of Bound Exception

- Array Index Out Of Bounds Exception
- String Index Out Of Bounds Exception
- Null Pointer Exception.

4. జావాలో మినహాయింపులుగా నిర్మించబడినది ఏవి ?

What are the built-in exception in java.

జ. Built-in exception in Java: బిల్ట్ ఇన్ మినహాయిలు ఇవి ఇప్పటికే జావాలో లభ్యం అవుతున్నాయి. నిర్దిష్ట దోష పరిస్థితులను వివరించడానికి ఇవి తగినవి. దిగువ పేర్కొన్నవి కొన్ని బిల్ట్ ఇన్ మినహాయింపులు

మినహాయింపు తరగతి (exception)	అర్థం (Meaning)
1.అర్థమేటిక్ మినహాయింపు	1.అర్థమేటిక్ ఆపరేషన్ లో అసాధారణ పరిస్థితి లో తలెత్తినప్పుడు వినరండి
2.బౌండ్ మినహాయింపు నుంచి array ఇండెక్స్ అవుట్	2.ఒక array ఒక అక్రమ ఇండెక్స్ తో ప్రాప్తి చేయబడింది అని సూచించడానికి వినరవేయబడింది.
3.class లో మినహాయింపు కనుగొనబడలేదు	3.ఒక class యొక్క నిర్వచనం కనుగోనలేని విధంగా యాక్సెస్ చేసుకోవడానికి ప్రయత్నించినప్పుడు ఈ మినహాయింపు లేవనేత్తబడినది.
4.file మినహాయింపు కనుగొనబడలేదు	4.file యాక్సెస్ చేసుకోలేనప్పుడు లేదా అంతరాయం కలిగించినప్పుడు లేవ నేత్తాడం I/O ఆపరేషన్ విఫలమైనాదా లేదా అంతరాయం కలిగినప్పుడు విసిరివేయడం.
5.IO మినహాయింపు	5.దారం వేచి ఉండడం ,నిద్రపోవడం లేదా ఏదైనా ప్రా సెసింగ్ చేయడం మరియు దానికి అంతరాయం కలిగించడం.

6.అంతరాయం మినహాయింపు	6.ఒక class లో విఫలమైన (లేదా వేరియబుల్ పేర్కొనబడనప్పుడు విసిరివేయబడతుంది.
7.అటువంటి file మినహాయింపు లేదు	7.కనుగోనలేని ఒక పద్ధతిని ప్రాప్తి చేస్తున్నప్పుడు విసిరివేయబడ్డాయి.
8.అటువంటి పద్ధతి మినహాయింపు లేదు	8.శూన్య వస్తువు యొక్క సభ్యుని రిఫర్ చేసేటప్పుడు లేవనెత్తడం శూన్య ప్రాతినిధ్యం ఏమి.
9.శూన్య పాయింట్ మినహాయింపు	9.ఒక పద్ధతి string ను సంఖ్యా ఆకృతిగా మార్చలేనప్పుడు పెంచింది.
10.నెంబర్ ఫార్మాట్ మినహాయింపు	10.ఇది రన్ టైం సమయంలో సంభవించే మినహాయింపు ను సూచిస్తుంది.
11.రన్ టైం మినహాయింపు	11.సూచించడానికి string class విధానాల ద్వారా విసిరివేయబడ్డారు మరియు
12.string ఇండెక్స్ బఫర్ మినహాయింపు	12.ఇండెక్స్ అనేది నెగటివ్ (-) లేదా string యొక్క సైజు కంటే ఎక్కువగా ఉంటుంది.

**5. User Defined Exceptionలను ఏ విధంగా Throw చేస్తారు (లేదా ) గురించి “Throw” వివరించండి. Or Explain user defined exceptions.**

**జ. User defined exception :**

జావాలో చాలా రకాల exception ఇన్ఫర్మల్ గానే లభిస్తాయి. కానీ కొన్ని exceptionలు మనం create చేసుకునే అవకాశం ఉంది. దీని కొసం user defined exceptionలు ఉపయోగపడతాయి.

user defined exception లు తయారుచేయాలంటే ఈ క్రింది సూచనలు పాటించాలి.

- ✓ వీటిని exception క్లాసుకు sub class గానే తయారుచేయాలి.
- ✓ To string ( ) Methodను overridden చేయాలి.

**Throw:** Throw"అనే keyword exception handling చేయడానికి ఉపయోగపడుతుంది. దీనిని ఎర్రర్స్ ను manuallyగా చేయడానికి ఉపయోగిస్తారు.

Example:-

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

6. Error కి Exception కి మధ్య తేడాలను తెలుపండి.

What are the differences between error and exceptions ?

Error	Exception
1.Compile time లో వచ్చే తప్పులను errors అని అంటారు	1.Runtime లో వచ్చే తప్పులను exceptions అని అంటారు.
2. Java లో errors కి తొలగించుట కష్టము.	2. Java లో మనము runtime లో solve చేయగలము
3.Java లో error ని java.lang.error	3.Javaలోexceptions ని java.lang.exception
4. Java లో errors కి compiler మాత్రమే handle చేస్తుంది.	4. Exceptions ని JVM లేదా User handle చేసి solve చేస్తాడు.
5. User తన సొంత error classes ను define చేయలేడు.	5.User తన సొంత exception classes ను తయారు చేయగలడు.

6. Errors మూడు రకాలు. అవి 1. Compile errors 2. Runtime errors 3. Logical errors	6. Exceptions రెండు రకములు. అవి 1. User defined exceptions 2. System defined exception
7. Error అనేవి non-recoverable	7. Exceptions అనునవి recoverable exception

7. దోషానికి, మినహాయింపుకు మధ్య తేడాలు ఏమిటి?

**What are the differences between error and exceptions?**

జ. దోషానికి, మినహాయింపుకు మధ్య తేడాలు

దోషం (Error )	మినహాయింపు (Exception )
1. In check చేయని type లో దోషాలు	1. చేయడం ద్వారా మినహాయింపుల మినహాయింపు రెండు check చేయబడి, check చేయని రకం
2. దోషాల నుంచి కోలుకోవడం అసాధ్యం	2. వీటిని try, catch blocks ల ద్వారా handling నుంచి recovery సాధ్యమవుతుంది.
3. జావాలో దోషాలు అనేవి java .lang దోషం	3. జావాలో మినహాయింపులు రకం java .lang మినహాయింపులు
4. లోపాలు ఎక్కువగా application నడుస్తున్న పర్యవరణం వలన కలుగుతాయి	4. మినహాయింపులు ప్రధానంగా application దానంతట అదే కలుగుతుంది.
ఉదాహరణలు 1. Java .lang. స్టాక్ ఒవర్ ఫ్లో దోషం 2, java.lang . మెమరీ దోషం	ఉదాహరణలు 1. తనిఖీ మినహాయింపులు sql మినహాయింపు, IO మినహాయింపు 2. on check మినహాయింపులు, array index out of bound మినహాయింపు, class cast మినహాయింపు, null point మినహాయింపు



## UNIT -VII

### Threads (థ్రెడ్స్)

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

#### 1. Threads అనగానేమి? What is a thread?

##### జ. Thread:

Threads అనునది ఒక task లేదా flow of execution. వీటిని టైం షేరింగ్ ద్వారా run అయ్యే విధంగా తయారు చేస్తారు. thread అనునవి ఒకసారి రన్ అవుతు. ఇవి parallel పద్ధతి ద్వారా Run అవుతాయి. ప్రతి thread, java.lang.Thread class తో తయారు అయి run అవుతాయి.

