Government College of Engineering, Jalgaon
Department of Computer Engineering
Experiment No: 01

Name: **PRN:**
Subject:CO310U (Application programming Lab) **Sem:**V(Odd)
Class:T.Y. B.Tech **Academic Year:**2024-25
Date of Performance: **Date of Completion:**

Aim:
A.Write a JAVA program to display default values of all primitive data types of JAVA.
B. Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Take as input the speed of each racer and print back the speed of qualifying racers.

Required Software: OpenJDK version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-8u131-b11-2ubuntu1.16.04.3-b11)
OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)

Java Compiler Version - JAVAC 1.8.0_131

Theory:

Introduction to Java- Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

Java is:

Object Oriented: In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
Platform independent: Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machines, rather into platform independent bytecode. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.
Simple: Java is designed to be easy to learn. If you understand the basic concept of OOP Java would be easy to master.
Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
Architectural-neutral: Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors, with the presence of Java runtime system.
Portable: Being architectural-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary which is a POSIX subset.
Robust: Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded: With Java's multithreaded feature it is possible to write programs that can do many tasks simultaneously. This design feature allows developers to construct smoothly running interactive applications.
Interpreted: Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light weight process.
High Performance: With the use of Just-In-Time compilers, Java enables high performance.
Distributed: Java is designed for the distributed environment of the internet.
Dynamic: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.
Tools you will need:
For developing a java program, you will need a Pentium 200-MHz computer with a minimum of 64 MB of RAM (128 MB of RAM recommended).

You also will need the following software-

1) Ubuntu 14.04 (or any higher version) operating system.

2) Java JDK 8

3) Pico or any other open source text editor

When we consider a Java program it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods and instance variables mean.

Object - Objects have states and behaviours. Example: A dog has states - colour, name, breed as well as behaviours -wagging, barking, eating. An object is an instance of a class.
Class - A class can be defined as a template/ blueprint that describes the behaviours/states that object of its type support.
Methods - A method is basically a behaviour. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
Instance Variables - Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.
Basic Syntax:
About Java programs, it is very important to keep in mind the following points.
Case Sensitivity - Java is case sensitive, which means identifier Hello and hello would have different meanings in Java.
Class Names - For all class names the first letter should be in Upper Case.
If several words are used to form a name of the class, each inner world's first letter should be in UpperCase.

Example- class MyFirstJavaClass

Method Names - All method names should start with a lower-case letter. If several words are used to form the name of the method, then each inner world's first letter should be in Upper Case.

Example- public void myMethodName()

Program File Name - Name of the program file should exactly match the class name. When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java'

to the end of the name (if the file name and the class name do not match your program will not compile).
Example- Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as 'MyFirstJavaProgram.java'
public static void main(String args[]) - Java program processing starts from the main() method which is a mandatory part of every Java program.
Let us look at a simple code that would print the words Hello World.

```
public class MyFirstJavaProgram {

/* This is my first java program.
* This will print 'Hello World' as the output
*/
public static void main(String []args) {
        System.out.println("Hello World"); // prints Hello World
        }
}
```

Let's look at how to save the file, compile and run the program. Please follow the steps given below:

1) Open text editor and add the code as above.

2) Save the file as: MyFirstJavaProgram.java.

3) Open the terminal and go to the directory where you saved the class.
Assume it's student@gcoej-ThinkCentre-M70z:~$.

4) Type ' javac MyFirstJavaProgram.java' and press enter to compile your code. If there are no errors in your code, the command prompt will take you to the next line (Assumption: The path variable is set).

5) Now, type ' java MyFirstJavaProgram ' to run your program.

6) You will be able to see ' Hello World ' printed on the window.

```
student@gcoej-ThinkCentre-M70z:~$ javac MyFirstJavaProgram.java
student@gcoej-ThinkCentre-M70z:~$ java MyFirstJavaProgram

Hello World
```

Conclusion:

Name & Sign of Course Teacher

Program:

A.

```
import java.lang.*;

class p1 {
   static byte b;
   static short s;
   static int i;
   static long l;
   static float f;
   static double d;
   static char c;
   static boolean bl;

   public static void main(String[] args) {
      System.out.println("The default values of primitive data types are:");
      System.out.println("Byte :"+b);
      System.out.println("Short :"+s);
      System.out.println("Int :"+i);
      System.out.println("Long :"+l);
      System.out.println("Float :"+f);
      System.out.println("Double :"+d);

      // For char, show its default Unicode value
      System.out.println("Char :"+c + " (Unicode: \\u" + String.format("%04x", (int)c) + ")");

      System.out.println("Boolean :"+bl);
   }
}
```

Output:

```
koliv@J4RVIS MINGW64 /d/Codes/APL
$ javac p1.java

koliv@J4RVIS MINGW64 /d/Codes/APL
$ java p1.java
The default values of primitive data types are:
Byte :0
Short :0
Int :0
Long :0
Float :0.0
Double :0.0
Char : (Unicode: \u0000)
Boolean :false
```

B.

```
import java.util.*;

class p1{
   public static void main(String[] args) {
      float s1, s2, s3, s4, s5, average;
      Scanner s = new Scanner(System.in);

      // Input for the speeds of the racers
      System.out.println("Enter speed of first racer:");
      s1 = s.nextFloat();

      System.out.println("Enter speed of second racer:");
      s2 = s.nextFloat();

      System.out.println("Enter speed of third racer:");
      s3 = s.nextFloat();

      System.out.println("Enter speed of fourth racer:");
      s4 = s.nextFloat();

      System.out.println("Enter speed of fifth racer:");
      s5 = s.nextFloat();

      // Calculate the average speed
      average = (s1 + s2 + s3 + s4 + s5) / 5;

      // Determine which racers qualify
      if (s1 > average) {
         System.out.println("First racer is qualify racer:");
      } else if (s2 > average) {
         System.out.println("Second racer is qualify racer:");
      } else if (s3 > average) {
         System.out.println("Third racer is qualify racer:");
      } else if (s4 > average) {
         System.out.println("Fourth racer is qualify racer:");
      } else if (s5 > average) {
         System.out.println("Fifth racer is qualify racer:");
      }
   }
}
```

Output:

```
koliv@J4RVIS MINGW64 /d/Codes/APL
$ javac p1.java

koliv@J4RVIS MINGW64 /d/Codes/APL
$ java p1.java
Enter speed of first racer:
200
Enter speed of second racer:
150
Enter speed of third racer:
140
Enter speed of fourth racer:
120
Enter speed of fifth racer:
110
First racer is qualify racer:
```