VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY:: NAMBUR

MID TERM ASSIGNMENT

ACADEMIC YEAR:20 TO 20

Hall Ticket No. :

1	9	B	Q	1	A	0	5	G	3

Name of the Student : NEELA SAI HARSHIT HA

Course : B.TECH /II-year

Branch : ~~ECE~~/CSE/~~EEE~~/~~IT~~

Subject : java programming

ASSIGNMENT / MARKS DETAILS

To be filled by the Student			To be filled by the Subject Teacher		
Submission Date	***Assignment***	***Signature of the Student***	***Max Marks***	***Marks Obtained***	***Signature of Subject Teacher***
20/09/2020	01	N.SaiHarshitha	**5**		

INSTRUCTIONS TO THE STUDENTS

1. The assignment should be submitted to the subject teacher on or before the given schedule.
2. Answer should be written on both sides of the paper.

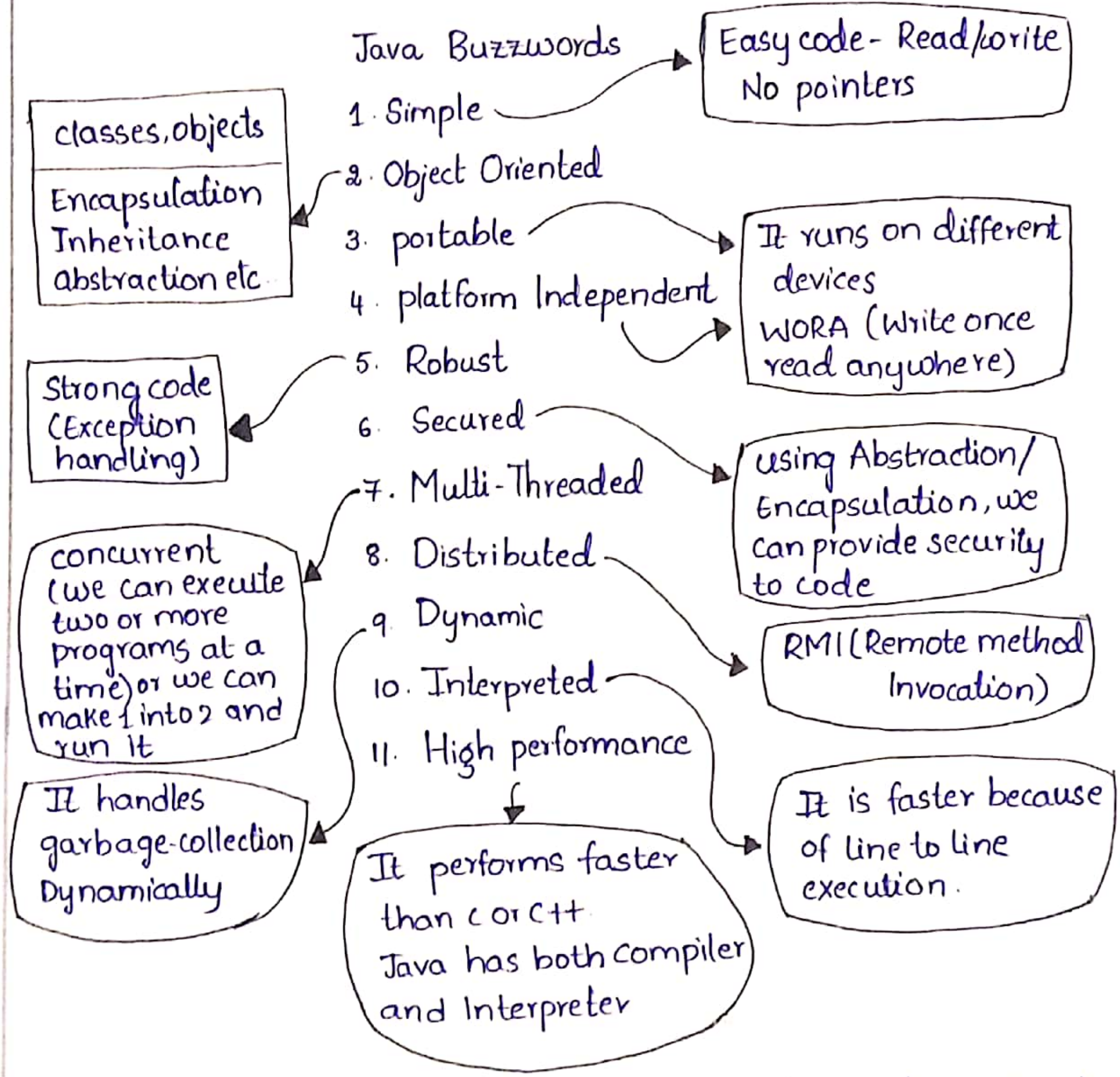
INSTRUCTIONS TO THE SUBJECT TEACHER

1. The Subject teacher has to value with red ball point pen only.
2. The Subject teacher should award the marks on the left hand side of the margin and at the end of the each answer.
3. Do not correct the marks by overwriting or by scratching and writing.
4. The Subject teacher has to post marks in the space provided.