#### 2. Multitasking అనగానేమి? What is Multitasking ?

##### జ. Multitasking:

ప్రతి operating system లో ఒకటి కన్న ఎక్కువ tasks, computer ప్రాసెసర్ ద్వారా ఒకేసారి ఎగ్జిక్యూట్ చేయబడే విధానాన్ని multitasking అంటారు. multitasking ఎక్కువగా చేయగలిగితే సిస్టం యొక్క performance ఎక్కువగా ఉంటుంది.

#### 3. Multithreading అనగానేమి? What is Multithreading?

##### జ. Multithreading:

Java లో multi threading ప్రోగ్రాం in-built గానే ఉంటుంది. ఒకటికన్నా ఎక్కువ thread లు ఒకేసారి రన్ అయ్యే విధానాన్ని multithreading అంటారు. దీని ద్వారా ప్రాసెసర్ ను ఎక్కువ సమర్థవంతంగా ఉపయోగించుకొనవచ్చును.

#### 4. Stopping a thread అనగానేమి? Write about Stopping a thread .

##### జ. Stopping a thread

ఎగ్జిక్యూట్ అవుతున్న thread ను ఆపాలంటే Stop() method ని ఉపయోగిస్తాము. Stop() అని statement thread ను ఆపి dead stateకు తీసుకువస్తుంది. Like T1.Stop();

#### 5. Block a thread అనగానేమి? Write about Block a thread.

##### జ. Block a thread:

ఎగ్జిక్యూట్ అవుతున్న thread ను కొంత సమయం కొరకు ఆపడాన్ని blocking అని అంటారు. Block చేయుట కొరకు మనము కొన్ని methods ను ఉపయోగిస్తాము అవి sleep (), suspend () మరియు wait () method లు ఉపయోగపడతాయి.

1. Thread ని ఎలా క్రియేట్ చెయ్యాలి ఉదాహరణతో సహా వివరించండి ?

**Explain creating a thread with example.**

**జ. Creating a thread with example:**

Threads ను objects రూపంలో తయారుచేస్తారు. వీటిని run() methods ద్వారా తయారుచేస్తారు. run() method అనేది thread కు చాలా ముఖ్యమైనది.

```
Public void run()  
{  
    Statements for implemented thread  
}
```

సాధారణంగా thread లను రెండు రకాలుగా తయారుచేస్తారు. అవి:-

1. By Extending Thread class
2. By Implementing Runnable Interface

**1. By Extending Thread Class :**

- ఈ పద్ధతి లో ఒక thread క్లాస్ ను Extend చెయ్యడం ద్వారా మరియొక క్లాసు ను తయారుచేయవచ్చు. దీనిలో run() method ను override చెయ్యవలెను.
- ఈ పద్ధతిలో threadలో ఉన్న అన్ని method లను డైరెక్టుగా యాక్సెస్ చేసుకొనవచ్చును. ఈ method కాల్ చేయవలసినట్టే స్టార్ట్ () methodను ఉపయోగించవలెను.

**2. By implementing Runnable Interface :**

- ఈ పద్ధతి runnable interface ద్వారా thread లను క్రియేట్ చేస్తారు. దీనిలో ఒకే Run() methodను తయారు చేస్తారు.
- ఏ క్లాసు ద్వారా thread రన్ చేస్తేరో ఆ ఆబ్జెక్ట్ ను క్రియేట్ చేయాలి. తయారు చేసిన థ్రెడ్ ను start() method ద్వారా కాల్ చేస్తారు.

## 2. పొడిగించే దూరాన్ని (Extending Thread) ఉదాహరణతో సహా వివరించండి?

Explain extending thread with example .

### జ. Extending thread with example:

తరగతి జవాను విస్తరించడం ద్వారా తయారుచేయవచ్చు. దీనిలో దిగువ దశలు తరగతి పొడిగించే దారం పలె డిక్లేర్ చెయ్యాలి.

declare the class as extending thread

```
{  
    ----  
}
```

Thread అమలు చేసే కోడ్ క్రమాన్ని అమలు చెయ్యడానికి బాధ్యత వహించే run()

విధానాన్ని అమలు చెయ్యండి.

```
Public void run()
```

```
{  
    -----  
}
```

### For 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

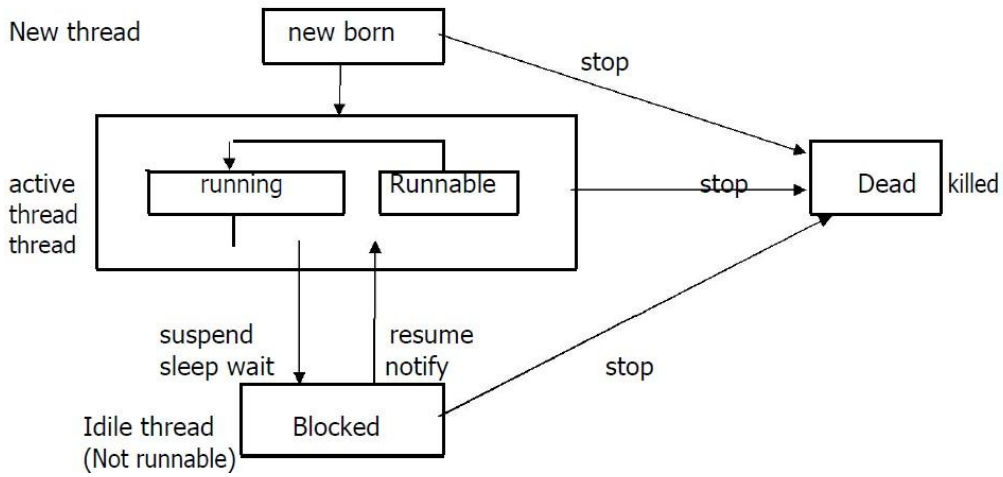
```
{
    Public static void main (string args[])
    {
        New A ().start ();
        New B ().star ();
        New C ().start();
    }
}
```

### 3. Thread యొక్క జీవిత చక్రాన్ని తెల్పండి? Explain the life cycle of the thread .

#### జ. The life cycle of the thread:

Thread యొక్క life time ను అనుసరించి thread వివిధ states లో ఉంటుంది. అవి

1. New born state
2. Runnable state
3. Running state
4. Blocked state
5. Dead state



#### 1. New born state:

క్రొత్తగా thread ను create చేసినప్పుడు అది new born state లో ఉంటుంది. దీనిని state( ) method ద్వారా చేయవచ్చు. ఈ స్థితిని ఆపివేయాలంటే Stop( ) method ను ఉపయోగిస్తారు.

