Regd No: 19B01A0549
Name: Sk. Reshma
Sec: CSE-D

Set 4

What are the components of JAVA platform? Explain. Write a java program to illustrate the usage of conditional statements and looping statements

The java platform has two components. They are

1 The Java Virtual Machine (JVM)
2 The Java ~~(Pr)~~ Application Programming Interface (API)

Java Platform:

It is a collection of programs that helps programmers to efficiently develop and run Java applications. It includes an execution engine, a compiler and a set of libraries in it. It is a set of computer software and specifications. James Gosling developed the java platform at Sun Microsystems and it was later acquired by the Oracle corporation.

JVM:

It is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. Simply we can say that the main work of JVM is to

load & execute the byte code & gives the output
JVM is a virtual / abstract machine It has it's own
OS & memory
It is divided into 3 units They are:

- Class loader
- Memory area
- Execution engine

After executing the filename command a JVM instance will be created, the byte code will be loaded into class loader. The work of classloader is to load, link & initialise

→ Load is classified into 3 components

1. Bootstrap class loader - to load all pre-defined classes
2. Extension class loader - load external libraries
3. Application class loader - loads our application

→ Link is also classified into 3 components

Verify - verifies malicious code
Prepare - the memory is allocated to static methods, variables with a default value
Resolve - here the original values will be assigned
→ symbols resolves to original references
Initialise - The original values will be assigned

Memory area is classified into 5 parts.

Method area - which stores the class level data

In before days, the default memory area is 64MB upto Jdk 1.8. It has static blocks, variables, methods

Now, it is called as meta space / permaGen & the memory is allocated unlimited.

Heap - It stores object level variables, objects

Stack - It stores local variables, running methods, per thread

Pc register - It stores the location of the next instruction which means where it has to go.

JN area - Whenever we write the code in other programming languages we use it.

Enterprise Edition

Interpreter - It is used to execute the code.

It has JIT compiler, Garbage collector, Security manager etc

The main function of JIT compiler is whenever a method is executed repeatedly then it identifies it, it's interpretered & again doesn't send it to the interpreter because it only converts into machine code.

→ JN interface is used to interact with JN libraries. & it will executes the code

Whenever we are using external library, the next it will go to memory area

JIT - The method which repeats more, it takes it & stores & give output fast ie, compile it fast

→ The Developed Java code isn't executed on our physical machine. It'll be executed on its own run-time environment. So this is called as a platform

→ JAVA is platform independent but JVM is platform dependent.

JAVA API :

The API is a large collection of ready-made software components that provides many useful capabilities. It is groped into libraries of related classes and interfaces, these libraries are called packages.
These are java predefined libraries.

→ JAVA has the following conditional statements

- if
- if - else
- nested - if
- else - if
- switch

if statement : It is used to decide whether a block of statements or not a statement will be executed i.e, if the condition is true then the block of statements is executed otherwise not.

Program:

```
class If
{
  public static void main(String args[])
  {
     int n=50;
     if (n%5 ==0)
           System.out.println ("50 is a multiple of 5");
     System.out.println ("Out of block");
  }
}
```

if-else : It is used when a condition is true then it will executes block of statements otherwise it executes other block of statements when it is false.

Program:

```
class ifElse
{
  Public static void main(String args[])
  {
    int n=15;
    if (n%3==0)
         System.out.println ("It is a multiple of 3");
    else
         System.out.println ("Not a multiple of 3");
  }
}
```

Nested-if : It is used when an if statement is inside another if statement

```
class NestedIf
{
    public static void main(String args[])
    {
        int n=7;
        if (n>0)
        {
            if (n<10)
                    System.out.println(n+" is less than 10");
            if (n>5)
                    System.out.println(n+" is greater than 5");
            else
                System.out.println (n+" is greater than zero and
                                    less than 5");
        }
    }
}
```

Else if ladder:

```
class ElseIf
{
  Public static void main (String args[])
  {
    int n=10;
   if (n<10)
          System.out.println ("n is less than 10");
  else if (n>10)
          System.out.println ("n is greater than 10);
  else
      System.out.println ("n is equal to 10);
  }
}
```

Switch Case: It is a multi-way branch statement

```
class SwitchCase
{
  public static void main (String args[])
  {
   int i=1;
   switch (i)
   {
    Case 0:
          System.out.println ("i is zero");
           break;
    Case 1:
          System.out.println ("i is one");
            break;
    Case 2:
            System.out.println ("i is two");
             break;
     default:
            System.out.println ("i is greater than 2");

    }
  }
```

→ looping statements are the statements that execute one or more statement repeatedly several no. of times

for-loop : It is used when you know exactly how many times you want to loop a block of code.