(01) SET-05 19B91A05G3
N·SaiHarshitha

01 List and explain Java Buzzwords. Which factors are making Java famous language

The features of Java are also known as java buzzwords.


Java Buzzwords
1. Simple
Easy code - Read/write
No pointers
2. Object Oriented
classes, objects
Encapsulation
Inheritance
abstraction etc.
3. portable
4. platform Independent
It runs on different devices
WORA (Write once read anywhere)
5. Robust
Strong code (Exception handling)
6. Secured
using Abstraction/Encapsulation, we can provide security to code
7. Multi-Threaded
concurrent (we can execute two or more programs at a time) or we can make 1 into 2 and run it
8. Distributed
RMI (Remote method Invocation)
9. Dynamic
It handles garbage-collection Dynamically
10. Interpreted
It is faster because of line to line execution.
11. High performance
It performs faster than c or c++
Java has both compiler and Interpreter

1 Simple: Java is very easy to learn, and its syntax is simple, clean and easy to understand. Java has removed many complicated and rarely-used features for example, explicit pointers, operator overloading, etc. There is no need to remove unreferenced objects because there is an Automatic Garbage collection in Java.

2 Object-oriented: Java is an object-oriented programming language. Everything in Java is a Object.

Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules. Basic concepts of OOP's are:

19BQ1A05Q3
N·SaiHarshitha

1 Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation.

3 Portable: Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any Implementation.

4. Platform Independent: Java is platform independent because it is different from other languages like C, C++ etc. which are (complicated) compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms, Software based and hardware based. Java provides a software-based Platformthat runs on the top of other hardware-based platforms. It has two components:

1. Run-time Environment
2. API (Application programming Interface)

Java code is complied by a compiler and converted into a byte code. The byte code is a platform Independent code because it can be run on multiple platforms i.e, write Once and run Anywhere (WORA)

5 Robust: Robust simply means strong. Java is robust because:

- It uses strong memory management
- There is alack of pointers that avoids security problems..
- There is automatic garbage collection in java which runs on (JVM) Java Virtual Machine to get rid of objects by a Java application anymore
- There are expection handling and the type checking mechanism in Java. All these points make Java robust.

6 Secured : Java is best known for its Security with Java, we can develop virus-free systems. Java is secured because :

- No explicit pointer
- Java Programs run inside a virtual machine Sandbox
- classloader: Classloader in Java is apart of the Java Runtime Environment (JRE) which is used to load Java classes into the Java virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network Sources.
- Bytecode verifier: It checks the code fragments for illegal code fragments for illegal code that can violate access right to objects
- Security Manager: It determines what resources a class can access such as reading and writing to the local disk.

Java provides these securities by default. Some security can also be provided by an application developer explicitly through SSL, JAAS , Cryplography, etc.

7. Multi-threaded : A thread is like a separate program, executing concurrently. we can write Java Programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi.media, Web application, etc

8 Distributed : Java is distributed because it facilities users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the Internet.

9 Dynamic : Java is a dynamic language. It support dynamic loading of classes. It means classes are loaded on

demand. It also supports functions from its native languages i.e., C and C++. Java supports dynamic compilation and automatic memory management (garbage collection).

10. Interpreted: The bytecode generated is interpreted using a Interpreter. It is faster because of line to line execution.

11. High Performance: Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code. It is still a little bit slower than a compiled language (e.g.., C++). Java is an interpreted language that is why it is slower than compiled languages, e.g.., C, C++ etc..

02 What is the benefits of Inheritance? Explain various forms of Inheritance using suitable code segments.

The process in which one class acquires the properties and functionalities of another class is called Inheritance. The main Benefits of Inheritance are:-

1) It provides reusability of code, so that the subclass can only have unique properties and functionalities, the rest can be inherited from the super class.

2) Reusability enhanced reliability. The base class code will be already tested and debugged.

3) As the existing code is reused, it decreases the maintainence costs.

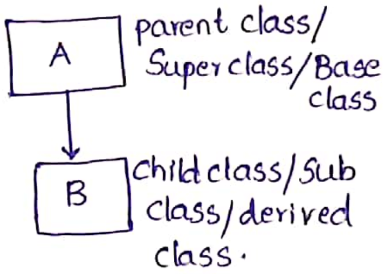
4) Inheritance facilitates creation of class libraries

5) Inheritance helps to reduce code redundancy and supports code extensibility.

