Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	**2**
Student Name:	**Deep Patel**
Roll No :	**8**

Title:

To Add Two Numbers, Print Number Entered by User, Swap Two Numbers, Check Whether Number is Even or Odd

1.2 Implement using Java

Learning Objective:

- Students will be able to write java program for simple arithmetic operations and take input from user.

Learning Outcome:

- Ability to execute a simple Java program with and without any inputs to the program.
- Understanding the constructs in Java.

Course Outcome:

ECL304.1	Understand object-oriented programming concepts and implement using Java

Theory:

(1) Difference between procedural and object oriented language:

Object-oriented programming and procedural programming both are used to develop the applications. Both of them are high-level programming languages; and it is also important to know the difference between them.

- Procedural Language:

It is defined as a programming language derived from the structure programming and based on calling procedures. The procedures are the functions, routines, or subroutines that consist of the computational steps required to be carried. As compared to object-oriented programming, procedural programming is less secure.

- Object Oriented Programming language:

In Object oriented programming, program is divided into small parts called objects. Object oriented programming follows bottom-up approach. Object oriented programming have access specifiers like private, public, protected etc. Object oriented programming provides data hiding so it is more secure.

(2) Application of object orientation:

1. Security: In OOP, Data is encapsulated with methods in the class so that data is protected and secured from accidental modification by other external nonmember methods.
2. Reusability: Through inheritance, we can use the features of an existing class in a new class without repeating existing code that saves a lot of time for developers, and also increases productivity.
3. Effective communication: In OOP, objects can communicate via message passing technique that makes interface descriptions with outside systems much simpler.
4. Developing complex software: OOPs is the most suitable approach for developing complex software because it minimizes the complexity through the feature of inheritance.
5. Easily upgraded: Object-oriented system can be easily upgraded from small to large systems because OOP uses bottom-up approach.
6. Easy partition of work: It is easy to partition complicated work in a project based on objects.
7. Maintenance: The maintenance of object-oriented code is easier.

(3) Brief introduction to C++ and Java:

1. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible.

2. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to c/cpp .

3. The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. James Gosling in 1995 developed

Java, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

4. JAVA was developed by Sun Microsystems Inc in 1991, later acquired by Oracle Corporation. It was developed by James Gosling and Patrick Naughton. It is a simple programming language. Writing, compiling and debugging a program is easy in java. It helps to create modular programs and reusable code.

Program:

```
import java.util.Scanner;


public class extc {
    public static void main(String[] args) {


      Scanner sc = new Scanner(System.in);  // Create a Scanner object
     /* System.out.println("Enter username");
      String userName = sc.nextLine();  // Read user input
      System.out.println("Username is: " + userName);  // Output user input
    */
      int n1,n2,temp;
      System.out.println("Enter first number");
      n1=sc.nextInt();
      System.out.println("Enter second number");
      n2=sc.nextInt();
      System.out.println("Number 1 = "+n1+" Number 2 = "+n2);
      System.out.println("\n ADDITION\n");
      System.out.println("\nAddition of both numbers is: " +(n1+n2));
      System.out.println("\n SWAPPING\n");
      temp=n1;
      n1=n2;
```

```
        n2=temp;
        System.out.println("After swapping Number 1 = "+n1+" Number 2 = "+n2);


        System.out.println("\n EVEN/ODD\n");
        if(n1%2==0)
            System.out.println(n1+" is Even");
        else
            System.out.println(n1+" is Odd");
      }
}
```

Input given: 6 and 7

Output:

```
Enter first number
6
Enter second number
7
Number 1 = 6 Number 2 = 7

 ADDITION


Addition of both numbers is: 13

 SWAPPING

After swapping Number 1 = 7 Number 2 = 6

 EVEN/ODD

7 is Odd
```