19/09/20

SIRI LALITHA. A
19BQ1A05L4
II-CSE-D

JAVA ASSIGNMENT-I

SET-I

1. Write about the role of JVM, Java API in developing independent java programming with suitable example.

A:- JVM - (JAVA VIRTUAL MACHINE):

JVM depends on the operating system. JVM is platform-dependent and that is the reason why Java is able become "Platform Independent". The magic of byte code that makes it platform independent. This also adds to an important feature in the JAVA Language termed as 'portability'. Every System has its own JVM which gets installed automatically when the jdk software is installed. For every operating system seperate JVM is available which is capable to read the .class file or byte code. An important point to be noted is that while JAVA is platform-independent language, the JVM is plat-form -dependent. Different JVM is designed for different OS and byte code is able to run on different OS.

JVM is an abstract machine. It is specified that provides run time environment in which java byte code can be executed. JVM are available for many hardware and software platforms.

JVM take cares of creating of class file related to your program and you can usesame class file in different operating system.

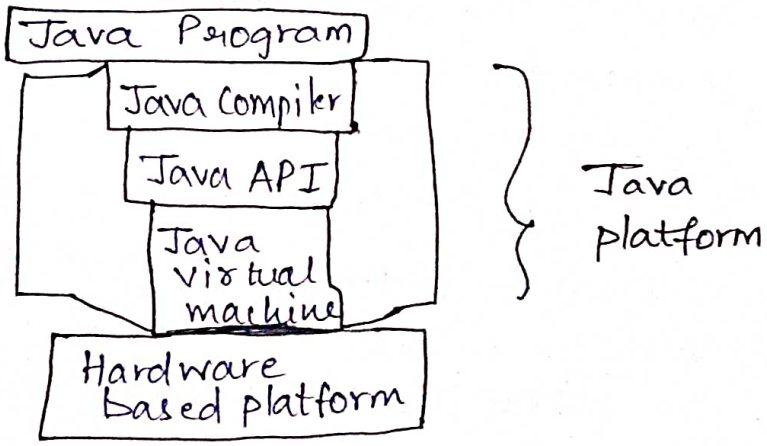
JVM acts as interface between your OS and your program code.

→ It also responsible for calling main method and running your program.

→ It also provides security related features.

→ It is also responsible for calling garbage collector for clean up operations.

Java Application programming Interface (API) is the area of Java development kit JDK. An API includes classes, Interfaces, packages and also their methods fields and constructors.



The JAX-RS (Java API for Restful Web services) client provides following capabilities:-

1. Invoke REST endpoints.
2. Use similar paths / templates as server API
3. Improved async invoker support.
4. Server-Sent Events support.

Examples:-

→ Using JAVA API top sites like Twitter, Yahoo uses this kind of pattern.

→ Most social media sites like facebook.com benefits REST Web services.

→ Mobile App development, is developing quickly as well as their server connection, by using this REST pattern since it is quicker in processing request and response data.

2. Write an example program explain the concept of nested classes in Java.

In Java, it is possible to define a class with in another class, such classes are known as nested classes. They enable you to logically group classes that are only used in once place, thus this increase the use of encapsulation, and creates more readable and maintainable code.

* TO scope of a nested class is bounded by the scope of its enclosing class. Thus in above example, class Nested Class doesnot exist independently of class outer class.

* It has access to members, classes within another class, these classes are called nested classes. They enable to logically group classes i.e., are only used in 1 place, thus increase use of encapsulation,

Creates more readable and maintainable code.

* The scope of a nested class is bounded by scope of its enclosing class.

* A nested class has access to members, including private members, of the class i.e., mentioned. As a member of its enclosing class, nested class can be declared as private, public, protected.

