Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering, Semester: Spring, Year: 2024, B.Sc. in CSE (weekend)

Lab Report # 01

Course Title: Object Oriented Programming

Course Code: CSE-202

Section: 223 E1

Student Details

Name		ID
1.	Waliullah	223015026

Date : 23-02-2024

Submission Date : 01-03-2024

Course Teacher's Name : Md. Ashiqussalehin

Assignment Status

Marks: ………………………………… **Signature:....................**

Comments:.. **Date:............................**

Title: Implement checking of odd and even numbers.
Implement summation of factorial odd number series below this series.

$$\text{Sum} = \frac{x^2}{1!} + \frac{x^4}{3!} + \frac{x^6}{5!} + \dots \; \frac{x^n}{(n-1)!}$$

1. Introduction:

This program demonstrates two functionalities in Java:

- **Checking if a number is even or odd:** This is a common task used in various applications for data validation, filtering, and manipulation.
- **Calculating the sum of the factorials of a series of odd numbers:** This can be useful in mathematical calculations or specific problem-solving scenarios.

2. Objective:

This Java program defines three functions:

- **checkOddEven(num)**: This function checks if a given number is even or odd and returns a string indicating the result.
- **factorial(num):** This function calculates the factorial of a given number.
- **factorialSumOddSeries(n and x):** This function calculates the sum of factorials of the **x** and **n** odd series.

  Example: Below this series

$$\text{Sum} = \frac{x^2}{1!} + \frac{x^4}{3!} + \frac{x^6}{5!} + \dots \; \frac{x^n}{(n-1)!}$$

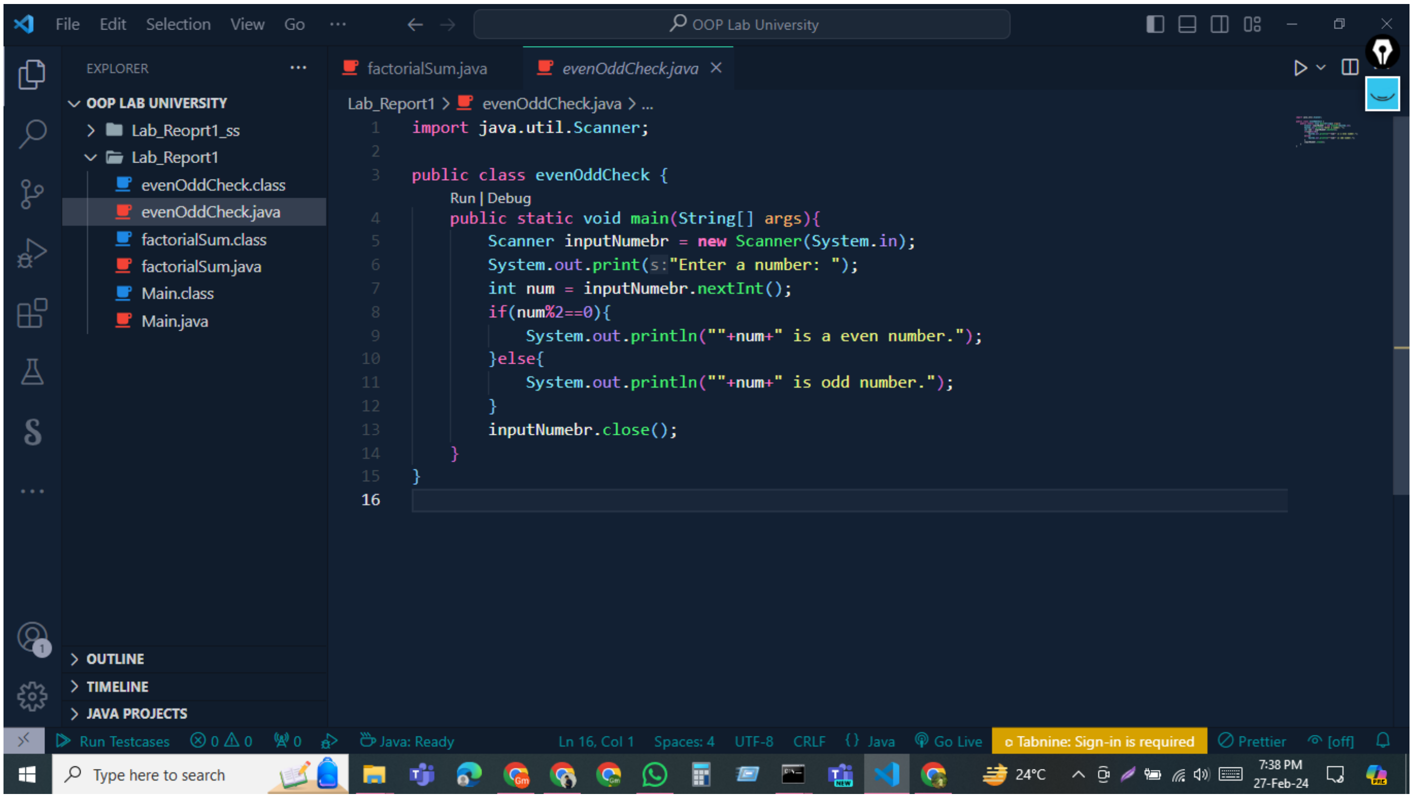
3. Use Case:

