

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:	1.2
Student Name:	Bhagyesh Manjrekar
Roll No:	07

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:

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

Theory:

(1) Difference between procedural language 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.

○ **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.

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

Object oriented programming is based on real world. Overloading is possible in Object oriented programming. Object oriented programming is based on real world. The examples of Object-oriented programming (Oopm) programming are Java, Python, C# etc.

○ **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. Procedural programming follows a top-down approach during the designing of a program.

It gives importance to the concept of the function and divides the large programs into smaller parts or called as functions. Procedural programming is straightforward. Unlike object-oriented programming, there are no access modifiers introduced in procedural programming. The examples of procedural programming are BASIC, Pascal and C etc.

(2) Application of object orientation in JAVA

=> OOPs concept in Java offers several advantages that are not available in procedural programming like C, Pascal, etc. Some of the major benefits of object-oriented programming in java are as follows:

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 non-member 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.

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

- 7. Maintenance: The maintenance of object-oriented code is easier.
- 8. Efficiency: The concepts of OOP provide better efficiency and an easy development process.

The concepts of OOPs provide many benefits for the programmer to design an efficient program. Due to its reusability feature, it is widely used in many areas. Some of the application areas of OOP are as follows:

- a. Real-time systems
- b. Object-oriented database
- c. Graphical user interface design in the Windows operating system.
- d. Artificial intelligence and expert systems
- e. Parallel programming
- f. CAD/CAM software and in many areas.

(3) Brief introduction to 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.

5. JAVA Terminology as:

- Java Virtual Machine (JVM)

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

- Bytecode
- Java Development Kit (JDK)
- Java Runtime Environment (JRE)

PROGRAM 1: Program to print number Entered by user

Algorithm:

Step 1: START

Step 2: Initialize variables n1, n2 for storing the numbers entered by User.

Step 3: Store numbers entered by user.

Step 4: Display the entered numbers.

Step 5: STOP

Program:

```
// print numbers entered by user
import java.util.Scanner;

public class printnumbers
{

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n1,n2;

        System.out.println("Enter first number");
        n1=sc.nextInt();
```

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

```
System.out.println("Enter second number");

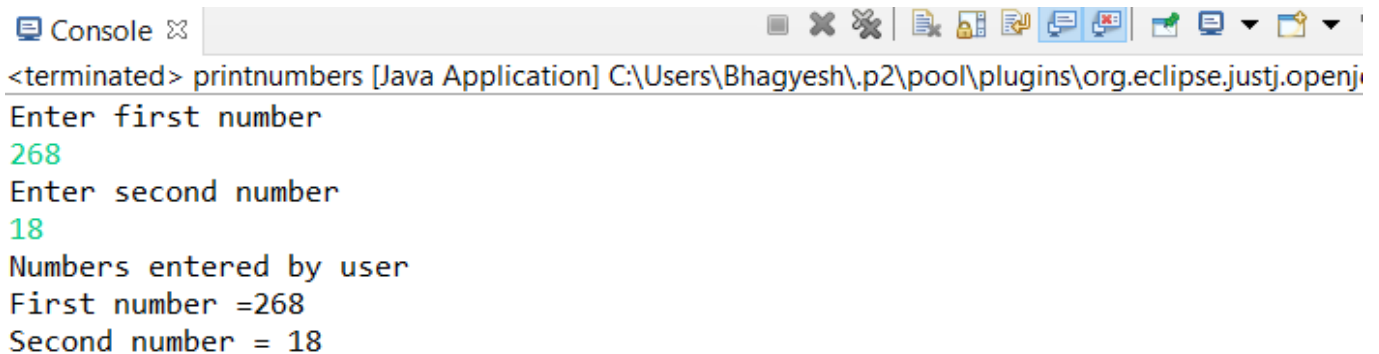
n2=sc.nextInt();

System.out.println("Numbers entered by user" + "\n" + "First number ="
+n1+ "\n" +"Second number = " +n2);

}

}
```

OUTPUT:



```
<terminated> printnumbers [Java Application] C:\Users\Bhagyesh\.p2\pool\plugins\org.eclipse.justj.openj
Enter first number
268
Enter second number
18
Numbers entered by user
First number =268
Second number = 18
```

PROGRAM 2: To add Two numbers entered by user

Algorithm:

Step 1: START

Step 2: Initialize variables **n1**, **n2**, **add** for storing the numbers entered by user and storing the addition value of **n1** and **n2** in **add** variable.