* Nested classes are divided into 2 categories:

1) Static nested class :-

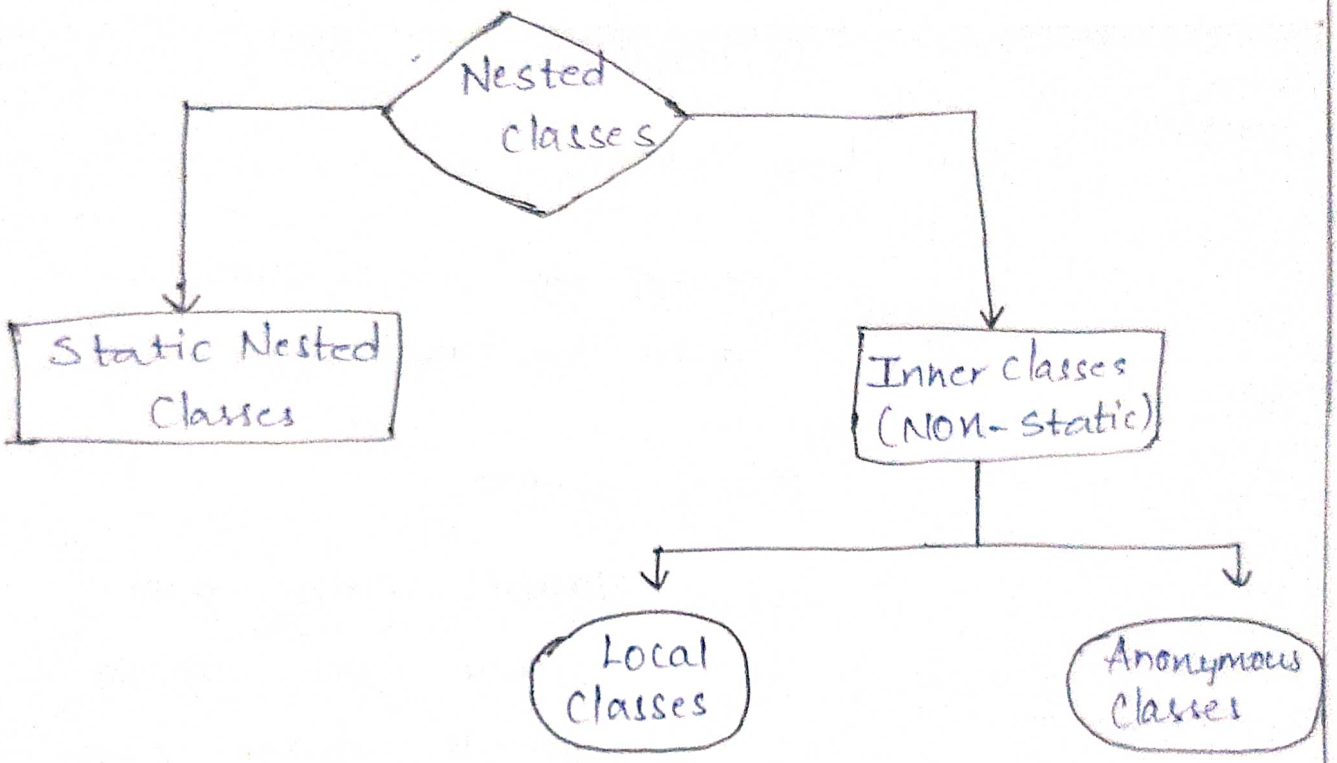
Nested classes that are declared static are called static Nested classes.

2) Inner class :-

An inner class is a non-static nested class.

Syntax :-

```
Class Outer Class{
    ....
    class Nested class{
        ....
    }
}
```


Nested classes
Static Nested Classes
Inner classes (NON-static)
Local Classes
Anonymous Classes