- Checking Odd/Even Numbers:
  1. Determines whether an integer is odd or even.
  2. Commonly used in various programming applications, data structures, and algorithms.
  3. Can be implemented using the modulo operator (%) in most programming languages.
- Factorial Summation of Odd Numbers:
  1. Calculates the sum of the factorials of an odd number series.
  2. Has specific use cases in mathematics and computer science, such as generating odd perfect numbers or exploring properties of odd factorials.

4. Implementations and Output:

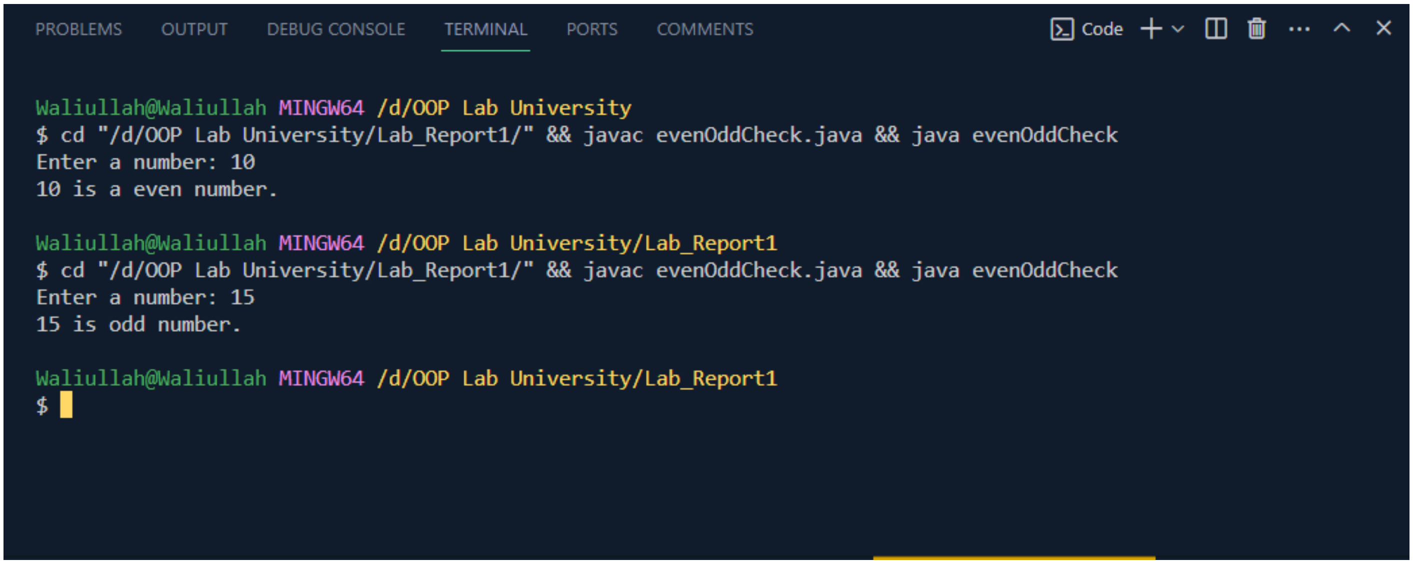
- **Problem 1:**

Code:



```java
import java.util.Scanner;

public class evenOddCheck {
    public static void main(String[] args){
        Scanner inputNumebr = new Scanner(System.in);
        System.out.print(s:"Enter a number: ");
        int num = inputNumebr.nextInt();
        if(num%2==0){
            System.out.println(""+num+" is a even number.");
        }else{
            System.out.println(""+num+" is odd number.");
        }
        inputNumebr.close();
    }
}
```