Syntax : for (initialization condition; test condition; increment/decrement)

```
         {
            statement(s);
         }
```

Program :

```
        class ForLoop
        {
          public static void main (String args[])
          {
           for (int i=1 ; i<4 ; i++)
                System.out.println (" 3*" + i + " = " + 3*i);
          }
        }
```

While loop: A while loop iterates through a set of statements till its boolean condition returns false i.e., when we don't know the exact no of iterations

Syntax:
```
while (boolean condition)
{
   loop statements
}
```

Program:
```
class WhileLoop
{
 public static void main (String args[])
 {
  int n = 56, S=0;
  while (n>0)
  {
     int r = n%10;
       S+ = r;
       n= n/10;
  }
  System.out.println ("Sum of digits of 56 is" +s);
 }
}
```

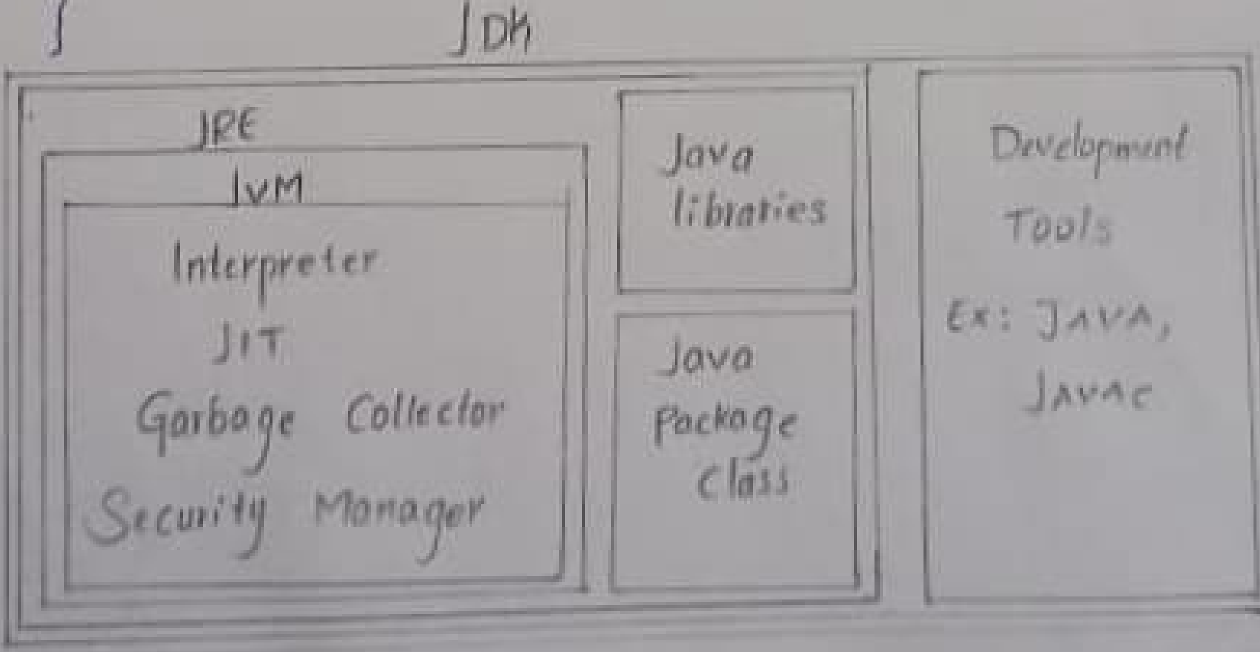
do-while : It is similar to a while loop, except that it will definitely executes atleast once. It is an exit-controlled loop.

