



VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY:: NAMBUR

**MID TERM ASSIGNMENT
ACADEMIC YEAR:2020 TO 2021**

Hall Ticket No. :

1	9	B	Q	1	A	0	5	L	8
---	---	---	---	---	---	---	---	---	---

Name of the Student : SURA RANAPRATHAP REDDY

Course : BTECH

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

Subject : JAVA PROGRAMMING

Set 5

1. List and explain Java buzzwords. Which factors are making Java famous language.
2. What are the benefits of inheritance? Explain various forms of inheritance with suitable code segments.
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

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

- 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.

1A) The JAVA programming language is a high-level language that can be characterized by all of the following buzz words.

- * Simple
- * Object oriented
- * Distributed
- * Interpreted
- * Robust
- * Secure
- * Architecture neutral
- * Portable
- * High performance
- * Multithreaded
- * Dynamic

⇒ Simple:

- JAVA was designed to be easy for professional programmer to learn & use effectively.
- It's simple and easy to learn if you already known the basics concepts of object oriented programming.

⇒ Object oriented:

- JAVA is true object oriented language.
- Almost "Everything is an object" paradigm. All program code and data reside within objects and classes.
- JAVA comes with an extensive set of classes,

Arranged in packages that can be used in our programs through inheritance.

⇒ Distributed :-

- JAVA is designed for distributed environment of the internet. It's used for creating applications on networks.
- JAVA applications can access remote objects on Internet as easily as they can do in local system.

⇒ Compiled & Interpreted :-

- usually a computer language is either compiled or interpreted. JAVA combines both this approach and makes it a two-stage system
- Compiled : JAVA enables creation of a cross platform programs by compiling into an intermediate representation called JAVA Bytecode.
- Interpreted : Bytecode is then interpreted, which generates machine code that can be directly executed by the machine that provides a Java virtual machine.

⇒ Robust :-

- It provides many features that make the program execute reliably in variety of environments.
- JAVA is a strictly typed language. It checks code both at compile time and runtime.
- JAVA takes care of all memory management problems with garbage collection
- Java, with the help of exception handling, captures all types of serious errors, & eliminates any risk of crashing the system.

⇒ Secure:-

- Java provides a "firewall" between a networked application and your computer
- When a Java compatible web browser is used, downloading can be done safely without fear of viral infection or malicious intent.

⇒ Architecture Neutral:

- Java language and Java virtual machine helped in achieving the goal of "write once; run anywhere, any time, forever".

- Changes and upgrades in operating systems, processors and system resources will not force any changes in Java codes.

⇒ Portable:-

- Java provides a way to download programs dynamically to all the various types of platforms connected to the Internet.
- It helps in generating portable executable code

⇒ High Performance:

- Java performance is high because of use of bytecode.
- The bytecode was used, so that it was easily translated into native machine code.

⇒ Multithreaded:

- Multithreaded programs handled multiple tasks simultaneously which was helpful in creating interactive, networked programs.

- Java run-time system comes with tools that support multiprocess synchronization used to construct smoothly interactive systems.

⇒ Dynamic:-

- Java is capable of linking in new class libraries, methods, and objects.
- It can also link native methods.

factors making JAVA famous language

- 1) JAVA is Easy to learn.
 - 2) Java is an object-oriented programming language.
 - 3) Java has Rich API.
 - 4) powerful development tools e.g., Eclipse, Netbeans.
 - 5) great collection of open source libraries.
 - 6) wonderful Community Support.
 - 7) JAVA is free.
 - 8) Excellent documentation support - Javadocs.
 - 9) JAVA is platform independent.
- Q) The process by which one class acquires the properties and functionalities of another class is called inheritance.
- Benefits of inheritance
- 1) for method overriding.
 - 2) To avoid duplication of the code in an application by sharing common code amongst several subclasses.
 - 3) It makes application code more flexible.
 - 4) It increases code Reusability.

Child class: The class that extends the features of another class is known as subclass / derived class / child class.

Parent class: The class whose properties and functionalities are inherited by another class is called superclass or Base class or parent class.

Types of inheritance:

① Single inheritance:

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

Example:

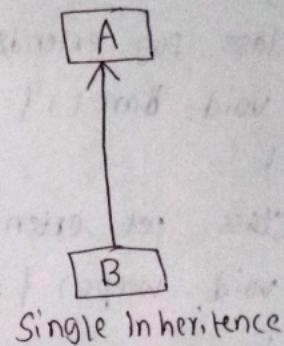
```
Class Animal {  
    void eat () { System.out.println ("eating..."); }  
}  
class Dog extends Animal  
{  
    void bark () { System.out.println ("barking..."); }  
}
```

```
Class TestInheritance {  
    public static void main (String args []) {
```

```
        Dog d = new Dog ();
```

```
        d.bark ();
```

```
        d.eat ();
```

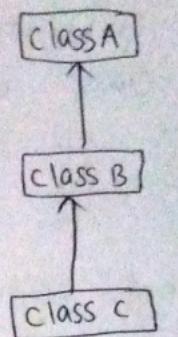


② Multi level inheritance:

It refers to a class and parent class relationship where a class extends the child class. for example CLASS C EXTENDS CLASS B and CLASS B EXTENDS CLASS A.

