Aman Mukesh Agrawal

2020BTECS00074

Programming Lab – 2

Assignment-1

Write steps for Installation of jdk package. Write procedure to set environment variable PATH.

* Steps to install JDK:
  + Step 1: Download JDK from the Site

Go to the Oracle site and open the Java SE download page. Under the latest version of Java Platform, Standard Edition, click on the JDK download button.

* + Step 2: Install the JDK exe File
* In this step, we will be running the executable JDK file (It will be a file with .exe as an extension) once the download is done. This installs JDK as well as JRE. For running this file on Windows, we will need Administrator rights.
* To begin the installation, we need to double-click on the downloaded file, and we will be presented with the below window.
* Click on Next to proceed with the installation, and follow the Installation guide provided for any queries.
* Click on the Close button once the installation has finished.
* To recover some of our system’s disk space, it is good practice to delete the downloaded exe file once the download has been done.
  + Step 3: Check the Directory
* JDK gets installed in the C directory of our system by default having the path “C:\Program Files\Java\jdk-11.0”. If we make any change to this path at all, we need to make a note of it as it will be required in the upcoming steps.
* Steps to set environment variable PATH:
  + Step 1: Go to the JDK bin folder to copy the file location path.
  + Step 2: Right click on my PC and select properties.
  + Step 3: Select the Advanced System at the left hand panel
  + Step 4: Choose the option “environment variables”.
  + Step 5: Click on New, type PATH in the Variable Name, and paste the path of the bin of installed JDK in the Variable Value field
  + Step 6: Click on the OK button to apply the changes.

1. Describe JDK, JRE, and JVM.

* JDK: The Java Development Kit (JDK) is a cross-platform software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the [**JVM (Java Virtual Machine)**](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/) and the JRE (Java Runtime Environment).
* JRE: Java Runtime Environment (JRE) is an open-access software distribution that has a Java class library, specific tools, and a separate JVM. JRE is one of the interrelated components in the Java Development Kit (JDK). It is the most common environment available on devices for running Java programs. Java source code is compiled and converted to Java bytecode. If you want to run this bytecode on any platform, you need JRE. The JRE loads classes check memory access and get system resources. JRE acts as a software layer on top of the operating system.
* JVM: JVM (Java Virtual Machine) acts as a run-time engine to run Java applications. JVM is the one that actually calls the main method present in a java code. JVM is a part of JRE (Java Runtime Environment).

Java applications are called WORA (Write Once Run Anywhere). This means a programmer can develop Java code on one system and can expect it to run on any other Java-enabled system without any adjustment. This is all possible because of JVM.

When we compile a *.java* file, *.class* files (contains byte-code) with the same class names present in *.java* file are generated by the Java compiler. This *.class* file goes into various steps when we run it. These steps together describe the whole JVM.

1. Differentiate between C, C++, and JAVA.



1. Explain simple hello word program in Java. Explain compilation and execution of java program.

* Program:

public class hello

{

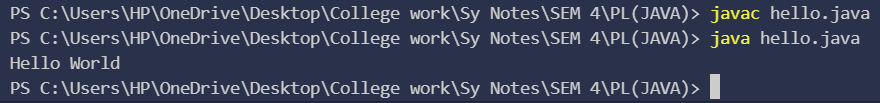
    public static void **main**(String[] *args*) {

        System.out.**println**("Hello World");

    }

}

Output:



Explanation:

Here, we have written a program for printing “Hello World” on terminal. Then, we have to compile the code by typing “javac hello.java” on terminal. And for executing it, we type “java hello.java”.

About Program: This program is the simplest program of java language. The hello world program consist of 3 primary components:

* 1. Class definition: a class called “hello” is defined.
  2. Class method: In java language, every program must contain a main method as shown in the code. (public static void main(String[] args)

Public: this is access modifier of the class and JVM can execute the method from anywhere.

Static: The main method is to be called without an object. The modifiers public and static can be written in either order.

Void: The main method doesn’t return anything.

Main(): name of method (must for every program in JVM)

String[]: The main method accepts a single argument, i.e., an array of elements of type String.

The next code:

System.out.println("Hello, World");

This line outputs the string “Hello, World” followed by a new line on the screen. Output is accomplished by the built-in println(*)* method. The System is a predefined class that provides access to the system, and out is the variable of type output stream connected to the console.