Step 3: Store numbers entered by user in n1, n2.

Step 4: Display the addition of numbers entered by user.

Step 5: STOP

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

PROGRAM:

```
// Add 2 numbers entered by user

import java.util.Scanner;

public class add_2_numbers
{

    public static void main(String[] args)
    {

        try (Scanner sc = new Scanner(System.in))
        {

            int n1,n2;

            System.out.println("Enter first number");
            n1=sc.nextInt();

            System.out.println("Enter second number");
            n2=sc.nextInt();

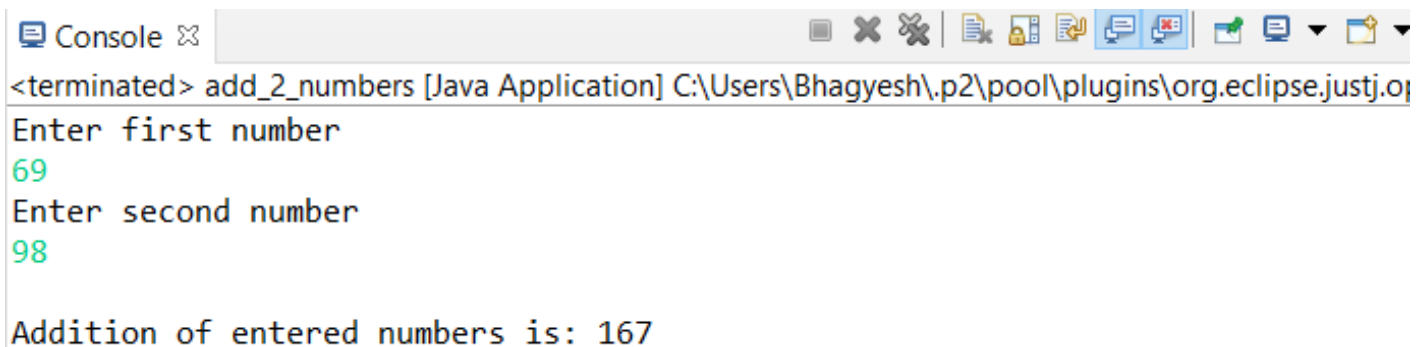
            System.out.println("\nAddition of entered numbers is: " +(n1+n2));

        }

    }

}
```

OUTPUT:



The screenshot shows the Eclipse IDE's Console window. The title bar reads "Console" with a close icon. The text in the console is as follows: "<terminated> add_2_numbers [Java Application] C:\Users\Bhagyesh\.p2\pool\plugins\org.eclipse.justj.oj". Below this, the program's output is displayed: "Enter first number" followed by the input "69" in green; "Enter second number" followed by the input "98" in green; and finally, "Addition of entered numbers is: 167".

PROGRAM 3: To swap values of two variables with each other

Algorithm:

Step 1: START

Step 2: Initialize variables **n1, n2, temp** for storing the numbers entered by user.

Step 3: Store numbers entered by user in n1, n2 and replacing the value of n1 in temp and n2 in n1 and then replacing the n2 value by temp value.

Step 4: Display the numbers before swapping.

Step 5: Display the numbers after swapping.

Step 6: STOP

PROGRAM:

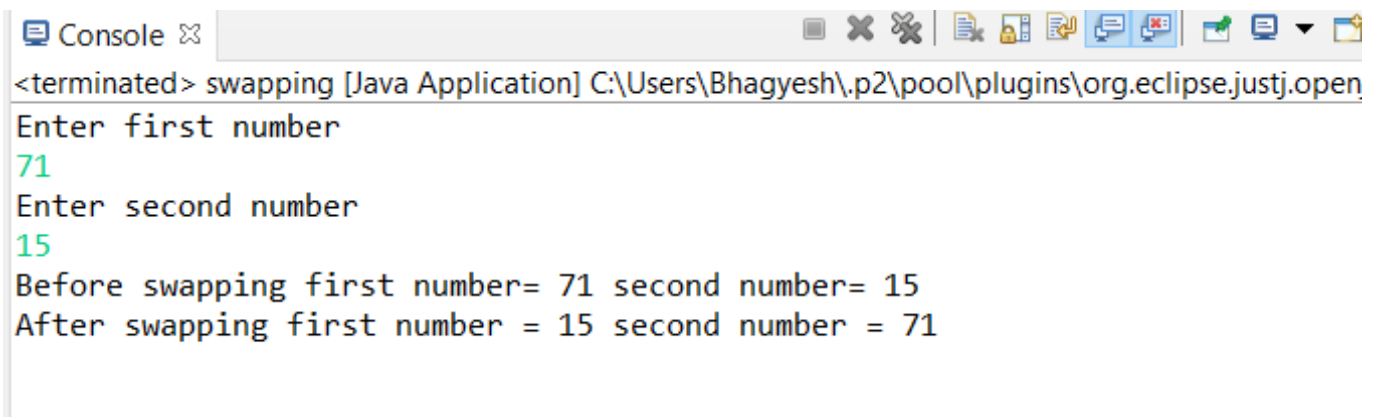
```
// Swapping two numbers
import java.util.Scanner;
public class swapping
{

    public static void main(String[] args)
    {
        try (Scanner sc = new Scanner(System.in))
        {
            int n1,n2,temp;
            System.out.println("Enter first number");
            n1=sc.nextInt();
            System.out.println("Enter second number");
            n2=sc.nextInt();
```

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

```
        System.out.println("Before swapping first number= " +n1+ " second number=
"+n2);
        temp=n1;
        n1=n2;
        n2=temp;
        System.out.println("After swapping first number = "+n1+" second number =
"+n2);
    }
}
```

OUTPUT:

A screenshot of a Java console window titled "Console". The window shows the execution of a Java application. The output text is as follows:
<terminated> swapping [Java Application] C:\Users\Bhagyesh\.p2\pool\plugins\org.eclipse.justj.open
Enter first number
71
Enter second number
15
Before swapping first number= 71 second number= 15
After swapping first number = 15 second number = 71

PROGRAM 4: To Check Whether the entered Number is Even or Odd

Algorithm:

Step1: START

Step 2: Enter the number (num).

Step 3: By using If-else condition number is divide by 2; if remainder is 0 then even number or it will be odd number.

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

Step 4: Display that the number is even or odd.

Step 5: STOP

PROGRAM:

```
// Number is even or odd
import java.util.Scanner;

public class even_or_odd
{

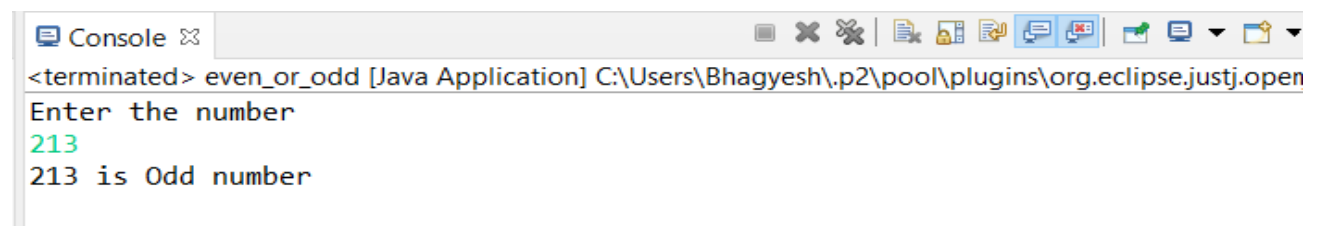
    public static void main(String[] args)
    {
        {
            int n1;

            System.out.println("Enter the number");
            n1=sc.nextInt();

            if(n1%2==0)

                System.out.println(n1+ " is Even number");
            else
                System.out.println(n1+ " is Odd number");
        }
    }
}
```

OUTPUT:

A screenshot of the Eclipse IDE's Console window. The title bar shows 'Console' with a maximize icon. The console text displays the execution of the 'even_or_odd' Java application. It starts with a prompt '<terminated> even_or_odd [Java Application] C:\Users\Bhagyesh\.p2\pool\plugins\org.eclipse.justj.open' followed by the user input 'Enter the number' and the program output '213' and '213 is Odd number'.

```
<terminated> even_or_odd [Java Application] C:\Users\Bhagyesh\.p2\pool\plugins\org.eclipse.justj.open
Enter the number
213
213 is Odd number
```