#### 2. Runnable State:

Runnable State అంటే thread అనునది Run చేయటానికి అనువుగా ఉంది అని అర్థము. అంటే ప్రాసెసర్ కోసం wait చేస్తుంది అని అర్థము.

#### 3. Running state:

Running state అంటే thread ప్రాసెసర్ చేత ఎగ్జిక్యూట్ చేయబడుతుంది .అని అర్థము. దీనికోసం Run ( )method ను ఉపయోగిస్తారు .

#### 4. Blocked state

Blocked state అంటే ఒక thread running state నుండి Runnable stateకు మారింది అని అర్థము .దీని కోసంSuspend( ), Sleep( ) లేదా Waite( ) Methodలను ఉపయోగిస్తారు .

#### 5. Dead State:

Thread యొక్క life cycle అనునది dead stage తో ముగుస్తుంది. Run state తరువాత dead stage వస్తుంది. ఒకవేళ thread ను మధ్యలోనే ఎగ్జిక్యూట్ కాకుండా ఆపినా state వచ్చే అవకాశం ఉంది. దీనికోసం stop method ను ఉపయోగిస్తారు.

#### 4.మల్టీత్రెడింగ్, మల్టీ టాస్కింగ్ మధ్య తేడాలు ఏమిటి?

What are the differences between multithreading and multitasking ?

మల్టీ త్రెడింగ్ (multi threading)	మల్టీ టాస్కింగ్ (multi tasking)
1.ఇది ప్రోగ్రామింగ్ కాన్సెప్ట్,ఇది ఒక ప్రోగ్రాం యొక్క ప్రాసెస్ రెండు లేదా అంతకంటే ఎక్కువ సుబ్ ప్రోగ్రామ్ లు లేదా థ్రెడ్ లు సమాంతరంగా ఒకే సమయంలో అమలు చెయ్యబడతాయి	1.ఇది ఒక ఆపరేటింగ్ system కాన్సెప్ట్, దీనిలో ఏక కాలంలో అనేక టాస్క్ లు నిర్వహించబడతాయి
2.ఏక కాలంలో single కార్యక్రమం యొక్క బహుళ భాగాలను అమలు చేయడానికి ఇది మద్దతు ఇస్తోంది	2. ఏక కాలంలో పలు కార్యక్రమాల అమలుకు మద్దతు
3.ప్రాసెసర్ ఒక ప్రోగ్రామ్ యొక్క విభిన్న భాగాలు లేదా త్రెడ్ లమధ్య మారాల్సి ఉంటుంది	3.ప్రాసెసర్ వివిధ ప్రోగ్రామ్ ల మధ్య మారాల్సి ఉంటుంది
4.ఇది అత్యంత సమర్థ వంతమైనది	4.మల్టీ త్రెడ్స్ తో పోలిస్తే ఇది తక్కువ సమర్థవంతంగా ఉంటుంది.

5.దారం ఒక చిన్న యూనిట్ అది బహు త్రెడ్స్	5.ఒక ప్రోగ్రామ్ లేదా ప్రాసెస్ అనేది మల్టీ టాస్కింగ్ వాతావరణం లో అతి చిన్న యూనిట్
6.సమర్థవంతమయిన కార్యక్రమాలకు అభివృద్ధి చేయడంలో ఇది సహాయపడుతుంది.	6.ఇది డెవెలప్ మెంట్ సమర్థవంతమయిన ఆపరేటింగ్ system కు దోహదపడుతుంది.
7.ఇది ఖర్చు అవుతుంది సందర్భోత్కంగా మారే సందర్భంలో ప్రభావవంతంగా ఉంటుంది.	7. సందర్భం మారే సందర్భంలో ఇది ఖరీదైనది.

### 5. దారాలను (Threads) సవరించడం? How to manipulating the threads ?

#### జ. To manipulating the threads:

##### 1. నిద్ర(): Sleep( )

➤ కొంత సమయం మరియు CPU కోసం దారం బ్లాక్ చెయ్యవచ్చు

##### 2. సస్పెండ్(): Suspend()

➤ Thread త్రెడ్ క్లాస్ కు ఒక పద్ధతి సస్పెండ్() త్రెడ్ తాత్కాలికంగా ఆపివేయడానికి () ఒక పద్ధతి రెసూమ్ () దానిని రీస్టార్ట్ చెయ్యాలి.

##### 3. రెసూమ్మును(): Resume()

➤ ఈ పద్ధతిని తాత్కాలికంగా నిలిపివేయడం కాకుండా వేరే దారం ద్వారా పిలవాలి.

##### 4. Wait(), notify() and notify all():

➤ Running thread అల కాల్ చేసినప్పుడు ( అది వేచి ఉన్న నిర్దిష్ట ఆబ్జెక్ట్ కొరకు వేచి ఉండే స్థితి లో ప్రవేశిస్తుంది.

##### 5. దిగుబడి (): Yield()

➤ ఒక త్రెడ్ యొక్క కొన్ని CPU ఇంటన్సివ్ ఆపరేషన్ లు ఇతర త్రెడ్ లు execute కాకుండా నిరోధించవచ్చు.

# UNIT-VIII

## THE JAVA APPLET

స్వల్ప సమాధాన ప్రశ్నలు మరియు జవాబులు

### 1. Applet అనగానేమి? What is Applet?

జ. Applet:

జావాలో html పేజీలలో embedded చేసిన ప్రోగ్రాములనే applet లు అంటారు. Applet అనునది జావా ప్రోగ్రాంలు. వీటిని వెబ్ బ్రౌజర్ ద్వారా execute చేస్తారు. వీటిని బైట్ పేజీలను ఎక్కువ dynamicగా మరియు entertaining గా చేయటానికి ఉపయోగిస్తారు. Applet = java byte code + HTML page

### 2. Appletలను క్రియేట్ చేసే పద్ధతిని వివరించండి?How can we Create an applet?

జ. Appletలను క్రియేట్ చేసే పద్ధతి

జావా ప్రోగ్రాంలో యాపలెట్లను ఉపయోగించాలంటే మొదట జావా కంప్యూటర్ లో install చేయాలి. దీనిలో తప్పనిసరిగా java applet viewer లేదా బ్రౌజర్ ప్రోగ్రాంను install చెయ్యాలి. ఒక java appletను తయారుచేసి test చేసి రన్ చేయాలంటే ఈ క్రింది క్రమాన్ని పాటించాలి.

