Java Assignment

19BQ1A05G4
~ Dinesh

Set-1

1. Write about the role of JVM, JAVA, API in developing Platform-independent java Program with suitable example.

A: The meaning of platform independent is that the java compiled code can run on all operating system

Role of JVM while executing JAVA Programs:

JAVA being a platform independent programming language doesn't work on one-step-compilation. It involves two-step execution, first through an Os independent compiles. and second JVM

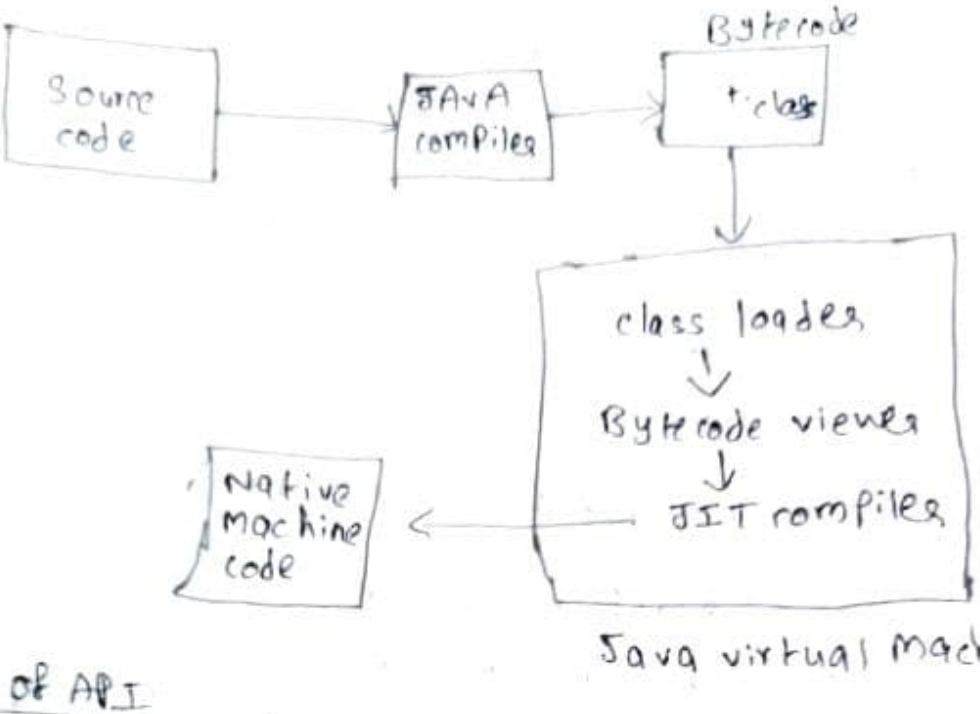
Classloader:

The main class is loaded into the memory by passing its Add two numbers. class file to the JVM, through involving the latter. All the others classes if references in the Programare loaded through the class loader.

Bytecode verifier:

After the Bytecode of the Add Two Numbers class is loaded by the class loaders. it has to be inspected by the byte code verifier.

3)
```
import java.io.*;
import java.util.*;
class RailwayTicket {
    String name, coach;
    long mobno;
    int amt, totalamt;
    Public void accept() {
    Scanner sc = new Scanner(System.in);
    System.out.println("enter name");
    name = sc.next();
    System.out.println("enter mobile number");
    mobno = sc.nextlong();
    System.out.println("enter coach type");
    coach = sc.next();
    System.out.println("enter basic amount of
                                    ticket");
    amt = sc.nextInt();
}
    Public void update() {
    if (coach.equals("first-AC"))
        total amt = amt+700;
    else if (coach.equals("second-AC"))
        total amt = amt+500;
    else if (coach.equals("Third-AC"))
        total amt = amt+250;
    else total amt = amt;
}
    public void display() {
    system.out.println("name:"+name);
```



Role of API

The full form of API is application Programming interface. it is a document which gives us the list of all the Packages, classes and interfaces along with their fields and methods, fields classes, interfaces Provided by java.

libraries

In JAVA most basic programming tasks are performed by the API's classes and Packages, which are helpful in minimizing the number of lines written within pieces of code. The Java API included with JDK.

```
System.out.println("coach:"+coach);
System.out.println("mobile number:"+mobno);
}
    Public static void main(String args[]){
        Railway Ticket r= new Railway Ticket()
          r.accept();
          r.update();
          r.display();
}
}
```

Output :

```
Enter name
 Pavan
Enter mobile no
  9656 361864
 Enter coach type
  Third-AC
Enter basic amount of tickets
   62
name: Pavan
coach: Third-AC
Mobile no: 9656361864
total amt: 312
```

4. Design a class to overload a function volume as follows

(i) double volume(double r). with radius r as an argument, return the volume of sphere using formula: v: 4/3 * 22/7 * r3

(ii) double volume(double h, double r) - with height h and radius r as the arguments, return the volume of cylinder using the formula:

$v = 22/7 * r^2 * h$

Code:

```
class overloading{
    double volume (double r){
    double v= (4.0/3)*(22.0/7)* r*r*r;
      return v;
    }
    double volume (double h, double r){
      double v= (22.0/7)* r*r*h;
       return v;
    }
    double volume(double l, double b, double h)
       double v= l*b*h;
       return v;
    }
Public static void main (string args[]){
    overloading obj = new overloading();
```

```
        double x = obj.volume();
        double y = obj.volume(10.05, 2);
        double z = obj.volume(8.1, 7.5, 9.5);
        System.out.println("volume of sphere:" + x);
        System.out.println("volume of cylinder:" + y);
        System.out.println("volume of cuboid:" + z);
    }
}
```

Output:

```
v of sphere : 951.1584761902762
v of cylinder : 1566.7034285743
v of cuboid : 144.85
```

Q. With an example explain the concept of classes and nested classes in Java programming.

Classes in Java:

A class is a blue print from which individual objects are created. A class contain any of the following variable types:

1) Local variables
2) Instance variables
3) class variables

Ex:

```
Public class Dog{
    string breed;
     int age;
    string colour;
    void barking(){
    }
    void hungry(){
    }
     void sleeping(){
     }
}
```

Local variables:

variables defined inside methods constructors or blocks are called local variables the variable will be declared and initialized with in the method and the variable will be destroyed when the method has completed

Instance variables: These are the variables within class but outside any method. These are initialised when the class is instentiate

class variables:

These are declared within a class, outside any method, with the static keyword

Some of the important topics that are discussed in classes are:

1. Constructors:

Every class has a constructor. If we don't explicitly write a constructor for a class, the java compiler builds a default constructor for that class. The main rule is that a constructor should have same name as the class. A class can have more than

2. Creating an object: An object is created from a class. In Java the new keyword is used to create new objects.

* Declaration - A variable declaration with a variable name with object type

* Instantiation: The 'new' keyword is used to create the object

* Initialisation: The 'new' keyword followed by a call to a constructor. This call initializes the new object

3. Accessing instance variables and methods

These are accessed by creating objects.

First create an object

Object Reference = new Constructor();

Now call a variable

object reference. variable name;

Now call a class method

Object Reference. methodname();

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 one place, this increases the use of encapsulation, and creates more readable and maintainable code

Inner classes.

In case of inner class without object existing, there cannot be an inner class. object i.e an Obj of inner class is always strongly associated with an outer class object. To instantiate an inner class, you must first instantiate outer class and then create the inner obj within the outer object