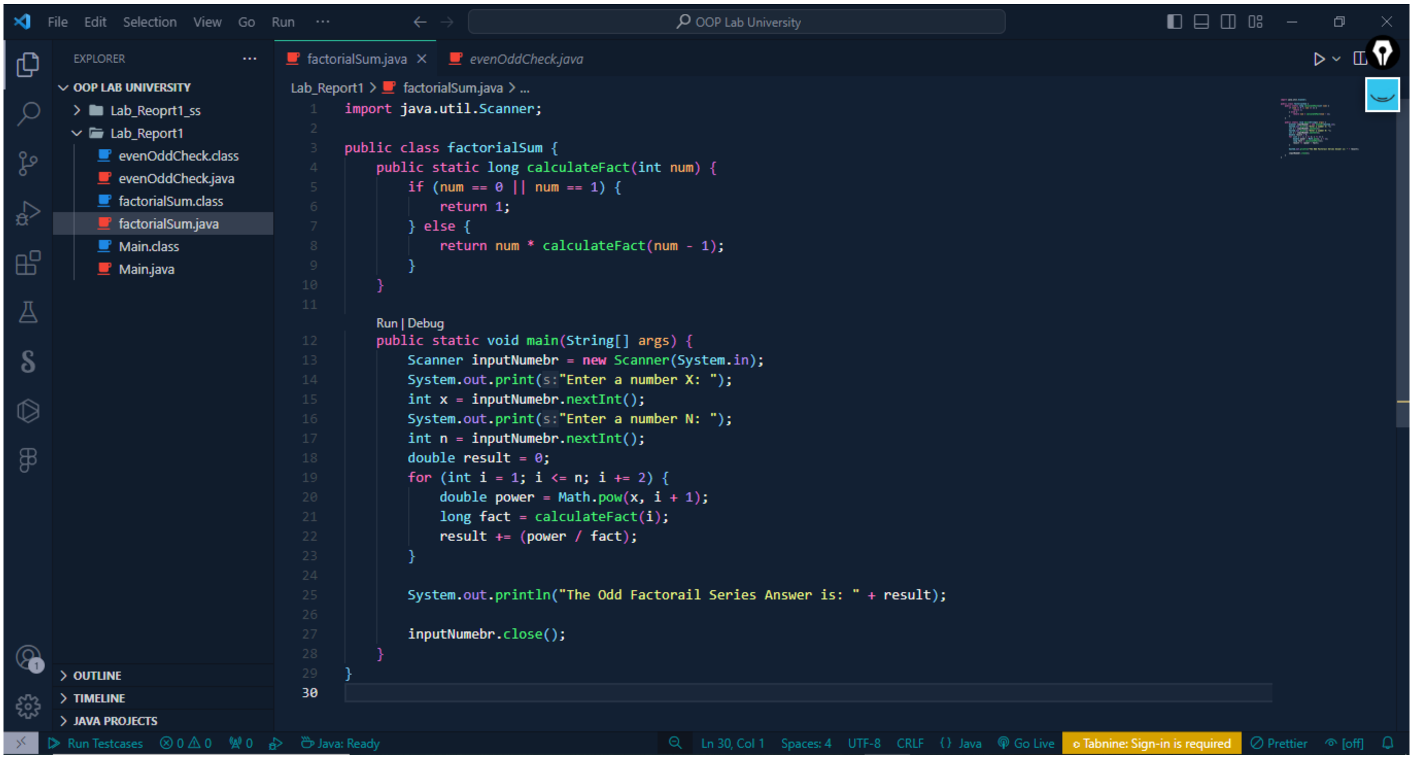
Output:



```
Waliullah@Waliullah MINGW64 /d/OOP Lab University
$ cd "/d/OOP Lab University/Lab_Report1/" && javac evenOddCheck.java && java evenOddCheck
Enter a number: 10
10 is a even number.

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report1
$ cd "/d/OOP Lab University/Lab_Report1/" && javac evenOddCheck.java && java evenOddCheck
Enter a number: 15
15 is odd number.

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report1
$ 
```