- 1: మొదటగా java applet code వ్రాయాలి. (.java file)
- 2: తరువాత ఎగ్జిక్యూట్ చేయబడే applet ఫైల్ ను తయారుచెయ్యాలి. (.class file)
- 3: HTML tagను ఉపయోగించి వెబ్ పేజీని తయారుచెయ్యాలి.
- 4: <APPLET> tagను తయారు చేయాలి.
- 5: < APPLET> tagను wab pageలో ఉంచాలి.
- 6: HTML ఫైల్ ను క్రియేట్ చేయాలి.
- 7: applet code ను test చేయాలి.
- 8: బ్రౌజర్ ప్రోగ్రాం లేదా applet viewer ద్వారా applet ప్రోగ్రాంను రన్ చేయాలి.

### 3. Applet సంకల్ప మరియు తొలగింపు అనగా నేమి?

Applet initialization and termination ?

జ. Public void int ():

- ఈ పద్ధతి బ్రౌజర్ ద్వారా పిలవబడే మొదటి పద్ధతి మరియు ఇది ఒకటి మాత్రమే అమలు చేయబడుతుంది.



- కాబట్టి, కాంపోనెంట్లను సృష్టించడం మరియు డ్రెడ్లను సృష్టించడం మొదలైన ఏవైనా వేరియబుల్లను ప్రారంభించేందుకు ప్రోగ్రామర్ ఈ పద్ధతిని ఉపయోగించవచ్చు. దీనిని initialization అని అంటారు.

**Public void stop ():**

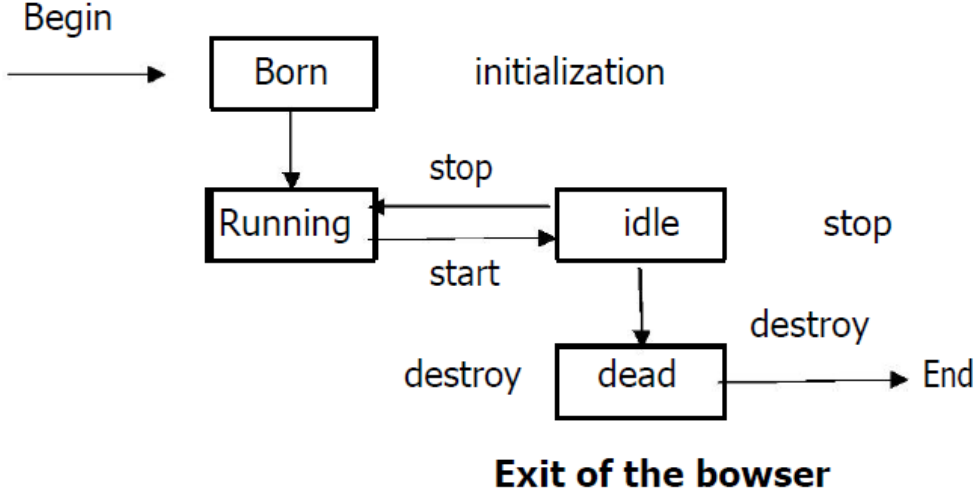
- ఆప్లెట్ పబ్లిక్ శూన్య నాశనం () నిలిపివేయబడినప్పుడు ఈ పద్ధతిని బ్రౌజర్ అంటారు.
- మెమరీనుండి ఆప్లెట్ నిలిపివేయబడినప్పుడు ఈ పద్ధతిని పిలుస్తారు. దీనిని termination అని అంటారు.

**4. ఏవైనా 4 HTML applet tag లు రాయండి? Write any 4 HTML applet tags?**

**జ. HTML applet tag:** HTML మద్దతు వెబ్ పేజీల ప్రదర్శనకు శైలిని మరియు ఆకృతిని నియంత్రించడానికి ఉపయోగించే ట్యాగ్ లు తక్కువ సంఖ్యలో ఉంటాయి.

ట్యాగ్ (TAG)	ఫంక్షన్ (FUNCTION)
1.<HTML></HTML>	1.HTML file యొక్క ప్రారంభం మరియు చివరను తెలియజేస్తుంది.
2.<HEAD.</EAD>	2.ఈ ట్యాగ్ లో వెబ్ పేజీ, సాధారణంగా ట్యాగ్ తో సహా ఉండవచ్చు.
3.<TITLE></TITLE>	3.దీనిలో ఉన్న టెక్స్ట్ బ్రౌజర్ యొక్క టైటిల్ బార్ లో కనిపిస్తుంది.
4.<BODY></BODY>	4.వెబ్ పుట ప్రధాన పాఠాన్ని ఈ ట్యాగ్ కలిగి ఉంది. ఇది ట్యాగ్ ని డిక్లర్ చేసిన ప్రదేశం <APPLET>.</APPLET>
5.<H1>...</H1>	5.డిస్ ప్లే ఉపయోగించిన హెడర్ ట్యాగ్ లు
6.<H6>...</H6>	6.హెడ్డింగ్ <H1> అతి పెద్ద ఫోర్ హెడర్ ని సృష్టిస్తుంది </H1>
7.<CENTER>...</CENTER>	7.పేజి యొక్క సెంటరు వద్ద టెక్స్ట్ కలిగి ఉన్న ప్రదేశాలు
8.<APPLET>...</APPLET>	8.ఇది అప్లెట్ వివరాలను ఇయాట్రిబ్యూట్ గా ప్రకటిస్తుంది

1. Applet యొక్క Life Cycle ను వివరించండి. Explain Applet life cycle?



జ. Java లో ఒక Applet తయారు చేసినప్పటినుండి run చేసి death state కి చేరే వరకు JVM ద్వారా ఎగ్జిక్యూట్ చేయబడుతుంది. ఒక applet ను క్రియేట్ చేసినప్పటి నుండి హార్టి అయ్యే వరకు వివిధ దశలను దాటుతుంది. వీటిని ఈ క్రింది విధంగా చూపావచ్చును.

1.new born state 2.running state 3.idle state 4.dead or destroyed state  
 applet యొక్క జీవితచక్రాన్ని (life cycle) ఈ క్రింది విధంగా చూపవచ్చును.

#### New born state:

ఒక applet అనునది క్రియేట్ అయిన తరువాత అది బ్రౌజర్ లోకి init ( ) మెథడ్ ద్వారా కాలి ఆవుతుంది. init ( ) method applet యొక్క జీవిత చక్రం లో ఒకేసారి వస్తుంది. దీనిలోకి పెరమిటార్ లను PARAM tag ద్వారా html file లోకి పంపుతారు. ఈ స్థితిలో ఏవైనా అబ్జెక్ట్ లు అవసరమైతే create చేస్తారు.

