

JAVA PROGRAMMING ASSIGNMENT 1

19BQ1A05F0

(1) List and explain java buzzwords. What factors are making java famous language.

(Ans) Java Buzzwords:

Simple: Java language manages to handle all of its concepts in flexible manner. It is simple to use and easy to learn. It extracts all most all the features of C/C++. Java was designed to be easy for the professional programmer to learn and use effectively. And java has no pointers that programmers need to handle directly.

Object oriented: Java programmer is forced to use classes and objects from the very beginning, since a function in java cannot exist outside a class. Java comes with an extensive set of built-in classes, arranged in "packages". The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust: To provide better reliability, Java has to implement applications on variety of platforms. Hence it requires being robust language. To do so, It has to concentrate on few areas like identifying the errors i.e, error handling & memory management. In fact, Java doesn't allow you to make any mistakes. As Java is strictly typed language, it checks your code at compile time. However it also checks your code at run time.

②

Secure: All the Java that provides the user is nothing but secured programming techniques. Java implements a separate Security Manager so that the user can be benefited in implementing the objects with ease of use. Java is intended to be used in networked/distributed environments. Towards the end, a lot of emphasis has been placed on security. Java enables the construction of virus-free, tamper-free systems.

Architecture neutral: A central issue for the Java designers was that of code longevity and portability. The Java language and JVM helped in achieving the goal of "write once; run anywhere, any time, forever". Changes and upgrades in OS, processors and system resources will not force any changes in Java programming.

Portable: Java provides a way to download programs dynamically to all various types of platforms connected to internet. It helps in generating portable executable code.

Strongly typed: Often, saying Java is a strongly typed language is absolute because it is very much particular about the type of data. The user needs to be careful while dealing with data types.

Dynamic: Dynamic nature of java gives more comfortness to the designer because dynamic declaration & redeclaration of data members becomes easy at runtime. This makes it possible to dynamically link the code in a safe manner.

Distributed: Java is designed for distributed environments like internet, because it handles TCP/IP protocols & other protocols. With this nature Java objects are distributed over the network and get executed remotely on demand.

Multithreaded: Java has another advantage of allowing the use to develop interactive, networked programs. To achieve this, Java supports multithreading this allows you to run many tasks simultaneously. Java provides built-in support for multithreading so that the user can design such application in a most sophisticated way.

Interpreted & High Performance: Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be interpreted on any system that provides a Java virtual machine. The Java bytecode was carefully designed so that it would be easy to

translate directly into native machine code for very high performance by using a just-in-time compiler. Java run-time systems that provide this feature lose none of the benefits of the platform-independent code.

→ The above features of Java are making Java famous language.

(2) What are the benefits of inheritance? Explain various forms of inheritance with suitable code segments.

Ans: Extracting the features of an existing class to a new class is known as Inheritance. The private data members and private member methods are strictly not inherited.

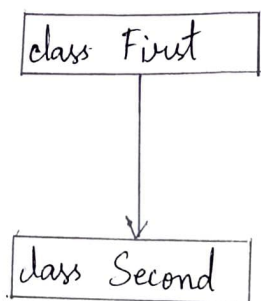
The advantages of Inheritance:

- Code reusability
- We can extend the code without disturbing the existing code.

The inheritances in Java are:

→ Single Inheritance: Extracting the features (data members and member methods) of only one base class to only one derived class is known as single Inheritance.

⑤

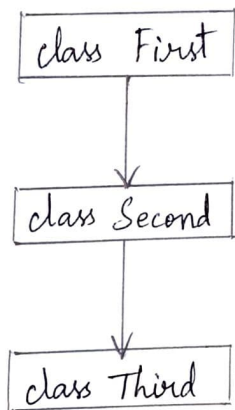


Example: class First
 {

 }
 class Second extends First
 {

 }

→ Multi Level Inheritance: Extracting the features from only one Base class to only one derived class for more than one level is known as Multi Level Inheritance.