Types of Inheritance in Java:-

1) Single level Inheritance
2) Multilevel Inheritance
3) Hierarchial Inheritance
4) Multiple Inheritance

1) Single Inheritance:- In Single level Inheritance A sub Class inherits a Super class. for example, A is Superclass and B is subclass.


A
parent class/ Super class/Base class
B
Child class/Sub class/derived class.

Example for Single Inheritance:-

Here, the superclass is teacher and subclass is Javateacher.

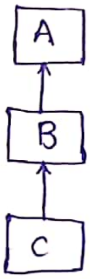
```
class Teacher{
        protected String name;
        protected String gender;

        public Teacher(){
        }
        public Teacher(String name,String gender){
                this.name=name;
                this.gender=gender;
        }
        public void teaches(){
                System.out.println("person" + name + "a" + gender
                        + "teacher, teaches");
        } //Eg:- Person Harshitha,a female teacher teaches.
}
class Java extends Teacher{
        private String collegeName;
        private String deptName;
        private String Subject;
        public Java(String name, String gender,String collegeName,
                    String deptName,String subject){
                super(name, gender);
                this.collegeName=collegeName;
                this.deptName=deptName;
        }       this.subject= subject;
```

```
            public void subjectTeach( ){
                    System.out.println ( name+" working in "
                         collegeName+ " , in department "+
                           deptName+ ", teaching "+ subject);
             }
}
public class SingleInheritanceExample{
        public static void main(String args[ ]){
             Java java1 = new Java( "Harsha", "male", "VVIT",
                                     "CSE", "JAVA");
              java1.teaches ( );
              java1 .subjectteach( );
        }
}
```

Here, we instantiated object for Java subclass .and Inherited all the properties from Teacher superclass .

2) Multi level Inheritance :- When a class extends a class, which extends another class then this is called multilevel Inheritance. For example class C extends class B and class B extends class A. then this type of Inheritance is known as multilevel Inheritance

so in this case class C is implicitly inheriting the properties and methods of class A along with class B that is called Multilevel Inheritance.

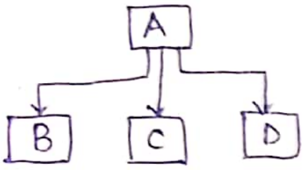


Example of Multilevel Inheritance :

```
Class X
{
    public void methodX( ){
             System.out.println ("class X method");
     }
}
class Y extends X
 {
       public void methodY( ){
              System.out.println("class Ymethod");
       }
}
```

```
class Z extends Y
{
        public void method Z ( ){
                System.out.println("class Z method");
         }
 }
public class MultiInheritance {
        public static void main (String args[ ]) {
                Z obj = new Z( );
                Obj.method X( );
                obj.method Y( );
                obj.method Z( );
         }
 }
```

3) Hierarchical Inheritance : In such kind of Inheritance, one class is inherited by many sub classes. In below example class B,C and D inherits the same class A. A is parent class(or base class) of B,C & D.

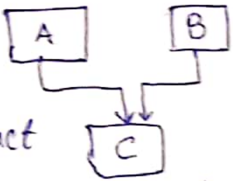


Example of Hierarchial Inheritance:-

```
class A {
        public void methodA( ){
                System.out.println ("method of Class A");
        }
}
class B extends A {
        public void method B( ){
                System.out.println("method of Class B");
        }
 }
 class C extends A {
        public void.methodC( ){
                System.out.println("method of class C");
        }
  }
```

```
class D extends A
{
        public void methodD( ){
                System.out.println("method of class D");
        }
}
class Hierarchical Inheritance{
        public static void main(String args[]){
                B obj1 = new B( );
                C obj2 = new C( );
                D obj3 = new D( );
                Obj1.methodA( );
                obj2. methodA( );
                obj3. method A ( );
        }
}
```

4) Multiple Inheritance :- It refers to the concept of one class extending more than one base class. The Inheritance with one childclass and one or more Superclasses is called Multiple Inheritance.



This type of multiple Inheritance is possible only when any of the parent class is Abstract class i.e, no definition methods (without any concreate methods).
This type of classes is called Interface.

(or)

It is a special kind of class which only contains Abstract methods. We can only define data which is constant.

Example :-

```
interface A{
        public static final int X = 20;
        public void print( );
}
class B {
        protected Int y = 30;
        public void print( ){
                System.out.println(y);
        }
}
```

9

19BQ1A05G3
N·SaiHarshitha

```
class C extends B implements A {
        private int z=40;
        public void print(){
                System.out.println("In class c, z= "+ z);
                System.out.println( y);
                System.out.println(X);
        }
}
public class InterfaceDemo{
        public static void main(String []args){
                C objC =new C( );
                Objc.print( );
        }
}
```

Scanned with CamScanner

19BQ1A05G3
N.SaiHarshitha

3. Define a class named movieMagic with the following description:

Instance variables/data members :

int year - to store the year of release of a movie

String title to store the title of the movie .

float rating - to store the popularity rating of the movie .