Syntax :
```
do
{
 statements
} while (condition);
```

Program :
```
class DoWhileLoop
{
 public static void main(String args[])
 {
   int i=15;
   do
   {
    System.out.println(" value of i=" +i);
      i++;
   }while (i<10);

 }

}
```



Write any six significant differences between Procedure Oriented Programming and Object Oriented Programming. Why JAVA is Robust programming language? Explain

PROCEDURAL ORIENTED PROGRAMMING	OBJECT ORIENTED PROGRAMMING
In procedural programming, program is divided into small parts called functions.	In object oriented programming, program is divided into small parts called objects
Procedural programming follows top down approach	Object oriented programming follows bottom up approach
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like public, private, protected etc
Adding new data and function is not easy	Adding new data and function is easy.
It ~~has~~ doesn't have any proper way for hiding data so it is less secure	It provides data hiding so it is more secure
In this programming overloading isn't possible	Overloading is possible in this programming.

In this function is more important than data	In this data is more important than function
In Procedural programming is based on unreal world	In object oriented programming is based on real world.
Examples: C, FORTRAN, Pascal, Basic etc	Examples: C++, Java, Python etc

Java is robust because it uses strong memory management. There is automatic garbage collection in java which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore. There are exception handling and the type checking mechanism in java

Define a class ParkingLot with the following description:
Instance Variables / data members:
int vno - To store the vehicle number
int hours - To store the no. of hours the vehicle is parked in the parking lot
double bill - To store the bill amount
Member methods:
void input () - To input and store vno & hours
void calculate() - To compute the parking charge at the rate of Rs 3 for the first hour or part

thereof, and Rs. 1.50 for each additional hour or part thereof.

void display() - To display the detail

Write a main method to create an object of the class and call the above methods

```
import java.io.* ;
import java.util.Scanner;
public class ParkingLot
 {
     Private int vno;
      private int hours;
      private double bill;
      public void input()
       {
        Scanner sc = new Scanner (System-in);
         System.out.println ("Enter vno:");
          vno = sc.nextInt();
         System.out.println ("Enter no.of hours:");
          hours = sc.nextInt();
       }
      public void calculate()
      {
         if(hours ==1)
                  bill = hours*3;
         else
               bill = 3+(hours-1)*1.5;
      }
```

```
    public void display() {
        System.out.println("Vehicle No is: " + vno);
        System.out.println("No of hours is: " + hours);
        System.out.println("Parking charges is: " + bill);
    }
   Public static void main (String[] args)
    {
      ParkingLot pl = new ParkingLot();
        pl.input();
        pl.calculate();
        pl.display();
    }
  }
```

Define a class to overload a function JoyString() as follows
(i) void JoyString (String s, char ch1, char ch2) with one String & 2 Character arguments that replaces the character argument ch1 with the character argument ch2 in the given String s and prints the new string

Example:

Input value of s = "TECHNALAGY"

Ch1 = "A"

Ch2 = "O"

Output : "TECHNOLOGY"

(ii) void JoyString (String s) with one string argument that prints the position of the first space and the last space of the given String s.

Example:

Input value of = "Cloud Computing means Internet based Computing"

First Index : 5

Last Index : 36

(iii) void JoyString (String s1, String s2) with two string arguments that combine the two strings with a space between them and prints the resultant string

Example :

Input value of s1 = "COMMON WEALTH"
s2 = "GAMES"

Output : "COMMON WEALTH GAMES"

```
import java.io.*;
import java.util.Scanner;
class OverLoad
{
  string s1,s2,s;
   Char ch1,ch2;
```

```
public void Joystring (String s, char ch1, char ch2)
{
  for (int i=0; i<s length(); i++)
   {
      if (s.charAt(i)== ch1)
      {
        s=s.replace(ch1,ch2);
      }
     System.out.println(s);
   }
public void Joystring (String s)
 {
     int  FirstIndex=0,  LastIndex=0;
      for (int i=0;  i<s.length(); i++)
      {
        if(s.charAt(i)==' ')
        {
            FirstIndex=i;
             break;
        }
      }
      LastIndex = s.LastIndexOf(' ');
    System.out.println ("First Index:  " + FirstIndex);
   System.out.println ("Last Index:  " + LastIndex);
  }
Public void  Joystring (String s1, String s2)
  {
    System.out.println (s1+"  "+s2);
  }
}
Public  class  Overload
  {
    Public  static  void  main  (String[] args)
     {
       Overload   ol = new  Overload ();
        ol.Joystring  ("Technalogy", "a", "o");
        ol.Joystring ("Cloud  Computing  means  internet
                               based  computing");
       ol.Joystring  ("Common  wealth", "Games");
     }
}
```