Java Assignment - 1...

1. What are the components of Java plat-form? Explain? Write a java program to illustrate the usage of conditional statements and looping statements?

Sol:- JAVA platform:-

Java platform is a software or collection of programs that help us to execute applications that help us to written in Java programming language. A Java platform consists of Java compiler, a set of libraries, and an execution engine.

Java platform is independent of any particular OS which makes Java programming language a platform-independent language.

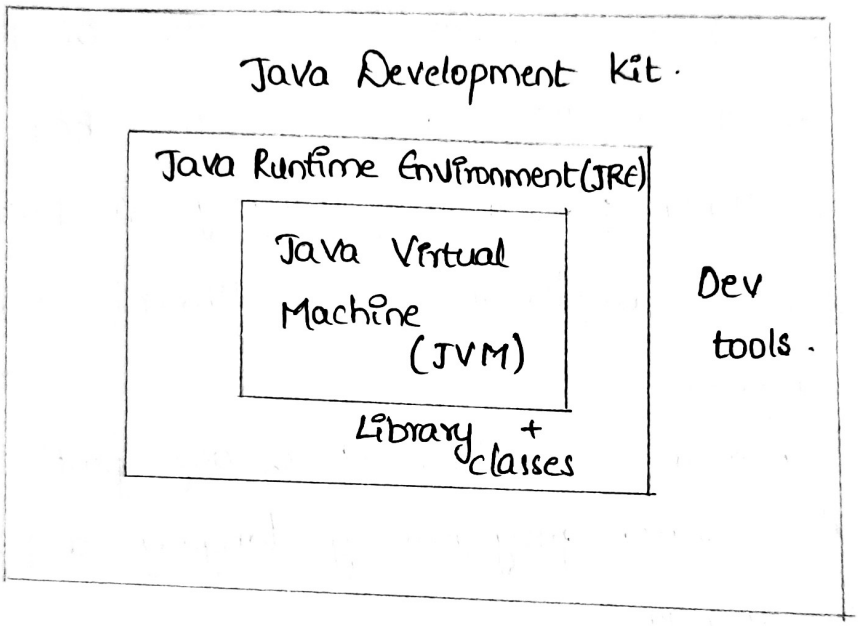
Java platform consists of the following components:-

- Java language.
- The Java Development kit (JDK).
- The Java Runtime Environment (JRE).
- The Java Compiler.
- The Java Virtual Machine (JVM)

→ A part from the above main components, the Java

platform also contains garbage collectors, a set of libraries and other additional components and tools that are required to efficiently run the Java applications.

diagram that shows relationship:-



Conditional statements in Java are:-

→ if
→ if -else
→ Nested -if
→ if-else -if
→ Switch case.

(i) if Case:- If we use if statement in a progeam if it is true the block will be executed otherwise

the block cannot be executed. (It is skipped).

Ex:

```
class IfDemo
{
  Public static void main (String args[])
  {
      int a = 20;
      if (a % 2 == 0) {
        System.out.println ("It is an even number");
      }
    System.out.println ("It is not an even number");
  }
}
```

(ii) if-else- this is used when a condition is true it executes the block of if. Otherwise it executes else block.

Ex:

```
class IfelseDemo
  {
   Public static void main (String args[])
   {
      int a = 20;
      if (a % 20 == 0) {
        System.out.println ("It is an even number");
```

```
    else {
        System.out.println ("It is an odd number");
        }
    }
}
```

(iii) Nested - if :- Nested if statements means an if statement inside an if statement.