Syntax:

```

Public void init ( )
{
    Statements;
}
    
```

**Running state:** Running state అనునది initialization తరువాత మరియు idle state తరువాత వస్తుంది. దీనిని ఎన్నిసార్లు అయినా కాల్ చేసి applet ను restart చేయవచ్చును. దీనిని override కూడా చేయవచ్చును.

**Syntax:** Public void start ( )  
{  
Statements;  
}

**Idle state:** దీని ద్వారా ఎగ్జిక్యూట్ ఆవుతున్న applet ను టెంపరరీ గా అపవచ్చును. దీనిని stop ( ) method ద్వారా చేయవచ్చును. idle state తర్వాత applet run state లోకి వెళుతుంది. **Syntax:** Public void stop ( )

{  
Statements;  
}

**Dead state:** Applet ను ఎగ్జిక్యూషన్ నుండి బయటకు తీసుకొని రావటానికి destroy function ను invoke చేస్తే అది dead state కు వస్తుంది. ఈ స్థితిలో appletను memory నుండి తొలగిస్తారు. దీనికి allocate చేసిన resources ను కూడా తొలగిస్తారు.

**Syntax:** Public void destroy ( )  
{  
Statements;  
}

**Display state:** Applet ను display state లోకి పంపాలంటే paint లేదా update method ను ఉపయోగిస్తారు. దీనిని applet ద్వారా సమాచారాన్ని స్క్రీన్ మీద display చేయటానికి ఉపయోగిస్తారు. paint ( ) method graphics ను ఆర్గ్యుమెంట్ గా తీసుకొంటుంది.

**Syntax:** Public void paint (Graphics g )  
{  
Statements;  
}

2. జావా లో HTML applet ట్యాగ్ ల గురించి వివరించండి.

**Explain the HTML applet tags in java?**

**A. Designing a web page :-** ఒక వెబ్ పేజిని ప్రాథమికంగా text తో తయారు చేస్తారు మరియు HTML tag లను web browser లేదా ఆ applet viewer ద్వారా అర్థం చేసుకోవచ్చు. ఈ క్రింది మూడు ప్రధాన విభాగాలకు విభజించబడింది.

1. Comment section (it purely option)
- 2 . Head section (it is also option)
3. Body section

ఈ మూడు విభాగాలను కలిగి ఉన్న వెబ్ పుట సారాంశము మరియు HTML tag లను తెరవడం మరియు ముసి వేయడం.

Example :- <HTML>

```
<HEAD> Title tag </HEAD>
<BODY> Applet tag </BODY> </HTML>
```

**1. Comment section :-** ఏ విభాగములో వెబ్ పేజి గురంచి ఉంటుంది.

**2. Head section:-** Starting తో head సెక్షన్ నిర్వచించబడుతుంది <HEAD> tag మరియు ఒక ముగింపు </HEAD> tag.

```
<HEAD > <TITLE> GJC(Boys) Palakonda </TITLE> </HEAD>
```

**3. Body section :-** Head సెక్షన్ తర్వాత body సెక్షన్ వస్తుంది. ఈ విభాగంలో వెబ్ పేజి మరియు దాని ప్రవర్తన గురంచి సమాచారం ఉంటుంది. కనుక మనం body సెక్షన్ గా పిలుస్తాం. కలర్ , లొకేషన్, సౌండ్ మొదలైన body సెక్షన్ లోపల వ్రాయవలసిందే.

```
<BODY>
```

```
<CENTER> <HI> < GJC(Boys) Palakonda / Hi> </CENTER>
```

```
</BODY>
```

```
<APPLET> tag
```

వెబ్ పేజీ యొక్క ప్రధాన విభాగం <APPLET> tagని కలిగి ఉంది, ఇది HTML పేజీలో applet ను చేర్చడానికి ఉపయోగపడుతుంది. ఈ tag ఆ applet యొక్క పేరును 10 జోడించేందుకు సప్లయ్ చేస్తుంది మరియు applet కు ఎంత స్థలం అవసరమో బ్రౌజర్ కి తెలియజేస్తుంది. ఇది క్రింది ఫారమ్ ను కలిగి ఉండాలి.

<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 name1 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”);

ట్యాగ్ (TAG)	ఫంక్షన్ (FUNCTION)
1.<HTML></HTML>	1.HTML file యొక్క ప్రారంభం మరియు చివరను తెలియజేస్తుంది.
2.<HEAD.</EAD>	2.ఈ ట్యాగ్ లో వెబ్ పేజీ, సాధారణంగా ట్యాగ్ తో సహా ఉండవచ్చు.
3.<TITLE></TITLE>	3.దీనిలో ఉన్న టెక్స్ట్ బ్రౌజర్ యొక్క టైటిల్ బార్ లో కనిపిస్తుంది.
4.<BODY></BODY>	4.వెబ్ పుట ప్రధాన పాఠాన్ని ఈ ట్యాగ్ కలిగి ఉంది. ఇది ట్యాగ్ ని డిక్లర్ చేసిన ప్రదేశం <APPLET>.</APPLET>
5.<H1>...</H1>	5.డిస్ ప్లే ఉపయోగించిన హెడర్ ట్యాగ్ లు

6.<H6>...</H6>	6.హెడ్డింగ్ <H1> అతి పెద్ద ఫోర్ హెడర్ ని సృష్టిస్తుంది </H1>
7.<CENTER>...</CENTER>	7.పేజి యొక్క సెంటరు వద్ద టెక్స్ట్ కలిగి ఉన్న ప్రదేశాలు
8.<APPLET>...</APPLET>	8.ఇది అప్లెట్ వివరాలను ఇయాట్రిబ్యూట్ గా ప్రకటిస్తుంది
9.<PARAM>	9.ట్యాగ్ లోపల ఈ ట్యాగ్ ఉంచబడుతుంది <APPLET> మరియు ఇది యూజర్ డెరివోమీటర్ ను హోల్డ్ </APPLET>లో ఉంచింది.
10.<B>...</B>	10.ఈ ట్యాగ్ ల మధ్య టెక్స్ట్ బోల్డ్ టైప్ డిస్ ప్లే అవుతుంది.
11. <P>...</P>	11. Paragraph tag.
12.<HR>	12. Draws horizontal line.
13. <FONT>...</FONT>	13 మనము లక్షణాలను ఉపయోగించి ఈ tag ల మధ్య ఉండే వచనం యొక్క రంగు మరియు పరిమాణాన్ని మార్చవచ్చు.

### 3. Explain writing code and creating simple applet?

#### A) Writing code and creating simple applet:

```

import java. awt.*;
import java. applet.*;
Public class myapp Extends applet {
Public void in it () {
Set back ground (color, yellow);
Public void main (graphics g) {
g. draw string ("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.