Example: class First
 {

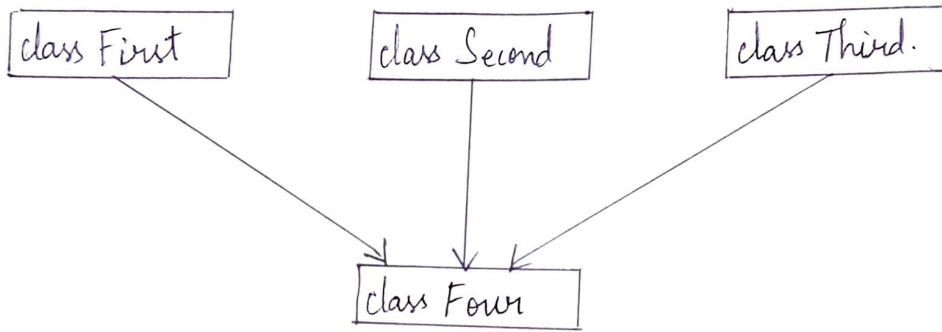
 }
 class Second extends First
 {

 }
 class Third extends Second
 {

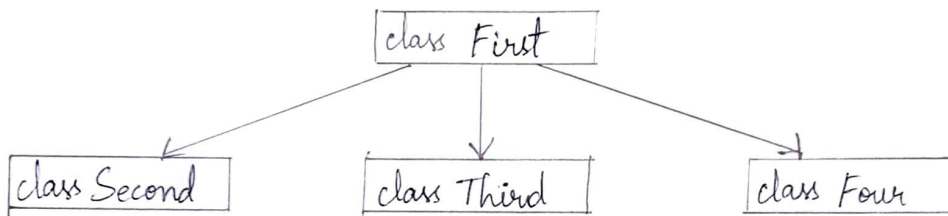
 }

→ Multiple Inheritance: Extracting the features from more than one base class to only one Derived class is known as Multiple Inheritance. Java does not support multiple Inheritance directly.

⑥



→ Hierarchical Inheritance: Extracting features of only one base class to more than one Derived class.

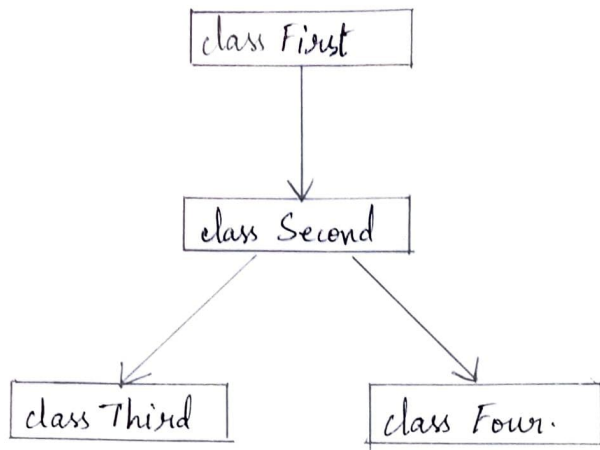


Example:

```

class First
{
    ---
    ---
    ---
}
class Second extends First
{
    ---
    ---
    ---
}
class Third extends First
{
    ---
    ---
    ---
}
class Four extends First
{
    ---
    ---
    ---
}
  
```

⑦ → Hybrid Inheritance: Combination of any two Inheritances is considered as Hybrid Inheritance.



Example:

```

class First
{
  ---
  ---
  ---
}
class Second extends First
{
  ---
  ---
  ---
}
class Third extends Second
{
  ---
  ---
  ---
}
class Four extends Second
{
  ---
  ---
  ---
}
  
```


③

(3) Define a class named movieMagic with the following description:

Instance/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 accept~~

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.

19BQ1AD5F0

⑨

```
import java.io.*;
import java.util.*;

class movieMagic
{
    int year;
    String title;
    float rating;

    movieMagic()
    {
        year = 0;
        rating = 0.0F;
        title = "";
    }

    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the title:");
        title = sc.nextLine();
        System.out.println("Enter the year of its release:");
        year = sc.nextInt();
        System.out.println("Enter rating:");
        rating = sc.nextFloat();
    }

    void display()
    {
        System.out.println("The title of the movie is: " + title);
    }
}
```

(11)

(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 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.

class overLoad

```
{
    void num-calc(int num, char ch)
    {
        if (ch == 's')
            System.out.println(num * num);
        else
            System.out.println(num * num * num);
    }
    void num-calc(int a, int b, char ch)
    {
        if (ch == 'p')
            System.out.println(a * b);
        else
            System.out.println(a + b);
    }
}
```

(12)

```
void num_calc (String s1, String s2)
{
```

```
    if (s1.equals(s2))
```

```
        System.out.println("Strings are equal");
```

```
    else
```

```
        System.out.println("Strings are not equal");
```

```
}
```

```
}
```

Resources: understanding computer Applications with BLUE J
www.aplustopper.com