Set-V 19BQ1A05K2

① List and explain Java buzzwords. Which factors are making Java famous language.

A) Java Buzzwords:-

1) Simple:- Easy to code, write, read, modularity. No pointers

2) Object oriented:- Class, objects, Encapsulation, interetance, polymerphism.

3) Portable:- WORA (Write once run anywhere).

4) platform independent:- Java programs are executed anywhere when we have code and its platform independent means it can be executed in any Operating System. It contains intermediate code which generates Byte code (JVM). Java Virtual Machine (or) JRE (Java Runtime Environment)

5) Robust:- Strong (with out errors), Contains exception hobling, it gives strength to your code, Contains garbage collection

6) Security:- It poorides security for data and members using Abstraction and Encapsulation.

7) Multi-Threaded:- Concurrent execution, Threads

8) High-Performance:- It is faster than C, C++. Java contains both compiler and interpretor.

9) Distributed:- TCP, P, RMI (Remote Method Innotation). Using internet we can execute the code present in other systems using our systems

10) Interpreted:- In Java the code is interpreted faster.

11) Dynamic :- We can execute programs dynamically Java is capable of linking in new classes, libraries, method and objects. It can also link ~~nature~~ native methods.

Factors making Java as famous language :-

1) Java is easy to learn

2) Java is an object oriented programming language.

3) Java has Rich API

4) Java has powerful development tools like Eclipse and Netbeans

5) It has great collection of Open Source libraries.

6) Java has wonderful Community Support.

7) Java is free Since java is free from the start i.e. you do not need to pay anything to create java application. This free thing also helped Java to become popular among larger organization

8) Java has excellent documentation Support – Java docs

9) Java is platform independent. The idea of platform independance is great and Java's tagline write once run anywhere was enticing enough to attract lots of new development in java

10) Java is Everywhere, It is on the desktop. It's on mobile, it's on the card and almost every where and so is Java Programmers. This vast ability of Java programmers is another reasons why organization prefer to so choose Java for their new development projects

- What are the benefits of inheritance? Explain various forms of Inheritance with suitable code segments.

Various forms of Inheritance:-

Below are the various types of inheritance in java

1) Single Inheritance:-

When a class extends another one class by only then we call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Here a parent class of B. And B would be a child class of A.

Single Inheritance

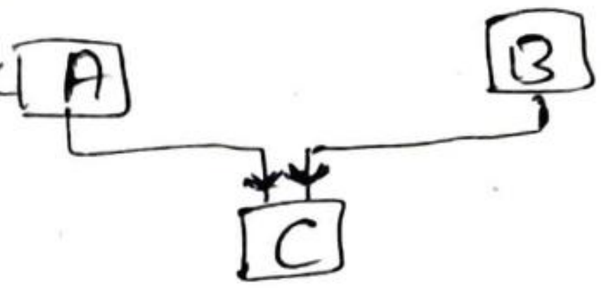
Eg:-

```
class A
{
   Public void MethodAC)
   {
     System.out.println("Base class Method");
   }
}
class B
{
  Public void method BC)
  {
   System.out.println("Child class method");
  }
  Public static void main(string args[])
  {
   B obj = new BC);
   Obj.Method A( ); // calling super class method
   obj.Method B( ); // calling local method
  }
}
```

2) Multiple Inheritance:-

Multiple Inheritance refers to the concept of one class extending (or inherits) more than one base class or parent. The problem with "Multiple inheritance" is that the derived class will have to manage the dependancy on two base class

-> Java doesn't allow Multiple inheritance



3) Multilevel Inheritance:-

Multi level Inheritance refers to the mechanism in OOP technology where one can inherit from a derived class, there by making this derived class the base class for the new class. As in below flow diagram C is subclass or Child class B and B is a child class of A.

Multilevel Inheritance Example:-

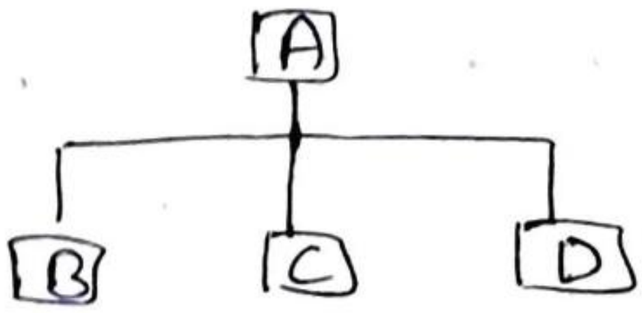
```
class x
{
  Public void method x()
  {
  System.out.println("class Xmethod");
  }
}
class y extends x
{
Public void method y()
{
System.out.println("class Ymethod");
}
}
```

```
class z extends y
{
  Public void method z()
  {
   System.out.println("class z method");
  }
  Public static void main (String args[])
  {
    z obj = new z();
    obj.method x(); // calling grant Parent class method
    obj.method y(); // calling parent class method
    obj.Method z(); // calling local method
  }
}
```