```
<!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. The applet opens in the browser or give the command at system prompt as

```
C:/> applet viewer myapps.html
```

4. పరమీటర్లను applet కు పాస్ చేసే ప్రక్రియను రాయండి.

**Write the procedure to passing parameters to applet?**

**A. Passing parameters to applets: -**

Applet tag తో పాటు, ఆ applet <PARAM> కు సమాచారాన్ని పాస్ చేయడానికి మనం tag ని ఉపయోగించవచ్చు. HTML పేజీ నుంచి ఆ applet కు ఉద్యోగి యొక్క పేరు మరియు వేతనాన్ని పాస్ చేయడానికి మనం ఆ applet మరియు PARAM tag లను ఇలా రాయవచ్చు.

```
<APPLET CODE = 'TAX CLASS' WIDTH = 300 HEIGHT = 300>
```

```
<PARAM NAME = "E1" VALUE = "SANKAR RAJU">
```

```
<PARAM EMPLOYEE = "E2" VALUE = "CJL 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 "SANKAR RAJU" and "CJL in CS&E" respectively. parameter యొక్క విలువలను అందుకోవడం కొరకు ఆ applet ఉపయోగిస్తుంది. parameter () విధానం. ఈ విధానం parameter పేరు మరియు దాని విలువలను string గా తిరిగి తీసుకుంటుంది.

5. జావాలో ఉండే లక్షణాల గురించి రాయండి. Write about the attributes in java?

A. Attributes in Java:

Attribute	Meaning
CODE = Applet file name class	ఇది applet పేరును నిర్దేశిస్తుంది లోడ్ చేయబడింది, అంటే పేరు ఇప్పటికే సంకలనం చేయబడింది. ఎక్సిక్యూటబుల్ జావా బైట్‌కి క్లాస్ ఫైల్ applet కోసం కోడ్ నిల్వ చేయబడుతుంది.
CODE BASE = code base – URL	ఇది డైరెక్టరీ యొక్క URLని నిర్దేశిస్తుంది ఏ applet నిల్వ చేయబడుతుంది (ఐచ్ఛికం) అయితే .html మరియు .class ఫైల్‌లు దీనిలో జరుగుతాయి అదే డైరెక్టరీ, ఈ లక్షణం కావచ్చు విస్మరించబడింది.
HEIGHT = pixels WIDTH= pixels	ఈ గుణాలు వెడల్పు మరియు HTML పేజీలో స్థలం యొక్క ఎత్తు.
ALIGN= alignment	ఈ ఐచ్ఛిక లక్షణం ఎక్కడ ఉందో నిర్దేశిస్తుంది applet వంటి పేజీ కనిపిస్తుంది ఎగువ, దిగువ, ఎడమ, కుడి, మధ్య
ALT = alternate-text	నాన్-జావా బ్రౌజర్‌లు ఈ వచనాన్ని ప్రదర్శిస్తాయి
NAME = applet-instance – name	applet పేరు ఐచ్ఛికంగా ఉండవచ్చు పేజీలో ఇతర applet లు ఉండేలా పేర్కొనబడింది ఈ పేజీని సూచించవచ్చు
VSPACE = pixels	ఇది కొన్ని నిలువుగా ఉన్నప్పుడు మాత్రమే ఉపయోగించబడుతుంది అమరిక ALIGNతో పేర్కొనబడింది గుణం. ఇది మొత్తాన్ని నిర్దేశిస్తుంది బ్రౌజర్‌కి నిలువు ఖాళీ స్థలం ఉండాలి పరిసరాన్ని ఆప్లెట్‌కి వదిలివేయండి.
MSPACE=pixels	ఇది మాత్రమే ఉపయోగించబడుతుంది ALIGN ఎడమకు సెట్ చేయబడింది లేదా కుడి. ఇది మొత్తాన్ని నిర్దేశిస్తుంది క్షితిజ సమాంతర ఖాళీ స్థలం



6 . Write a java program to create applet with some background color and foreground color with message.

æ. To create applet with some background 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 <HTML>

```
<APPLET CODE="Appl.Class" HEIGHT=300 WIDTH= 400> </APPLET>
```

</HTML> open the above appl. Html file in the applet viewer as

```
c:/> applet viewer app1.html
```

7. Applet కు పరామితులు ఎలా పాస్ చేయాలి.

**Write about How to passing parameters to applet.**

జ. Applet tag తో పాటు, ఆ applet <PARAM> కు సమాచారాన్ని పాస్ చేయడానికి మనం tag ని ఉపయోగించవచ్చు. HTML పేజీ నుంచి ఆ applet కు ఉద్యోగి యొక్క పేరు మరియు వేతనాన్ని పాస్ చేయడానికి మనం ఆ applet మరియు PARAM tag లను ఇలా రాయవచ్చు. <APPLET CODE = 'TAX CLASS' WIDTH = 300 HEIGHT = 300>

```
Example:- 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="Appl.Class"
HEIGHT=300 WIDTH= 400> </APPLETT>
```