Ex:

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

(iv) if-else ladder:-

Ex:

```
class ifelseif
{
  Public static void main (String args[])
   {
     int i = 20;
     if(i==10){
          System.out.println("i is 10") }
    elseif (i==15) {
          System.out.println("i is 15") }
     else {
        System.out.println("i is not present");
        }
   }
}
```

(v) Switch Case :- It is a multi-way branching statement

Ex:

```
class SwitchCase
   {
     Public static void main(String args[])
     {
       int i=b;
       Switch(i)
       {
         Case 1:
              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 statements repeatedly several no.of times.

(i) for loop :-

It is used when the no.of repeatitions are known.

Syntax:

```
for (initialize condition ; test condition ; increment,
                                              decrement)
{
    statement (s-1);
}
```

Ex:
```
class for-loop
{
  public static void main (String args[])
  {
    for(int i=1 ; i<5 ; i++){
      System.out.println(" 5* " +i+ " = " + 5*i);}
  }
}
```

(ii) While loop:- It is used when no.of iterations are not known.

Ex:
```
class While-if
  public static void main (String args[]) {
    int i=1;
    while (i<=10){
      System.out.println(i);
      i++;
    }
  }
}
```

(iii) do-while:- do-while is similar to while loop but here the body of loop is executed. Then it checks the exp statement (condition).

do while example:-

```
class do-while {
    public static void main (String args[])
    {
      int i = 11;
      do
      {
        System.out.println ("Value of i = " + i);
          i++;
      }
       while (i <10);
    }
}
```

2. Write any six significant differences between procedure Oriented programming and object Oriented programming. Why Java is Robust progeamming language? Explain?

A:-

Procedural Oriented progeamming	Object oriented progeamming.
1. It deals with algorithms.	1. It deals with data.
2. Programs are divided into functions.	2. progeams are divided into objects.
3. It is top down approach	3. It is Bottom Up approach.
4. It needs less memory	4. It needs more memory

5. Ex:- C, Fortan	5. Ex:- C++, JAVA.
6. It do not have any access specifiers.	6. It has access specifiers like private, public and protected.
7. It is less secure.	7. It is more secure.

* Java is a "Robust" because it is highly supported language. It is portable across many operating Systems. Java also has a feature of Automatic memory manag-ement and garbage Collection Strong type checking mechanism of Java also helps in making Java Robust. Bugs, especially system crashing bugs, are very rare in Java. It is portable across many languages. It is also Strong type Checking.

3. Define a class ~~package~~ parkingLot with following description:

Instance variables / data members.

int vno - To store the vehicle number.

int hours - To store the number 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 and hours

Void Calculate() - To Compare the parking charge at the rate of 3Rs. for the first hour or part there of, and Rs. 150 for each additional hour or part there of

Void display() - To display the detail.

Write a main method to Create an object of the Class and Call the above methods.

A:- Program:-

```
import java.util.*;
Class ParkingLot {
    Private int Vno, hours;
    double bill;
    Public Void Input()
    {
     Scanner sc = new Scanner (System.in);
     System.out.println("Enter Vehicle Number");
     Vno = sc.nextInt();
     System.out.println("Enter Number 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 Number" + vno);
   System.out.println("Number of Hours" + hours);
   System.out.println("Parking Charges" + bill);
}
Public static void main()
{
      ParkingLot p = new ParkingLot();
      P.Input();
      P.Calculate();
      P.display();
}
}
```

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

(i) Void Joystring (String s, char ch1, char ch2) with one string and two charachter arguments that replaces the charachter argument ch1 with charachter 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.

Ex:

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

First Index : 5

Last Index : 36.

(iii) Void Joystring (String s_1, String s_2) with 2 String argum-ents that combine the two strings with a space

between them and prints the resultant string.

Example:-

Input Value of S1 = "COMMON WEALTH".

S_2 = "GAMES"

Output : "COMMON WEALTH GAMES".

A:- Program:-

```
class Overload {
    Public Static void joyString (String s, charch1, charch2){
        S= S.replace (ch1, ch2);
        System.out.println (s);
    }
    Public Static void joyString (String s) {
        int first = S.indexOf (" ");
        int last = S.lastIndexOf (" ");
        System.out.println(" First Index : "+ first );
        System.out.println(" Last Index : " + last );
    }
    Public Static void joyString (String S1, String S2) {
        String S = S1 +" " + S2;
```

```
        System.out.println(s);
    }
}
```