4. Hierarchial Inheritance:-

In Such kind of Inheritance one class is inherited by many su class. In below example class B, C and B inherits the same c A. A is parent class (or base class) of B, C & D

Example for Hierarchial Inheritance:-




```
class A
{
  Public void method A()
  {
   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 method C()
  {
  System.out.println("method of class c");
  }
}
Class  D extends A
{
 Public void method D()
  {
  System.out.println("method of class D");
  }
}
Class Java Example
{
 Public static void main(String args [])
  {
    B obj1 = new B();
    C obj2 = new C();
    D obj3 = new D();
    Obj1.Method A();
    Obj2.Method A();
    Obj3.Method A();
  }
}
```

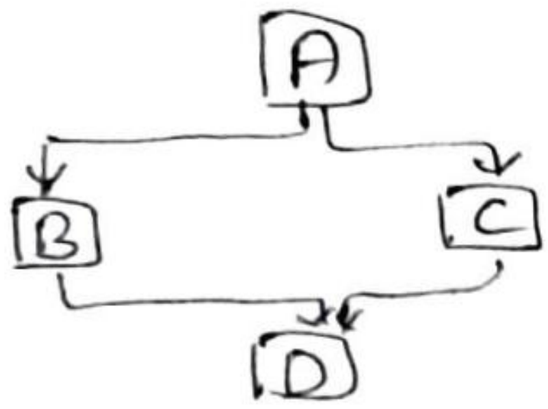
Output:-

Method of class A
Method of class A
Method of class A.

5. Hybrid Inheritance :-

In simple terms you can say that Hybrid inheritance is a combo of Single & Mutiple inheritance (or) more than one types of inheritance.

▲ The following example inherits together to form hybrid inheritance



In the example program

class A and B extends class C
→ Hierachical inheritance

class D extends class A
→ single inheritance

```
class C
{
  Public void disp()
  {
    System.out.println("c");
  }
}
class A extends C
{
  Public void disp()
  {
    System.out.println("A");
  }
}
class B extends C
{
  Public void dispc()
  {
    System.out.println("B");
  }
}
class D extends A
{
  Public void disp()
  {
    system.out.println("D");
  }
  Public static void main(String args[]){
      D obj = new DC();
    } Obj.disp();
}
```

output :- D

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 of the class and call the above member methods.

Ans

```
import java.util.*;

class movieMagic {
    int year;
    float rating;
    String title;
    movieMagic() {
        year=0
```

Ans:

```
import java.util.*;
class movieMagic {
    int year;
    float rating;
    String title;
    movieMagic() {
        Year = 0;
        rating = 0.0f;
        title = "";
}
Void accept() {
Scanner SC = new Scanner(System.in);
    Year = sc.nextInt();
    title = sc.next();
    rating = sc.nextFloat();
 }
Void display() {
    System.out.println("Title:" + title);
    If (rating > 0 && rating <= 2)
       System.out.println("Flop");
    elsif( rating > 2 && rating <= 3.4)
       System.out.println("semi-Hit");
    else if (rating > 3.4 && rating <= 4.5)
       System.out.println("Hit");
    else if (rating > 4.5 && rating <= 5.0)
       System.out.println("Super Hit");
    else
       System.out.println("INVALID RATING");
}
```

```
Public static void main (String[] args){
    movieMagic obj = new movieMagic();
    obj.accept();
    obj.display();
}
}
```

4. 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 argument and one character argument. It computes the product of Integer arguments if ch is 'p' else adds the Integer

iii) Void num_calc (String s1, String s2) with two string arguments, which prints whether the strings are equal (or) not.

Ans:-

```
import java.util.Scanner;
Public class overloading {
    Void num_calc (int num, char ch) {
        int op=0;
        if (ch=='s')
            op = num* num;
        else
            op = num* num* num;
        System.out.println(op);
}
```

```
void num_calc ( int a, int b, char ch){
    int op;
    if (ch == 'p')
        OP = a*b;
    else
        OP = a+b;
    System.out.println(op);
}
void num_calc(String s1, String s2){
    if (s1.equals(s2))
        System.out.println("Both Strings are same");
    else
        System.out.println("Both strings are not same");
}
Public static void main(String[] args){
    overloading obj = new Overloading();
    Scanner sc = new Scanner (System.in);
    int num = sc.nextInt();
    char ch1 = sc.next().charAt(0);
    Int a = sc.nextInt();
    int b= sc.nextInt();
    char ch2 = sc.next().charAt(0);
    String s1 = sc.next();
    String s2 = sc.next();
    obj.num_calc(num, ch1);
    obj.num_calc(a,b,ch2);
} obj.num_calc(s1, s2);
}
```