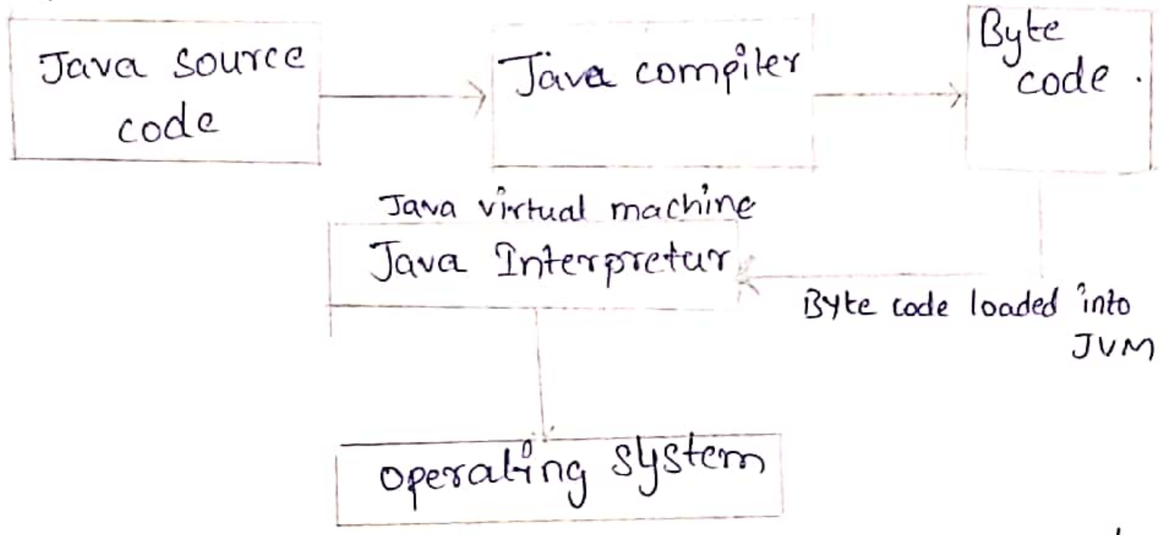
Assignment-1 JAVA PROGRAMMING SAI KRISHNA VINJAMURI
19BQ1A0505 | 2nd-CSE-D

Set-1

1) Write about the role of JVM, JAVA API in developing the platform independent java program with suitable example?

Ans. The Java Virtual Machine (JVM) is an abstract computing machine or virtual machine interface that drives the java code. When we compile a Java program then byte code is generated. Byte code is the source code that can be used to run on any platform. Bytecode, is an intermediary language between java source and the host system. it is the medium which, compiles java code to bytecode. which gets interpreted on a different machine and hence it makes it platform/operating system independent



Java is called platform independent because of Java virtual machine, when we submit a .class file to any operating system, JVM interprets the bytecade into machine level language

* JVM is responsible for allocating the necessary memory needed by the java program.
* JVM is responsible for deallocating memory space

API :- An application programming Interface (API) in the context of java, is a collection of prewritten packages, classes and interfaces with their respective methods, fields and constructors. similar to a user interface which faciliates interaction between humans and computers an API servers as a software program interface faciliating interaction example :- for example processing reference is an API in the classes and functions we used to write processing code similary, the java API is the classes and functions we used to write processing code similarly, the java API is the list of classes and functions we use to write java codes. The point is that an API is a collection of things we can do when writing code

example for JVM :- for example, If we are running mac os we will have a different JVM than if we are running windows os some other operating system this can be verified while downloading the JDK.. which gives a list of targeted files Hence, we conclude that the programming language we write in an JDK is same, while the JDK file we use is platform dependent. Therefore, JVM is platform dependent and java is platform independent

// Resources :- geeks forgeeks / w3schools

19BQ1A0505

2 With an example program explain the concept of classes and nested classes in java ?

Ans:- Class:- A class is a user defined blueprint or prototype from which objects are created . it represents the set of properties or methods that are common to all objects of one type .

The components of classes are :-

1) Modifiers :- A class can be public or has default access

2) Class name :- The name should begin with a initial letter

3) Super class :- The name of the class's parent (super class) if any , preceded by the keyword extends . A class can only extend (sub class) one parent

4) Body :- The class body surrounded by braces , { }

eg:-
```
import java.util. Scanner;
  class Shape {
      int numberOf Sides;
      Public void Rectangle shape (numberof sides) {
              this. numberof sides = number of sides ;
              System.out.println(" the polygon has many sides "+
                                                  number of sides )

      }

  } ~~eta~~
  class Main {
      Public static void main(String []args) {
           Scanner s = new scanner (System.in);
           int Sidenumber = s. nextInt ();
           Shape sh = new shape ( )
           sh. Rectangle shape (Sidenumber);
      }
  }
```

Scanned with CamSc

here class shape is the blueprint and sh is the working model of blueprint.

