OBJECT ORIENTED PROGRAMMING LABORATORY

21SC1204-R

STUDENT ID: 2100040024 ACADEMIC YEAR: 2023-24

STUDENT NAME:M.H.G.Subhang

Table of Contents

A.Y. 2023-24 LAB/SKILL CONTINUOUS EVALUATION

S.No	Date	Experiment Name	Pre-Lab (10M)	In-Lab (25M)			Post-Lab (10M)	Viva Voce (5M)	Total (50M)	Faculty Signature
				Program/ Procedure (5M)	Data and Results (10M)	Analysis & Inference (10M)				
1.		Introductory Session	-NA-							
2.		Basic Control Structure								
3.		Loop Statements								
4.		Programs On Recursion								
5.		Programs On Arrays								
6.		Constructors								
7.		Method Overloading								
8.		String Buffer Classes								
9.		Inheritance								
10.		Packages								
11.		Exception Handling								
12		Multi-Threading								
13.		Swing package-based event driven programming								

Following is a step by step guide to download and install Eclipse IDE:

Step 1) Installing Eclipse.

Step 2) Click on “Download” button.

Step 3) Click on “Download 64 bit” button.

Step 4) Click on “Download” button.

Step 4) Install Eclipse.

Step 5) Click on Run button.

Step 6) Click on “Eclipse IDE for Java Developers”

Step 7) Click on “INSTALL” button

Step 8) Click on “LAUNCH” button.

Step 9) Click on “Launch” button.

Step 10) Click on “Create a new Java project” link.

Step 11) Create a new Java Project

Write project name.

Click on “Finish button”.

Organization of the STUDENT LAB WORKBOOK

The laboratory framework includes a creative element but shifts the time-intensive aspects outside of the Two-Hour closed laboratory period. Within this structure, each laboratory includes three parts: Prelab, In-lab, and Post-lab.

a. Pre-Lab

The Prelab exercise is a homework assignment that links the lecture with the laboratory period - typically takes 2 hours to complete. The goal is to synthesize the information they learn in lecture with material from their textbook to produce a working piece of software. Prelab Students attending a two-hour closed laboratory are expected to make a good- faith effort to complete the Prelab exercise before coming to the lab. Their work need not be perfect, but their effort must be real (roughly 80 percent correct).

b. In-Lab

The In-lab section takes place during the actual laboratory period. The First hour of the laboratory period can be used to resolve any problems the students might have experienced in completing the Prelab exercises. The intent is to give constructive feedbackso that students leave the lab with working Prelab software - a significant accomplishmenton their part. During the second hour, students complete the In-lab exercise to reinforce the concepts learned in the Prelab. Students leave the lab having received feedback on their Prelab and In-lab work.

c. Post-Lab

The last phase of each laboratory is a homework assignment that is done following the laboratory period. In the Post-lab, students analyze the efficiency or utility of a given system call. Each Post-lab exercise should take roughly 120 minutes to complete.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE F2ILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment Title: Basic Control Structure

Aim/Objective: The aim/objective of basic control structures is to enable the execution of specific sequences of instructions based on certain conditions or criteria.

Description:

The student will understand the concept of Basic control structures are fundamental building blocks in programming that allow developers to control the flow of execution in their programs

Pre-Requisites:

1. **Java Development Kit (JDK)**
2. **Text Editor**
3. **Integrated Development Environment (IDE)**

Pre-Lab:

1) WHAT ARE JAVA TOKENS?

Ans:

Java tokens are the basic building blocks of a Java program. They are the smallest unit of a program, and they include keywords, identifiers, operators, literals, separators, and comments.

2) Explain Data Types?

Ans:

Primitive data types - includes byte , short , int , long , float , double , boolean and char. Non-primitive data types - such as String , Arrays and Classes.

3) What Are Different Operators?

Ans:

Arithmetic Operators , Assignment Operators , Logical Operators, Relational Operators, Unary Operators, Bitwise Operators, Ternary Operators, & Shift Operators.

4)what are the advantages of oop?

Ans:

Benefits of OOP language allows to break the program into the bit-sized problems that can be solved easily.

Troubleshooting is easier with the OOP language, Code Reusability, Productivity, Data Redundancy, Code Flexibility, Solving problems, & Security.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE F2ILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

In-Lab:

1) To solve the Basic Control Structure

. Java program to read data from user – data types

.Write a Java Program to demonstrate different types of arithmetic operators support by java. Illustrate using an example, the use of arithmetic operators.