CODE :-

```
public class college {

    static int numofstudents;
    static int numoffaculity;

    class Students {

        private String name;
        private String rollNO;
        private string course;

        public student() { numofstudents++; }
        public Student (String name, string rollNo,
                        String course) {
            this.name = Name;
            this.rollNO = rollNO;
            this.course = course;
            numofcourse++;
        }
        public String getName() {
            return name;
        }
        public void setName (String name) {
            this.name = name;
        }
        public string getRollNO() {
            return rollNo;
        }
        public string getCourse() {
            return course;
        }
```

```
public void setRollNO(String rollNO) {
        this.rollNO = rollNO;
}
public void setCourse(string course){
        this.course = course;
}
public string getInformation(){
    return rollNO + " who's name is " + name + "studies"
            + course;
    }
}
class Faculity{
        private String name;
        private string subject;
        private int salary;
        public Faculity() { numOffacuilty++; }
        public Faculity(String name, string Subject, int
                                                salary){
            this.name = name;
            this.subject = subject;
            this.salary = salary;
            numOfFaculity++;
        }
        public String getInformation(){
            return name + " teaches " + subjects + " and earns "
```

```
    + String.valueOf(salary);
  }
}

// Driver code
public static void main(String[] args) {
        College VVIT = new College();
        College.Student Student1 = VVIT.new Student
                ("Siri Lalitha", "19BQ1A05L4",
                "CSE");
        College.Student student2 = VVIT.newStudent("karthik"
                "19BQ1A05L5","CSE");

        College.Faculity faculity1 = VVIT.new Faculity
                ("Hari Prasad","C",50000);
        College.Faculity faculity2 = VVIT.new Faculity
                ("Naga Sriharsha","JAVA",50000);

        System.out.println(Student1.getInformation());
        System.out.println(faculity2.getInformation());
        System.out.println("No.of students in college:"+
                VVIT.numofstudents);
        System.out.println("No.of teacher in college;"+
                VVIT.nomofFaculity);
    }
}
```

3. Design a class Railway ticket with the following description . . . .

```
import java.util.*;
public class RailwayTicket
{
    String name, coach;
    int amt, totalamt;
        long mobno;
     void accept()
    {
      Scanner sc = new Scanner(System.in);
      System.out.println(" enter name");
      name = sc.next();
       System.out.println(" enter coach");
       System.out.println(" 1.) First_AC, 2) Second_AC, 3) Third_Ac,
                                    4.) Sleeper");
       System.out.println(" enter your option");
        coach = sc.next();
       System.out.println(" enter phone no");
        mobno = sc.nextLong();
       System.out.println(" enter amount");
       amt = sc.nextInt();
    }
    void Update()
    {
        System.out.println(" 1) First_AC, 2) Second_AC, 3) Third_Ac,
                                                    4.) Sleeper");
```

```
System.out.println(" if u are intrested to change urcoach");
System.out.println(" choose Option");
Scanner  s = new scanner (System.in);
int  op = s.nextInt();
if (op == 1)
 {
    totalamt = amt + 700;
 }
 else if (op == 2)
 {
    totalamt = amt + 500;
 }
  else if (op == 3)
 {
   totalamt = amt + 250;
  }
   else
   {
     total amt = amt;
   }
}
void display()
{
   System.out.println (name);
   System.out.println (coach);
   System.out.println( mobno);
   Syston.out.println( totalamt);
}
public static void main( String args[])
{
```

(3)

```
RailwayTicket r = new RailwayTicket();
    r.accept();
    r.update();
    r.display();
  }
}
```

4. Design a class to overload a function volume() as follows:

i) double volume(double r) – with radius 'r' as an argument, returns the volume of sphere using the formula:

$V = 4/3 \times 22/7 \times r^3$.

ii) double volume(double h, double r) – with height 'h' and radius 'r' as the assignments, returns the volume of a cylinder using the formula: $V = 22/7 \times r2 \times h$.

iii) double volume(double l, double b, double h) – with length 'l', breadth 'b' and height 'h' as the $V = l \times b \times h$.

```
public class Volume{
    public double volume(double r){
        return (4.0/3.0) * (22.0/7.0) * Math.pow(r,3);
    }
    public double volume(double h, double r){
        return (22.0/7.0) * Math.pow(r,2)*h;
    }
    public double volume(double l, double b, double h){
        return l*b*h;
    }
    public static void main(String[] args){
        Volume vol = new Volume();
        System.out.println(vol.volume(3));
        System.out.println(vol.volume(3,4));
        System.out.println(vol.volume(3,4,5));
    }
}
```