```
<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.
```

8. జావాలో HTML tags గురించి రాయండి. Write about the html tags in java.

A. 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.

ట్యాగ్ (TAG)	ఫంక్షన్ (FUNCTION)
1.<HTML> </HTML>	1.HTML file యొక్క ప్రారంభం మరియు చివరను తెలియజేస్తుంది.
2.<HEAD> </EAD>	2.ఈ ట్యాగ్ లో వెబ్ పేజీ, సాధారణంగా ట్యాగ్ తో సహా ఉండవచ్చు.
3.<TITLE> </TITLE>	3 .దీనిలో ఉన్న టెక్స్ట్ బ్రౌజర్ యొక్క టైటిల్ బార్ లో కనిపిస్తుంది.
4.<BODY></BODY>	4.వెబ్ పుట ప్రధాన పాఠాన్ని ఈ ట్యాగ్ కలిగి ఉంది. ఇది ట్యాగ్ ని డిక్లర్ చేసిన ప్రదేశం <APPLET>.</APPLET>
5.<H1>...</H1>	5.డిస్ ప్లే ఉపయోగించిన హెడర్ ట్యాగ్ లు
6.<H6>...</H6>	6.హెడ్డింగ్ <H1> అతి పెద్ద ఫోర్ హెడర్ ని సృష్టిస్తుంది </H1>
7.<CENTER>...</CENTER>	7.పేజి యొక్క సెంటరు వద్ద టెక్స్ట్ కలిగి ఉన్న ప్రదేశాలు
8.<APPLET>...</APPLET>	8. ఇది అప్లెట్ వివరాలను ఇయాప్రిబ్యూట్ గా ప్రకటిస్తుంది
9.<PARAM>	9. ట్యాగ్ లోపల ఈ ట్యాగ్ ఉంచబడుతుంది <APPLET> మరియు ఇది యూజర్ డెరివోమీటర్ ను హోల్డ్ </APPLET>లో ఉంచింది.
10.<B>...</B>	10. ఈ ట్యాగ్ ల మధ్య టెక్స్ట్ బోల్డ్ టైప్ డిస్ ప్లే అవుతుంది.
11. <P>...</P>	11. Paragraph tag.
12.<HR>	12. Draws horizontal line.
13. <FONT>...</FONT>	13 మనము లక్షణాలను ఉపయోగించి ఈ tag ల మధ్య ఉండే వచనం యొక్క రంగు మరియు పరిమాణాన్ని మార్చవచ్చు.

# Question Bank

## UNIT-I

అబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రామింగ్ & జావా -1

Object Oriented Programming and Java

స్వల్ప సమాధాన ప్రశ్నలు

1. oops అనగా నేమి? Define Oops.
2. జావాను ఎవరు ఎక్కడ అభివృద్ధి చేశారు ? Who invented by Java?
3. అబ్జెక్ట్ ఓరియంటెడ్ ప్రోగ్రాం వాళ్ళ ఒపయోగాలు రాయండి.  
Need of Object Oriented Program?
4. Class అనగానేమి ? What is class?
5. Object అనగానేమి ? What is Object?
6. డేటా అబ్స్ట్రాక్షన్ అంటే ఏమిటి ? What is Data Abstraction?
7. JVM, JRE, JIT, OOP లను విస్తరించండి? Expand JVM, JRE, JIT, OOP.
8. Jvm అనగానేమి? What is JVM ?
9. JRE అనగానేమి? What is JRE ?      10. JIT అనగానేమి? What is JIT ?
11. బైట్ కోడ్ అనగానేమి? What is Byte code?
12. మెషిన్ న్యూట్రల్ అంటే ఏమిటి? What is Java machine neutral ?

దీర్ఘ సమాధాన ప్రశ్నలు

1. Oops యొక్క భావనలు వివరించండి? Explain OOPS Concept.
2. జావా ప్రోగ్రాం యొక్క లక్షణాలను వివరించండి? Explain the features of Java? (Or)

Write about the main features of Java.

3. జావా ప్రోగ్రాం యొక్క స్ట్రక్చర్ ప్రాయము Explain the Java program Structure?
4. Java Program ను implement చేయు విధానమును తెలుపండి.

Explain implementation a java program

5. Java కు వాడే వెబ్ బ్రౌజర్ల గురించి వివరించండి?

Explain the web browsers used to Java.

6. Java లో JDK installation procedure ని తెలుపండి.

How to install JDK software and set java home on window system.

## UNIT –II

### జావా ప్రోగ్రామింగ్ లాంగ్వేజ్ (Java Programmig Language)

స్వల్ప సమాధాన ప్రశ్నలు

- 1.వేరియబుల్ ను నిర్వచించండి ? Define Variable.
- 2.స్థిరాంకాలు నిర్వచించండి? Define constants.
3. లిటరల్స్ అనగానేమి? What are Literals?
4. ఐడెంటిఫైయర్స్ అంటే ఏమిటి ? What are Identifiers?
- 5.మిశ్రమ రీతి అంటే ఏమిటి ? What is Mixed Mode?
- 6.కాస్టింగ్ వాల్యూస్ అంటే ఏమిటి? What is Casting values?
- 7.యాక్సెస్ నిర్దేశిస్తుంది ఏమిటి? What is Access Specifier?
- 8.నిర్మాణ దారుడు అంటే ఏమిటి ? What is Constructor?
- 9.నిర్వచన పద్ధతి అంటే ఏమిటి? Define Method.
- 10.తరగతి నిర్వచించండి? Define Class.
- 11.ఆబ్జెక్ట్ ను నిర్వచించండి? Define Object.
- 12.ఏది శూన్యము? Define Void.
13. Instance variable అనగా నేమి? What is instance variable?
14. This ఉపయోగాన్ని తెలుపండి. What is this method?

దీర్ఘ సమాధాన ప్రశ్నలు

- 1.జావాలో యాక్సెస్ క్లాస్ విధానం గురించి వివరించండి?  
Explain Accessing method of a java class.
2. జావాలో యాక్సెస్ స్పెసిఫైయర్ (Access Specifier) విధానం గురించి వివరించండి?  
Explain the Access Specifier in Java?
3. జావాలో ఉండే ఆపరేటర్స్ ను వివరించండి? Explain the Operators in Java?
4. Java లో వివిధ రకాల data types ల గూర్చి తెలుపండి.  
Explain about different data types in java
5. Explain logical operators in java?
6. Explain Relational Operators with examples?
7. Write about separators in Java?

## UNIT-III

కంట్రోల్ స్టేట్ మెంట్ & అర్రెస్

### Control Statement and Array

స్వల్ప సమాధాన ప్రశ్నలు

1.If stament యొక్క syntax ను వివరించండి . Write the Syntax of If statement

2. Break statement అనగానేమి? Write the Syntax of Break statement ?

3. Continue Statement ను గురించి వ్రాయండి?

Write the Syntax of Continue statement ?

4 . Array అనగానేమి ? Define an array ?

5. Array యొక్క రకాలు ? Write different the Types of arrays?

6. Switch statement ను గూర్చి వ్రాయండి.

Explain switch statement with example?

7. String array అనగా నేమి? What is a String Array?

8. Buffer classes అనగా నేమి? What about Buffer classes?

9. String method అనగా నేమి? What is String method?

10. Nested loops అనగా నేమి? Write about Nested Loops ?

దీర్ఘ సమాధాన ప్రశ్నలు

1. Java లో ఉపయోగించే Conditional Control statement ను వివరించుము?

Explain the conditional statement in java?

2. Java లో switch statement యొక్క ఉపయోగము తెలిపి switch statement ను Syntax లో వివరించండి. Explain switch statement with example?

3. Java loop గురించి వివరించండి ? Explain the loops in java?

What are the string method explain?

4. what are the string method explain

5. జావాలో స్ట్రీంగ్ బఫర్ క్లాసు గురించి వివరించండి?

Explain the string Buffer class in Java?

6. String buffer మరియు builder class ల మధ్య తేడాలు ఏమిటి?

What are the difference between string buffer and strong builder classes.

7. జావాలో ప్రోగ్రాం లో 1 నుండి n వరకు ప్రింట్ చెయ్యడాన్ని వివరించండి?