.Procedure/Program:

```
1. import java.util.Scanner;


public class ReadDataFromUser {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Reading data for different data types
        System.out.print("Enter an integer: ");
        int numInt = scanner.nextInt();

        System.out.print("Enter a long integer: ");
        long numLong = scanner.nextLong();

        System.out.print("Enter a floating-point number: ");
        float numFloat = scanner.nextFloat();

        System.out.print("Enter a double-precision number: ");
        double numDouble = scanner.nextDouble();

        scanner.nextLine(); // Consume the newline left by previous nextXxx()
methods
        System.out.print("Enter a character: ");
        char charValue = scanner.nextLine().charAt(0); // Read a single
character
        System.out.print("Enter a string: ");
        String strValue = scanner.nextLine();

        // Displaying the user input
        System.out.println("Integer: " + numInt);
        System.out.println("Long: " + numLong);
        System.out.println("Floating-point: " + numFloat);
        System.out.println("Double: " + numDouble);
        System.out.println("Character: " + charValue);
        System.out.println("String: " + strValue);
        scanner.close();
    }
}
```

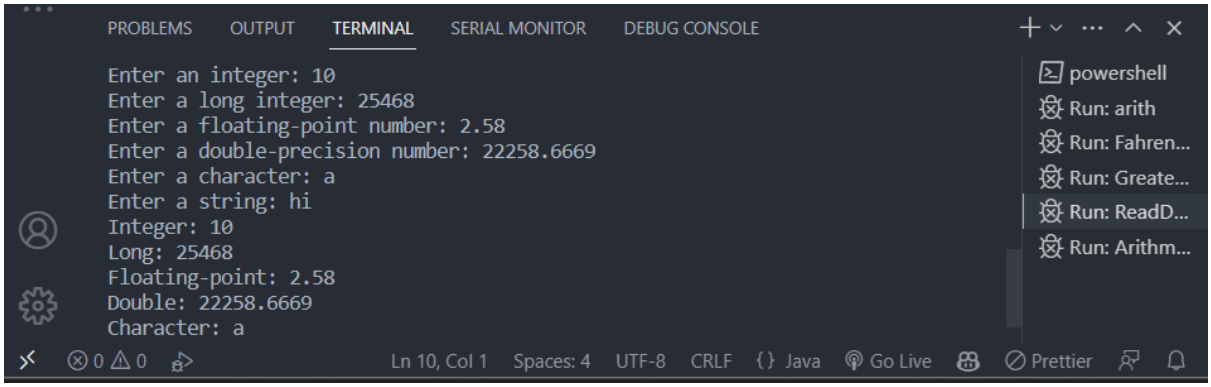
2.
```java
import java.util.Scanner;

public class ArithmeticOperatorsDemo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Reading data from the user
        System.out.print("Enter an integer number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter another integer number: ");
        int num2 = scanner.nextInt();

        System.out.print("Enter a floating-point number: ");
        double num3 = scanner.nextDouble();

        System.out.print("Enter another floating-point number: ");
        double num4 = scanner.nextDouble();

        // Demonstrating arithmetic operators
        int sum = num1 + num2;
        int difference = num1 - num2;
        int product = num1 * num2;
        int quotient = num1 / num2;
        int remainder = num1 % num2;

        double sumFloat = num3 + num4;
        double differenceFloat = num3 - num4;
        double productFloat = num3 * num4;
        double quotientFloat = num3 / num4;

        // Displaying the results
        System.out.println("Sum: " + sum);
        System.out.println("Difference: " + difference);
        System.out.println("Product: " + product);
        System.out.println("Quotient: " + quotient);
        System.out.println("Remainder: " + remainder);
        System.out.println("Floating-point Sum: " + sumFloat);
        System.out.println("Floating-point Difference: " + differenceFloat);
        System.out.println("Floating-point Product: " + productFloat);
        System.out.println("Floating-point Quotient: " + quotientFloat);
        scanner.close();
    }
}
```

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE F2ILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

