Course Name:	**Object Oriented Programming**	**Semester:**	**III**
Date of Performance:	**4 / 8 / 2025**	**Batch No:**	**EXCP A1**
Faculty Name:	**Prof. Amrita Naiksatam**	**Roll No:**	**20**
Faculty Sign & Date:		**Grade/Marks:**	**___/15**

Experiment No: 3
Title: Concept of static method and recursive function

Aim and Objective of the Experiment:

Learn the concept of static method and recursive function in Java

COs to be achieved:

CO1: Understand concepts of Object Oriented Programming and basic characteristics of Java.

Tools used:

Notepad, Command Prompt

Theory:

(About static method)

Code:

1. Write a recursive function 'GCD' to find the GCD of the given two numbers. Use this in main to find the GCD and LCM two given numbers.
   - Variation 1: Implementation with One class only
   - Variation 2: Accessibility with static and non-static methods within class and outside class.

Note : For input use *command-line-argument* instead of *scanner*

Program:

```
class MathOperations
{
    public static int gcdStatic(int a, int b)
    {
        if (b == 0)
            return a;
        return gcdStatic(b, a % b);
    }

    public int lcmNonStatic(int a, int b)
    {
        return (a * b) / gcdStatic(a, b);
    }
}

public class GCDLCMMain
{
    public static void main(String[] args)
    {
        if (args.length < 2)
        {
            System.out.println("Please enter two numbers:");
            return;
        }

        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);

        int gcdValue = MathOperations.gcdStatic(num1, num2);
        MathOperations mathOps = new MathOperations();
        int lcmValue = mathOps.lcmNonStatic(num1, num2);

        System.out.println("GCD of " + num1 + " and " + num2 + " = " + gcdValue);
        System.out.println("LCM of " + num1 + " and " + num2 + " = " + lcmValue);
    }
}
```

Output:

Post Lab Subjective/Objective type Questions:

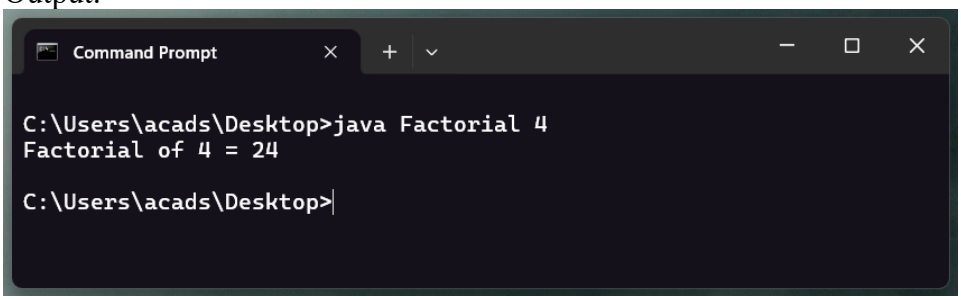
1. Write a recursive static method for calculation of factorial of n number.
   Program:

```
public class Factorial
{
    public static long factorial(int n)
    {
        if (n == 0 || n == 1)
            return 1;
        return n * factorial(n - 1);
    }

    public static void main(String[] args)
    {
        if (args.length < 1)
        {
            System.out.println("Please enter a number as command-line
arg.");
            return;
        }

        int num = Integer.parseInt(args[0]);
        long result = factorial(num);
        System.out.println("Factorial of " + num + " = " + result);
    }
}
```

Output:



2. What are command-line-argument and how are they used?

Ans:

Command-line arguments are values passed to a program when it starts, given after the program name in the terminal.

They are stored in String[] args in Java's main method and let you provide input without using Scanner.

Conclusion:

This experiment demonstrated the use of static methods and recursion in Java by implementing programs for GCD, LCM, and factorial calculation using command-line arguments. It reinforced the understanding of method accessibility (static and non-static), recursive logic, and passing inputs at runtime without interactive prompts.

Signature of faculty in-charge with Date: