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:	**1**
Student Name:	**RAVEENA RUPESH PITALE**
Roll No :	**27**

Title:

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

1.1 Implement using C++

1.2 Implement using Java

Learning Objective:

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

Learning Outcome:

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

Course Outcome:

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

Faculty: Ms. Deepali Kayande

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

Theory:

Q.1) Difference between procedural and object oriented language

PROCEDURAL ORIENTED PROGRAMMING	OBJECT ORIENTED PROGRAMMING
In procedural programming, program is divided into small parts called functions.	In object oriented programming, program is divided into small parts called objects.
Procedural programming follows top down approach.	Object oriented programming follows bottom up approach.
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way for hiding data so it is *less secure*.	Object oriented programming provides data hiding so it is *more secure*.
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on unreal world.	Object oriented programming is based on real world.
Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

Q.2) Application of object orientation

1. Client-Server Systems

Object-oriented client-server systems provide the IT infrastructure, creating Object-Oriented Client-Server Internet (OCSI) applications. Here, infrastructure refers to operating systems, networks, and hardware. OSCI consist of three major technologies:

- The Client Server
- Object-Oriented Programming
- The Internet

2. Object-Oriented Databases

- These databases try to maintain a direct correspondence between the real-world and database objects in order to let the object retain its identity and integrity. They can then be identified and operated upon.

3. Real-Time System Design

- Real-time systems inherent complexities that make it difficult to build them. Object-oriented techniques make it easier to handle those complexities. These techniques present ways of dealing with these complexities by providing an integrated framework, which includes schedulability analysis and behavioral specifications.

4. Simulation and Modeling System

- It's difficult to model complex systems due to the varying specification of variables. These are prevalent in medicine and in other areas of natural science, such as ecology, zoology, and agronomic systems. Simulating complex systems requires modeling and understanding interactions explicitly. Object-oriented programming provides an alternative approach for simplifying these complex modeling systems.

5. Hypertext and Hypermedia

- OOP also helps in laying out a framework for hypertext. Basically, hypertext is similar to regular text, as it can be stored, searched, and

edited easily. The only difference is that hypertext is text with pointers to other text as well.

- Hypermedia, on the other hand, is a superset of hypertext. Documents having hypermedia not only contain links to other pieces of text and information but also to numerous other forms of media, ranging from images to sound.

Q3) Brief introduction to C++ and Java

1) C++

C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a compiled language. C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the features & key-points to note about the programming language are as follows:

- **Simple**: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
- **Mid-level language**: It is a mid-level language as we can do both systemsprogramming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
- **Object-Oriented**: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
- **Compiled Language**: C++ is a compiled language, contributing to its speed.

2) JAVA

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. This tutorial gives a

complete understanding of Java. This reference will take you through simple and practical approaches while learning Java Programming language.

Java is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

- **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 machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamperfree systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** − Being architecture-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.

ALGORITHM:

STEP 1: Start
STEP 2: Take input N1 and N2 from user
STEP 3: Addition = N1+N2
STEP 4: Declare temporary variable with name ‘temp’
STEP 5: Temp=N1, N1=N2, N2=Temp
STEP 6: Check N1 divisible by 2, if yes number is even else odd
STEP 7: Print addition of N1 and N2
STEP 8: Print swapped numbers
STEP 9: Stop

PROGRAM:

```
import java.util.Scanner;
public class Lab1 {
        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");

         }
}
```

OUTPUT:

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent
Enter first number
13
Enter second number
12
Number 1 = 13 Number 2 = 12

 ADDITION


Addition of both numbers is: 25

 SWAPPING

After swapping Number 1 = 12 Number 2 = 13

 EVEN/ODD

12 is Even

Process finished with exit code 0
```