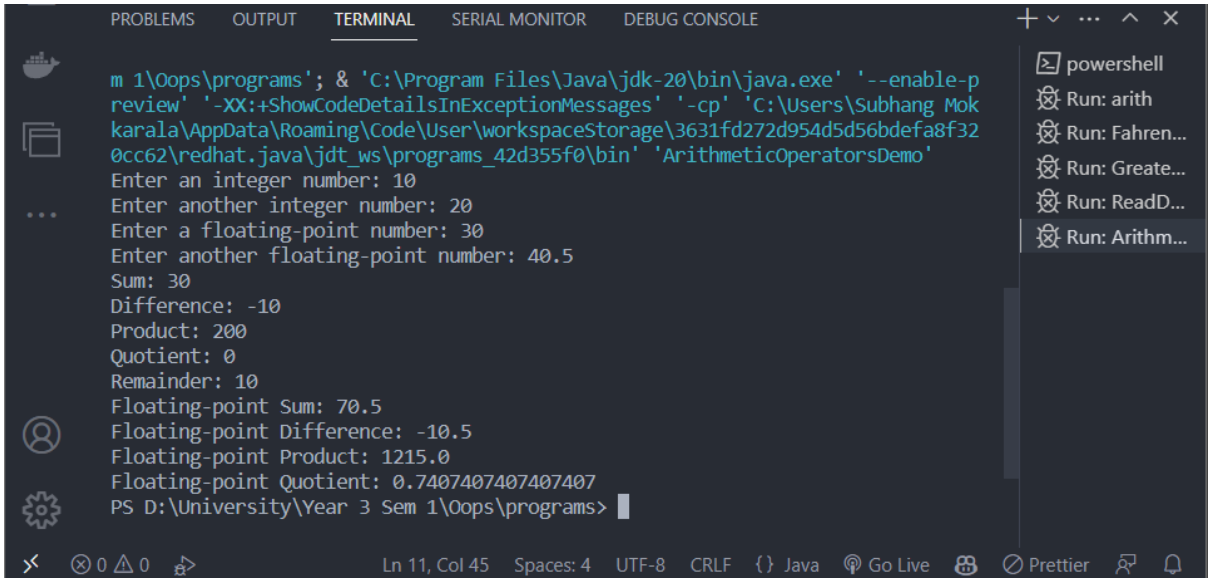
- **Data and Results:**

19-07-2023

1.



2.



- **Analysis and Inferences:**

Sample VIVA-VOCE Questions (In-Lab):

1.What is oop ?

Object-oriented programming (OOP) is a style of programming characterized by the identification of classes of objects closely linked with the methods (functions) with which they are associated.

2.What are the differences between C++ and Java?

C++ code can be called into C, C++ libraries, or API of operating systems. On the other hand, Java code is only ideal for Java-based libraries. In addition, C++ interacts with hardware more effectively than Java due to its low-level nature and lack of automatic memory management.

3.Will the program run if we write static public void main?

If you write static public void instead of public static void then it is perfectly OK. Your Java program will compile and run successfully. It doesn't really make any difference as long as method name comes last and return type of method comes second last

4.Can Java be said to be the complete object-oriented programming language

Java is not a fully object-oriented language as it supports primitive data types like int, byte, long, short, etc., which are not objects. Hence these data types like int, float, double, etc., are not object-oriented. That's why Java is not 100% object-oriented.

5.Can you implement pointers in a Java Program?

Pointers are a feature associated with the C programming language. They are responsible for providing the memory address in a situation where the programmer directly stores the data. But unfortunately, you won't find any pointers in Java. The closest thing to pointers are references in Java.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE F2ILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Post-Lab:

. Java Program to find greatest of three number

. Java program to convert Fahrenheit To Celsius degrees and display messages depends on the temperature

- **Procedure/Program:**

```
1. public class GreatestOfThreeNumbers {

    public static void main(String[] args) {

        java.util.Scanner scanner = new java.util.Scanner(System.in);

        System.out.println("Enter three numbers:");
        int num1 = scanner.nextInt();
        int num2 = scanner.nextInt();
        int num3 = scanner.nextInt();

        int greatest = num1;

        if (num2 > greatest) {
            greatest = num2;
        }

        if (num3 > greatest) {
            greatest = num3;
        }

        System.out.println("The greatest number among " + num1 + ", " + num2 +
", and " + num3 + " is: " + greatest);

        scanner.close();
    }
}
```

