**Implementation No. 1**

Date: 24/08/2020

Name: **Tushar Nankani** Roll No: **1902112** Batch: **C23**

**AIM:** Basic Java Programs

**THEORY:**

1. public static void main:

**public** - This is the access modifier of the main method. It has to be public so that java runtime can execute this method.

**static** - When java runtime starts, there is no object of the class present. That’s why the main method has to be static so that JVM can load the class into memory and call the main method. If the main method won’t be static, JVM would not be able to call it because there is no object of the class is present.

**void** - Java programming mandates that every method provide the return type. Java main method doesn’t return anything, that’s why it’s return type is void.

**main** - This is the name of java main method. It’s fixed and when we start a java program, it looks for the main method.

1. Compile and Run Java Program:

Save the file as `file\_name.java`. We should always name the file same as the public class name. In the program, the public class name is `file\_name`, file name should be `file\_name.java`.

Open CMD, and change directory to where the `.java` file is saved, to **Compile,** type: javac file\_name.java

After compilation the .java file gets translated into the .class file(byte code). To **Run:** java file\_name

1. Features of Java:

* Java is simple, object-oriented, portable, platform independent, secured, interpreted, high performance, multithreaded, distributed & dynamic.
* Java syntax is based on C++ (so easier for programmers to learn it after C++).
* Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
* There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

1. JVM (Java Virtual Machine) Architecture:

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).

The JVM performs following operation:

* Loads code
* Verifies code
* Executes code
* Provides runtime environment

The JVM provides definitions for the:

* Memory area
* Class file format
* Register set
* Garbage-collected heap
* Fatal error reporting etc.

**PROGRAMS:**

1. Write a program to display Student’s Details on the screen.

**CODE:**

class Student

{

    public static void main(String args[])

    {

        int RollNo = 1209112;

        System.out.println("My name is Tushar Nankani.");

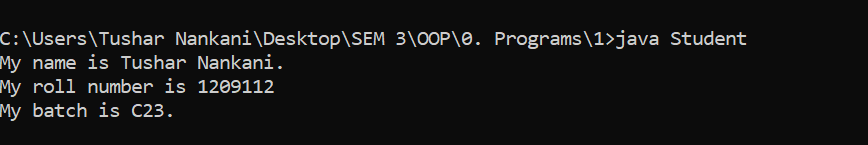
        System.out.println("My roll number is " + RollNo);

        System.out.println("My batch is C23.");

    }

}

**OUTPUT:**

****

1. **Write a program to add numbers, accepted via the command line.**

**CODE:**

class Sum

{

      public static void main(String ar[])

      {

       int x,y,s;

       x = Integer.parseInt(ar[0]);

       y = Integer.parseInt(ar[1]);

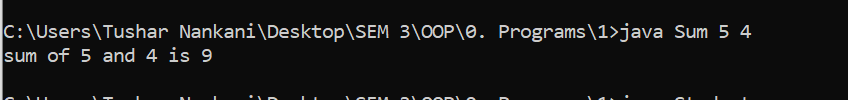
       s = x + y;

       System.out.println("sum of " + x + " and " + y +" is " + s);

      }

}

**OUTPUT:**

****

1. **Write a program to accept Student’s details and display on the Screen.**

**CODE:**

import java.io.\*;

class Example

{

    public static void main(String args[])throws IOException

    {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter Name:");

        String name = br.readLine();

        System.out.println("Enter your roll no:");

        int RollNo = Integer.parseInt(br.readLine());

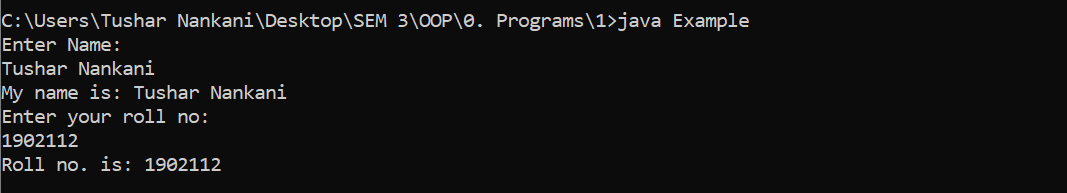
        System.out.println("My name is: " + name);

        System.out.println("Roll no. is: " + RollNo);

    }

}

**OUTPUT:**

****

1. **Write a program to calculate maximum of 3 numbers using conditional operators.**

**CODE:**

class Largest {

    public static void main(String ar[])

    {

        int x,y,z;

        x=Integer.parseInt(ar[0]);

        y=Integer.parseInt(ar[1]);

        z=Integer.parseInt(ar[2]);

        int max = ((x > y) ? ((x > z) ? x : ((y > z) ? y : z)) :

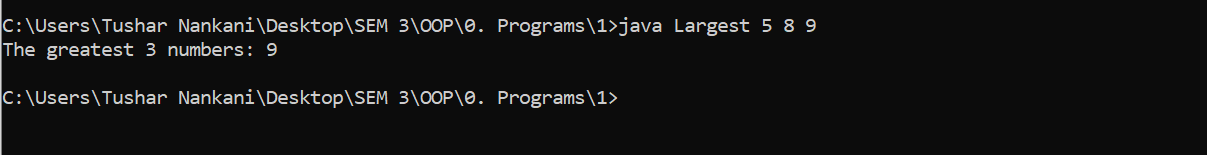
((y > z) ? y : ((z > x) ? z : x)));

        System.out.println("The greatest 3 numbers: " + max);

    }

}

**OUTPUT:**

****