Write a java program to print 1 to n?

8. జావా ప్రోగ్రామ్ లో addition of two matrices వ్రాయండి ?

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 -IV

### Implementing oops in Java, Java method

స్వల్ప సమాధాన ప్రశ్నలు

1. Method అనగానేమి? What is a method?

2. Java పద్ధతిని ఎలా పిలవాలి? How to call a Method?

3. పద్ధతుల (method) ద్వారా లాభాలు ?

What are the advantages of using methods?

4. Constructor అనగానేమి ? What is a constructor?

5. Inheritance అనగానేమి? What is a Inheritance?

6. Polymorphism అనగానేమి? What is a polymorphism?

7. Overloading అనగానేమి ? What is a overloading?

8. Overriding అనగానేమి? What is a overriding?

దీర్ఘ సమాధాన ప్రశ్నలు

1. Java methods ఆర్గ్యుమెంట్స్ మరియు రిటర్న్ value తో వివరించండి?

Explain the Java methods with arguments and return value?

2. తుది వేరియబుల్స్ మరియు methods గురించి వివరించండి?

Explain the final variables and methods?

3. Inheritance విధానాన్ని example లో వివరించండి.

Explain the inheritance with example in Java?

4. వివిధ రకాల inheritance ను తెలుపండి?

How many types of inheritance in Java?

5. Method overloading పద్ధతిని ఉదాహరణలతో వివరించండి

Explain the method overloading with example?

6. Operator overloading అనగా నేమి? ఉదాహరణలతో వివరించండి.

Explain the Operator overloading with example?

7. Polymorphism అనగా నేమి? ఉదాహరణలతో వివరించండి.

Explain the polymorphism with example?

---

## UNIT V

ప్యాకేజీస్ & ఇంటర్ఫేస్

### PACKAGE AND INTEERFACE

స్వల్ప సమాధాన ప్రశ్నలు

1. ప్యాకేజీ (Package) అనగానేమి ? Define package.
2. ప్యాకేజీ (Package) రూపొందించండి ? How to Create a package?
3. ప్యాకేజీ (Package) ని యాక్సెస్ చేసుకోవడం? How to Accessing a package?
4. ఇంటర్ఫేస్ అనగా నేమి? What is an Interface?
5. Abstract class అనగా నేమి? What is Abstract class?
6. Sub-class నిర్వచించండి. Define Sub-class.

దీర్ఘ సమాధాన ప్రశ్నలు

1. ప్యాకేజీ ల యొక్క రకాలను వివరించండి? Explain the types of packages.
2. Package ను తయారు చేయు విధానాన్ని తెలుపండి.

Explain to Create own package with example.

3. ఒక తరగతి అమలు ద్వారా ఎంటెర్ఫేస్ కు అమలు ఎలా అందించాలి ?

How to provide implementation to the interface by the implementation a class

4. Abstract class మరియు Interface మధ్య తేడాలను తెలుపండి?

What are the differences between Abstract class and Interface ?



## UNIT-VI

ఎక్స్ ప్షనల్ హ్యాండింగ్

Exception Handling

స్వల్ప సమాధాన ప్రశ్నలు

1. Debugging అనగానేమి ? What is Debugging?
2. errors యొక్క రకాలు ? Write the Types of errors.
3. Exception అనగానేమి ? Define exception.
4. ఎక్స్ థ్రోస్ అనగానేమి? Write about throws clause.
5. డోషం మరియు మినహాయింపు మధ్య తేడాలు ఏమిటి?

Write difference between error and exception.

6. Unchecked exceptions అనగానేమి? Write about Unchecked exceptions.
7. Try clause ను గూర్చి వ్రాయుము. Write about Try clause.

దీర్ఘ సమాధాన ప్రశ్నలు

1. వివిధ రకాల ఎర్రర్స్ గురించి వ్రాయండి ? Explain the types of errors in java.
2. Exception handling అనగానేమి ? java లో దీనిని ఎలా ఉపయోగిస్తారు?

How to handle the exception in java.

3. Exception యొక్క రకములను తెలుపండి. Explain types of exceptions in Java.
4. జావాలో మినహాయింపులుగా నిర్మించబడినది ఏవి ?

What are the built-in exception in java.

5. User Defined Exceptionలను ఏ విధంగా Throw చేస్తారు (లేదా ) గురించి “Throw” వివరించండి. Or Explain user defined exceptions.

6. Error కి Exception కి మధ్య తేడాలను తెలుపండి.

What are the differences between error and exceptions ?

7. డోషానికి, మినహాయింపుకు మధ్య తేడాలు ఏమిటి?

What are the differences between error and exceptions?

## UNIT -VII

### Threads (థ్రెడ్స్)

స్వల్ప సమాధాన ప్రశ్నలు

1. Threads అనగానేమి? What is a thread?
2. Mutitasking అనగానేమి? What is Multitasking ?
3. Multithreading అనగానేమి? What is Multithreading?
4. Stopping a thread అనగానేమి? Write about Stopping a thread .
5. Block a thread అనగానేమి? Write about Block a thread.

దీర్ఘ సమాధాన ప్రశ్నలు

1. థ్రెడ్ ని ఎలా క్రియేట్ చెయ్యాలి ఉదాహరణతో సహా వివరించండి ?

Explain creating a thread with example.

2. పొడిగించే దూరాన్ని (Extending Thread) ఉదాహరణతో సహా వివరించండి?

Explain extending thread with example .

3. Thread యొక్క జీవిత చక్రాన్ని తెల్పండి? Explain the life cycle of the thread .

4. మల్టీథ్రెడింగ్, మల్టీ టాస్కింగ్ మధ్య తేడాలు ఏమిటి?

What are the differences between multithreading and multitasking ?

5. దారాలను (Threads) సవరించడం? How to manipulating the threads ?

---

## UNIT -VIII

### జావా అప్లెట్

### THE JAVA APPLET

స్వల్ప సమాధాన ప్రశ్నలు

1. Applet అనగానేమి? What is Applet?
2. Appletలను క్రియేట్ చేసే పద్ధతిని వివరించండి? Explain the creating APPLETs?
3. Applet సంకల్ప మరియు తొలగింపు అనగా నేమి?  
Applet initialization and termination ?
4. ఏవైనా 4 HTML applet tag లు రాయండి?

Write any 4 HTML applet tags?

1. Applet యొక్క Life Cycle ను వివరించండి.

Explain applet life cycle?

2. జావా లో HTML applet ట్యాగ్ ల గురించి వివరించండి.

Explain the HTML applet tags in java?

3. Explain writing code and creating simple applet?

4. పరమీటర్లను applet కు పాస్ చేసే ప్రక్రియను రాయండి.

Write the procedure to passing parameters to applet?

5. జావాలో ఉండే లక్షణాల గురించి రాయండి.

Write about the attributes in java?

6 . Write a Java program to create applet with some Background color and Foreground color with message.

7. Applet కు పరామితులు ఎలా పాస్ చేయాలి.

Write about How to passing parameters to applet.

8. జావాలో HTML tags గురించి రాయండి.

Write about the html tags in java.

----"HARD WORK IS SECRETE OF SUCCESS" ----