Compilation and Execution of the program:

* After successfully [setting up the environment](https://www.geeksforgeeks.org/setting-environment-java/), we can open a terminal in both Windows/Unix and go to the directory where the file – hello.java is present.
* Now, to compile the hello program, execute the compiler – javac, to specify the name of the source file on the command line, as shown:

javac hello.java

* The compiler creates a hello.class (in the current working directory) that contains the bytecode version of the program. Now, to execute our program, JVM(Java Virtual Machine) needs to be called using java, specifying the name of the class file on the command line, as shown:

java hello

1. Write java program for reading input of various data types from user using scanner class.

Program:

import java.util.Scanner;

public class datatypes {

    public static void **main**(String[] *arg*){

        Scanner sc = new **Scanner**(System.in);

        System.out.**print**("Enter your age:");

        int a = sc.**nextInt**();

        System.out.**println**("Your Age:"+a);

        System.out.**print**("Enter your Percentage:");

        float c = sc.**nextFloat**();

        System.out.**println**("Your Percentage:"+c);

        System.out.**print**("Enter a Big Decimal:");

        double dob = sc.**nextDouble**();

        System.out.**println**("Your Answer:"+dob);

        System.out.**print**("Enter your Initial:");

        char i = sc.**next**().**charAt**(0);

        System.out.**println**("Your initial:"+i);

        System.out.**print**("Are U above 18:");

        boolean b = sc.**nextBoolean**();

        System.out.**println**("Your Answer:"+b);

        System.out.**print**("Enter a long number:");

        long l = sc.**nextLong**();

        System.out.**println**("Your Answer:"+l);

        System.out.**print**("Enter a short number:");

        short s = sc.**nextShort**();

        System.out.**println**("Your Answer:"+s);

        System.out.**print**("Enter a byte:");

        byte by = sc.**nextByte**();

        System.out.**println**("Your Answer:"+by);

        System.out.**print**("Tell about yourself:");

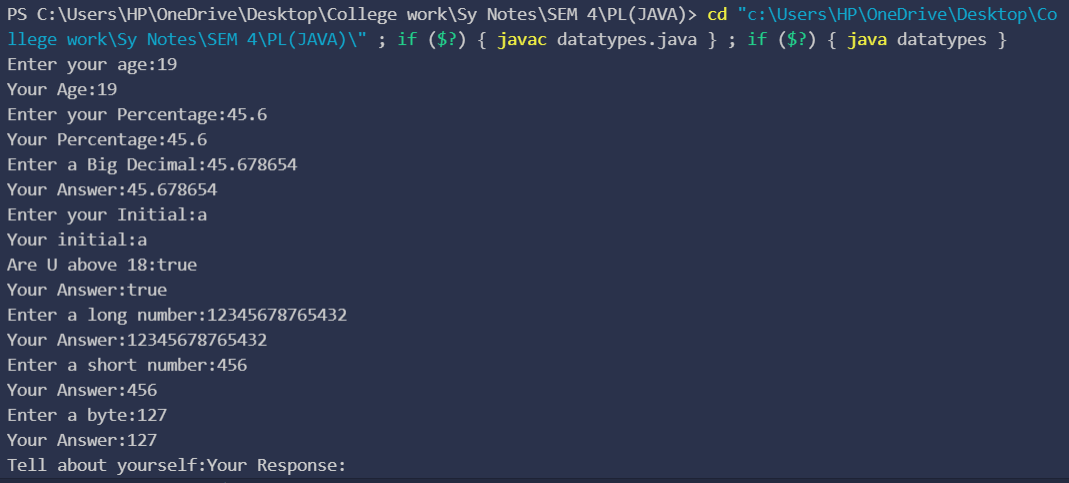
        String str = sc.**nextLine**();

        System.out.**println**("Your Response:"+str);

        sc.**close**();

    }

}

Output: 

1. Write a Java program to convert seconds to hour, minute and seconds.

Program:

import java.util.Scanner;

public class clock {

    public static void **main**(String[] *args*){

        Scanner scan = new **Scanner**(System.in);

        System.out.**print**("Enter the time in seconds:");

        int sec = scan.**nextInt**();

        int hour = sec/3600;

        sec%=3600;

        int min = sec/60;