(minimum rating = 0.0 and maximum rating = 5.0)

Member Methods :

i, movieMagic( ) Default Constructor to initialize numeric data members to 0 and String data member to " ".

ii, void accept( ) To input and store year, title and rating .

iii, void display( ) To display the title of a movie and a message based on the rating as per the table below .

Rating	Message to be displayed .
0.0 to 2.0	flop
2.1 to 3.4	Semi-hit
3.5 to 4.5	Hit
4.6 to 5.0	Super Hit

Write a main method to create an object an object of the class and call the above member methods .

ans
```
import   java.io.* ;
import   java.util.Scanner;
public class MovieMagic{
          // variables
          private int year;
          private String title;
          private float rating;
          //methods
          public MovieMagic( ){
                    year = 0;
                    title = "  " ;
                    rating = 0.0f;
          }
          public void accept( ){
                    Scanner scan = new Scanner(System.in);
                    System.out.println("enter details");
                    year = Scan.nextInt( );
                    title = scan.next( );
                    rating = scan.nextFloat( );
          }
public void display(){
        System.out.print(title+" is "};
        if(rating<=2.0){
                System.out.println("Flop");
        }
        else if (rating>=2.1 && rating <=3.4){
                System.out.println("Semi-hit");
        }
        else if (rating>=3.5 && rating <=4.5){
                System.out.println ( "Hit");
        }
        else{
                if(rating >=4.6 && rating <=5.0){
                        System.out.println("Super Hit");
                }
        }
}
public static void main(String args[]){
            MovieMagic obj=new MovieMagic();
            obj.accept();
            obj.display();
    }
}
```

github repository : github.com/SaiHarshitha11028/Java-Programming.

19BQ1A05G3
N.SaiHarshitha.

04. Write a class to overload a function num_calc() as follows:

i, void num_calc(int num, char ch) with one Integer argument and one character argument, computes the square of integer argument If choice ch is 's' otherwise finds its cube.

ii, void num-calc (int a, int b, char ch) with two Integer arguments and one character argument. It computes the product of Integer arguments if ch is 'p' else adds the integers.

iii, void num-calc (String S1, String S2) with two string arguments, which prints whether the strings are equal or not.

ans

```
class Overload {
        public Overload ( ){
        }
        public void num_calc (int num, char ch) {
                if (ch = = 's'){
                        System.out.println(num * num);
                }
                else {
                        System.out.println(num * num * num);
                }
        }
        public void num-calc (int a, int b, char ch) {
                if (ch = = 'p'){
                        System.out.println(a * b);
                }
                else{
                        System.out.println(a+b);
                }
        }
        public void num-calc (String s1, String s2){
                if (s1.equalsIgnoreCase (s2) {
                        System.out.println("Strings are equal");
                }
                else{
                        System.out.println("Strings are notequal");
                }
        }
}
```

// full program in gitHub repository :- github.com/SaiHarshitha11028/Java-Programming

Reference :-

01 :- Java class Notes, JavatPoint (Google chrome) .

02 :- Java class Notes (Abstract class , Singlelevel Inheritance)
↓
(Multiple Inheritance)

Beginners Book (Google Chrome) for (Multilevel Inheritance,

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision of the Department

To facilitate quality education by focusing on assimilation, generation and dissemination of knowledge in the area of Computer Science & Engineering to transform students into socially responsible engineers.

Mission of the Department

- Equip our graduates with the knowledge by ***student centric teaching-learning process*** and expertise to contribute significantly to the software industry and to continue to grow professionally.
- To train ***socially responsible, disciplined engineers*** who work with good leadership skills and can contribute for nation building.
- To make our graduates ***aware of cutting edge technologies*** and make them industry-ready engineers.
- To shape the department into a ***centre of academic and research excellence***.

Program Educational Objectives

PEO-1	To provide the graduates with solid foundation in Computer Science and Engineering along with the **fundamentals of Mathematics and Sciences** with a view to impart in them high quality technical skills like modelling, analyzing, designing, programming and implementation with global competence.
PEO-2	To prepare and motivate graduates with **recent technological developments related to core subjects** like programming, databases, design of compilers and Network Security aspects and future technologies so as to contribute effectively for Research & Development by participating in professional activities like publishing and seeking copy rights.
PEO-3	To train graduates to choose an appropriate **career in employment, higher education or entrepreneurship** by empowering them to excel in competitive examinations, by preparing them for lifelong learning and by inculcating in them ethical leadership skills.
PEO-4	To train the graduates to have basic **interpersonal skills and sense of social responsibility** that paves them a way to become good team members and leaders.