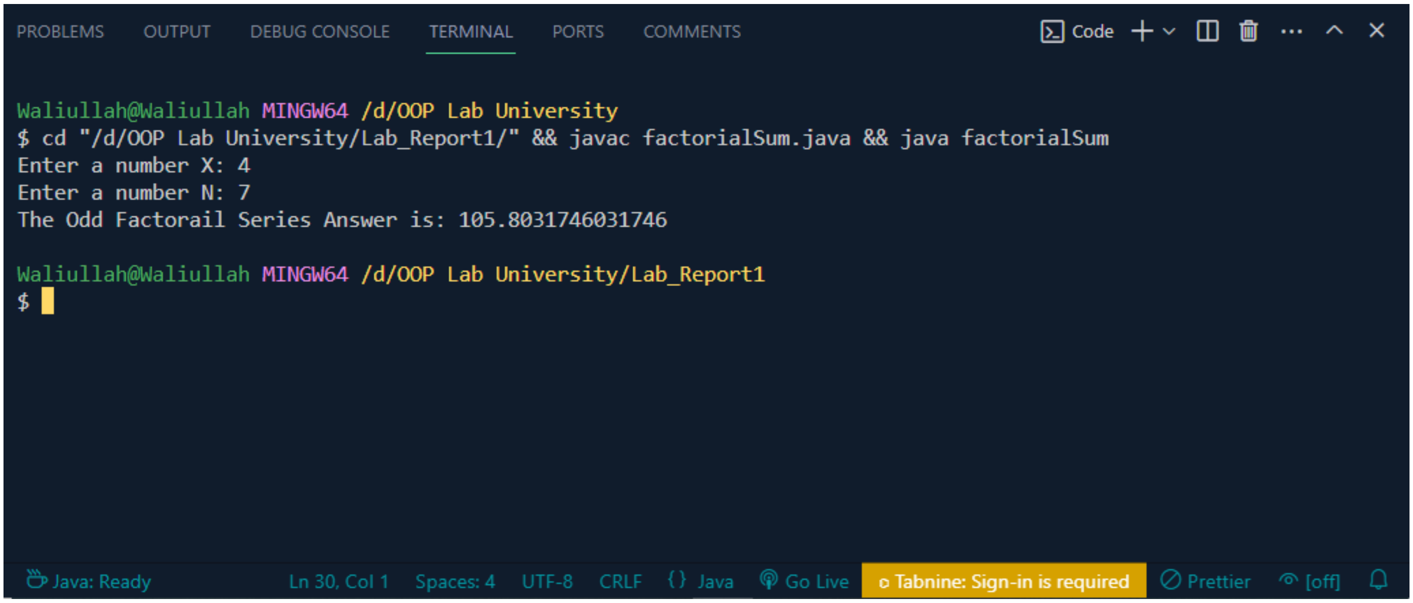
- **Problem 2:**

Code:



```java
import java.util.Scanner;

public class factorialSum {
    public static long calculateFact(int num) {
        if (num == 0 || num == 1) {
            return 1;
        } else {
            return num * calculateFact(num - 1);
        }
    }

    public static void main(String[] args) {
        Scanner inputNumebr = new Scanner(System.in);
        System.out.print(s:"Enter a number X: ");
        int x = inputNumebr.nextInt();
        System.out.print(s:"Enter a number N: ");
        int n = inputNumebr.nextInt();
        double result = 0;
        for (int i = 1; i <= n; i += 2) {
            double power = Math.pow(x, i + 1);
            long fact = calculateFact(i);
            result += (power / fact);
        }

        System.out.println("The Odd Factorail Series Answer is: " + result);

        inputNumebr.close();
    }
}
```

Output:



```
Waliullah@Waliullah MINGW64 /d/OOP Lab University
$ cd "/d/OOP Lab University/Lab_Report1/" && javac factorialSum.java && java factorialSum
Enter a number X: 4
Enter a number N: 7
The Odd Factorail Series Answer is: 105.8031746031746

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report1
$ 
```

5. Limitations:

- Integer Overflow: The factorial function can lead to integer overflow for large numbers. Consider using long or a specialized library for big integer calculations.
- Performance: Repeatedly calculating factorials within the loop can be computationally expensive for large limit values. Explore alternative approaches like pre-computing and storing factorials.
- Negative Input: The code doesn't handle negative input for checking odd/even and series calculation. Consider adding checks for valid input ranges.

6. Conclusion:

This program successfully implements functions to check odd and even numbers and calculates the sum of factorials of odd numbers within a given limit. It demonstrates basic Java control flow, looping, and mathematical operations, providing a valuable solution for these functionalities.