Nested classes:-
In java, it is possible to define a class within another class, such classes are known nested classes

Syntax:-

```
class OuterClass{
    ----
    ---
    class Nestedclass{
        ---
        ---.:
    }
}
```

Nested classes are divided in two categories:
1) Static nested class: Nested classes that are declared static.
2) inner class: An inner class is non-static nested class of an outer class.

Eg:- // To access static members of an outer class.

```
class Outerclass{
    static int outer_x =10;
    int outer_y =20;
    private static int outer_private =30;
    static class StaticNestedClass{
        void display(){
            System.out.println("outer_x = " + outer_x);
            System.out.println("outer_private = "+ outer_
                                                private);
            }
        }
    }
Public Class StaticNested Demo {
        Public static void main (String [] args) {
                Outerclass. StaticNestedClass. nestedObject =
                                new Outerclass-Static NestedClass() ;
                nestedObject.display () ;
        }
    }
```

output:

```
outer_x = 10
Outer_private = 30
```

// To ascess non-static for inner class.

```
Class Outer class
{
    static int outer_x = 10;
     int outer_y = 20;
    private int outer_private = 30;
    class Inner class {
        void display () {
            System.out.println("outer_x = "+ outer_x);
            System.out.println ("outer_Y = "+ outer_Y);
            System.out.println("Outer_private = "+ outer_private);
        }
    }
```

7.

```
Public class InnerClassDemo{
    public static void main(String [] args) {
        OuterClass.outer object = new Outerclass();
        OuterClass.InnerClass.innerobject = outerobject.
                                        newInnerclass();

        ~~In~~ innerobject.display();

    }

}
```

<u>Output</u>:-

```
Outer_x = 10
Outer_y = 20
Outer_Private = 30
```

//Resources:- geeksforgeeks (website), w3Schools (website)

3) Design a class RailwayTicket with the following description
Instance variables/data members:-
String name: to store name of customer
String coach: to store type of coach
long mobno:- to store customer's mobile number
int amt: to store basic amount of ticket
int totalamount: to store the amount to paid after updating the amount

Methods:-
void accept(): to take input for name, coach, mobilenumber and coach
void update(): To update the amount as per the coach selected. Extra amount to be added as follows

TYPE OF COACHES	AMOUNT
First – AC	700
Second – AC	500
Third – AC	250
Sleeper	None

void display():
To display all details of a customer such as name, coach, total amount and mobilenumber

Write a main() method to create an object of the class and call the above methods program:-

```
import java.util.Scanner;
class Ticket{
    private String name;
    private String coach;
    private long mobileNumber;
    private int amount;
    int totalamount;
    public Ticket(String name, String coach, long mobileNumber
                                    ,int amount){

        this.name = name;
        this.coach = coach;
        this.mobileNumber = mobileNumber;
        this.amount = amount;
    }
    public String getName(){
        return name;
    }
    public String getCoach(){
        return coach;
    }
    public long getMobileNumber(){
        return mobileNumber;
    }
    public int getAmount(){
        return amount;
    }
    public void accept() {
          name = getName();
          coach = getCoach();
          mobileNumber = getMobileNumber();
          amount = getAmount();
    }
    public int update() {
          if(coach.equals("First_AC"))
               totalAmount = amount + 700;
          else if(coach.equals("Second_AC"))
               totalAmount = amount + 500;
          else if(coach.equals("Third_AC"))
               totalAmount = amount + 250;
          else if(coach.equals("~~amount~~sleeper"))
               totalAmount = amount;
          else
               System.out.println("Choose valid coach");
          return totalAmount;
     }
     public void display(){
          System.out.println("Name : "+getName() +
               " \ncoach : "+ getCoach()+" \n amount : "
             + getAmount()+" \n Total amount : "+update());

     }

}
```

```
public class RailwayTicket{
    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your details as follows
        \n1) Your name \n2)choose your coach -+
          "\n First_AC. \n Second_AC. \n Third_AC
          \n sleeper \n 3)mobile number \n
          4) Basic amount = 150 ");

        Ticket s = new Ticket(~~sc~~ sc.nextLine(), sc.nextLine()
                    , sc.nextLong(), sc.nextInt());

        s.accept();

        s.update();

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

(ii) double volume(double h, double r) - with height 'h' and radius 'r' as the arguments and returns the volume of cylinder using $v = 22/7 \times r^2 \times h$

(iii) double volume(double l, double b, double h) - with length 'l', breadth 'b', height 'h' and returns the volume of cuboid using $V = l \times b \times h$.

program:-

```
class Input{
    double volume(double r)
    {
           return  ((4/3) * (22/7) * (r*r * r));

    }
    double volume( double h, double r) {
           return  ( (22/7) * (r* r) * h );

    }
    double volume (double l, double b, double h) {
           return ( l * b * h ) ;

    }
}
public class Overload {
    public static void main(String [ ]args){
         Input sc = new Input();
         System.out.println(sc.Volume(4.0));
         System.out.println(sc.Volume (1.3, 3.3)) ;
         system.out.println(sc.volume(1.3, 4.54,2.4));
    }

}
```