Example:

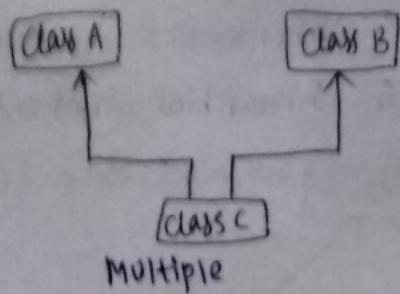
```
Class Animal {  
    void eat() { System.out.println("eating"); }  
}  
  
Class Dog extends Animal {  
    void bark() { System.out.println("barking"); }  
}  
  
Class Pet extends Dog {  
    void weep() { System.out.println("weeping"); }  
}  
  
Class TestInheritance2 {  
    public static void main(String args[]) {  
        Pet d = new Pet();  
        d.weep()  
        d.bark()  
        d.eat()  
    }  
}
```



Multilevel

③ Multiple Inheritance:

refers to the concept of one class extending more than one class, which means a child has two parent classes. for example CLASS C EXTENDS both CLASSES A & B. Java doesn't support multiple inheritance.



④ Hierarchical inheritance: It refers to a child and parent class relationship where more than one classes extends the same class?

Example

```

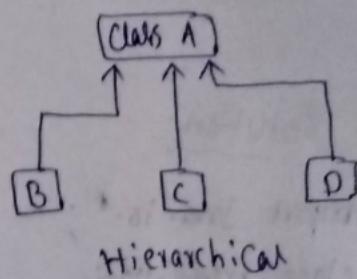
class Animal {
    void eat () { System.out.println ("eating"); }
}

class Dog extends Animal {
    void bark() { System.out.println ("Barking"); }
}

class Cat extends Animal {
    void meow() { System.out.println ("meowing"); }
}

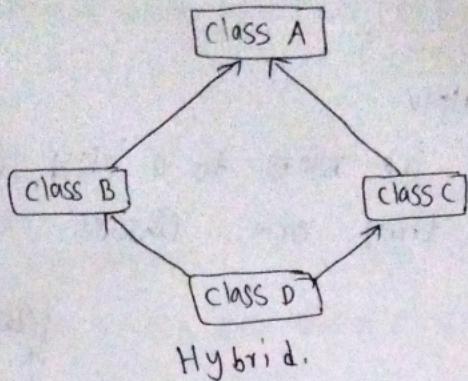
class TestInheritance3 {
    public static void main (String args[]) {
        Cat c = new Cat();
        c.meow();
        c.bark();
    }
}

```



⑤ Hybrid inheritance: Combination of more than one types of inheritance in a single program. for example class A & B extends class C and another class D extends class A

then this is a hybrid inheritance example because it is
a combination of single & hierarchical inheritance.



3A) Solution:

```
import java.io.*;
class movieMagic
{
    int year;
    String title;
    float rating;
    public movieMagic ()
    {
        year = 0;
        title = " ";
        rating = 0.0;
    }
    void accept() throws IOException
    {
        InputStreamReader IR = new InputStreamReader (System.in);
        BufferedReader br = new BufferedReader (IR);
        System.out.println ("Enter the year of release:");
        year = Integer.parseInt (br.readLine ());
        System.out.println ("Enter the title of movie:");
    }
}
```

```

        title = br.readLine();
        System.out.println("Enter the popularity rating:");
        rating = float.parseFloat(br.readLine());
    }

    void display()
    {
        System.out.println("The title of movie is: " + title);
        if (rating >= 0.0 && 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() throws IOException
    {
        movieMagic obj = new movieMagic();
        obj.accept();
        obj.display();
    }

```

Note

Variable Description:

S.No	variable name	Data type	Purpose.
1.	year	int	To store the year of release of movie
2.	title	string	To store the title of movie
3.	rating	float	To store rating of the movie.
4.	IR	—	for input stream reader
5.	br	—	for buffered reader
6.	obj	—	for object of the class.

5) Solution

```
import java.io.*;
import java.util.Scanner;
class Overloaded
{
    void num_calc (int num, char ch)
    {
        int s=0
        if (ch=='S')
            s = num*num;
        else
            s = num*num*num;
        System.out.println ("s = " +s);
    }
    void num_calc (int a, int b, char ch)
    {
        int s=0
        if (ch=='P')
            s = a*b;
        else
            s = a+b;
        System.out.println ("s = " +s);
    }
    void num_calc (String s1, String s2)
    {
        if (s1.equals(s2))
            System.out.println ("Both strings are equal!");
        else
            System.out.println ("Both strings are not equal.");
    }
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in),
```

```
overloaded obj = new overloaded();
obj.numCalc(5, 's');
obj.numCalc(8, 3, 'n');
obj.numCalc("Raman", "savan");
}
```