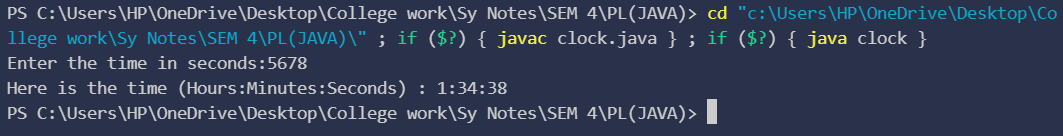
        sec%=60;

        System.out.**println**("Here is the time (Hours:Minutes:Seconds) : "+hour+":"+min+":"+sec);

        scan.**close**();

    }

}

Output: 

1. Write a Java program to check if there is a 10 in a given array of integers with a 20 somewhere later in the array.

Program:

public class Arr {

    public static void **main**(String[] *args*){

        int[] arr ={3,10,2,3,20,5};

        int n=0;

        for(int i =0;i<arr.length;i++){

            if(arr[i]==10){

                n=1;

            }

            if(n==1){

                if(arr[i]==20){

                    n=2;

                    System.out.**println**("The sequence '..10...20..' is there");

                }

            }

        }

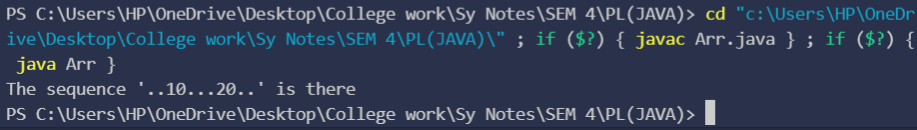
        if(n!=2){

            System.out.**println**("The sequence '..10...20..' is not there");

        }

    }

}

Output: 

1. Write a program to calculate the factorial of a number. (The number is passed as the command-line argument whose factorial we need to calculate)

Program:

public class factorial {

    public static void **main**(String *args*[]){

            int i=0,fac = 1,num;

            num = Integer.**parseInt**(args[0]);

            for(i=num;i>0;i--){

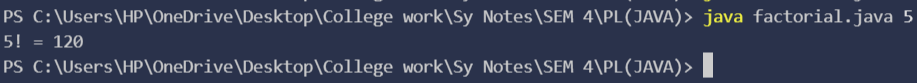
                fac\*=i;

            }

            System.out.**print**(num+"! = "+ fac);

        }

}

Output: 

1. Write a Java Program to find transpose of Matrix.

Program:

import java.util.Scanner;

public class transpose {

    public static void **main**(String[] *args*){

        int[][] matrix = new int[3][3];

        Scanner sc = new **Scanner**(System.in);

        for(int i=0;i<3;i++){

            for(int j = 0;j<3;j++){

                matrix[i][j] = sc.**nextInt**();

            }

        }

        System.out.**println**("\nTranspose of the matrix:");

        for(int i=0;i<3;i++){

            for(int j = 0;j<3;j++){

                System.out.**print**(" "+matrix[j][i]);

            }

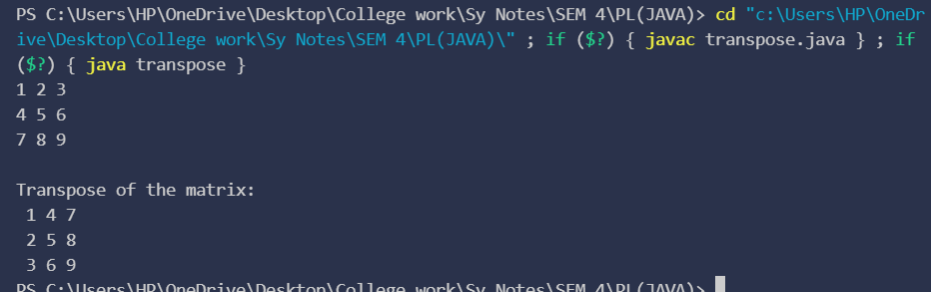
            System.out.**print**("\n");

        }

    }

}

Output:



1. Write a program to implement different types of constructors.

Program:

class Non\_Arg{

**Non\_Arg**(){

        System.out.**println**("This is a non-argument constructor!!");

    }

}

class Parametrized{

**Parametrized**(int *num*){

        System.out.**println**("This is a parameterized constructure which took the value: "+num);

    }

}

public class Constructors {

    public static void **main**(String[] *args*){

        Non\_Arg Arg = new **Non\_Arg**();

        Parametrized para = new **Parametrized**(3);

    }

}

Output: 