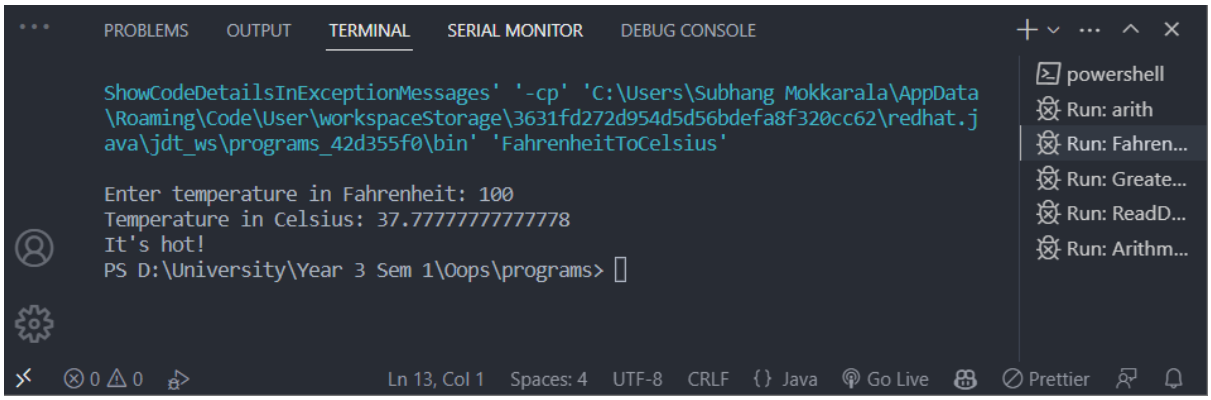
2.
```java
public class FahrenheitToCelsius {

    public static void main(String[] args) {

        java.util.Scanner scanner = new java.util.Scanner(System.in);

        System.out.print("Enter temperature in Fahrenheit: ");
        double fahrenheit = scanner.nextDouble();

        // Convert Fahrenheit to Celsius
        double celsius = (fahrenheit - 32) * 5 / 9;

        System.out.println("Temperature in Celsius: " + celsius);

        // Display messages based on the temperature
        if (celsius < 0) {
            System.out.println("It's freezing cold!");
        } else if (celsius >= 0 && celsius < 20) {
            System.out.println("It's cool.");
        } else if (celsius >= 20 && celsius < 30) {
            System.out.println("It's warm.");
        } else {
            System.out.println("It's hot!");
        }

        scanner.close();
    }
}
```

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE F2ILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

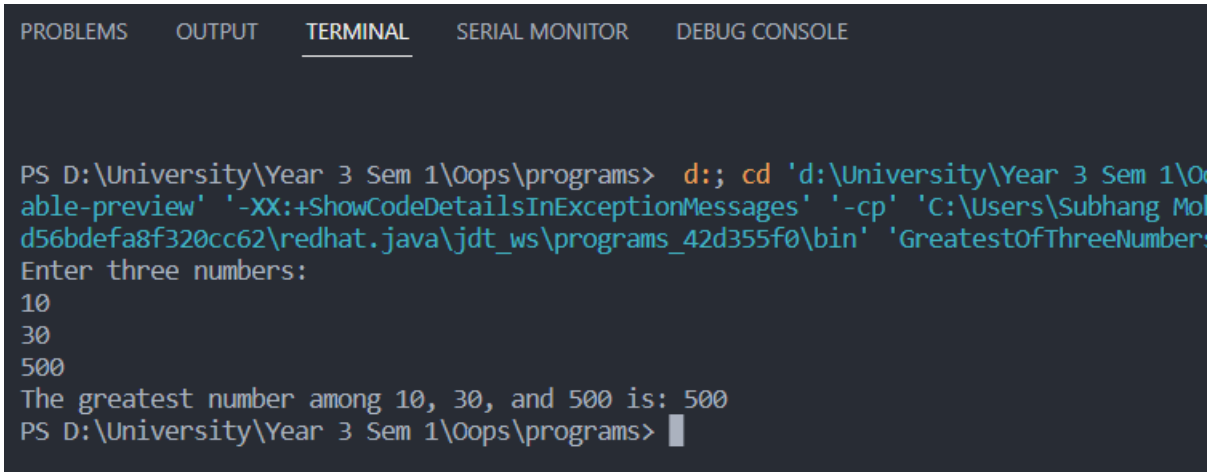
- **Data and Results:**

19-07-2023

1.



2.



- **Analysis and Inferences:**
  - The first Java program is designed to find the greatest of three numbers provided by the user. It uses an if-else statement to compare the numbers and determine the maximum. By defining a separate method, the program encapsulates the comparison logic, making it modular and easy to understand.
  -
  - The second Java program converts a temperature given in Fahrenheit to Celsius using a simple formula. It then displays the converted Celsius temperature and provides different messages based on the temperature range. This program showcases the use of basic arithmetic operations and conditional statements to handle temperature conversion and temperature-based messaging.
  -

- Both programs demonstrate fundamental concepts in Java programming, including user input handling, conditional statements, and method definition. They are simple yet effective examples of how Java can be used to perform basic calculations and decision-making tasks.

Evaluator Remark (if Any):	**Marks Secured: _____out of 50**
	Signature of the Evaluator with Date

Evaluator MUST ask Viva-voce prior to signing and posting marks for each experiment.