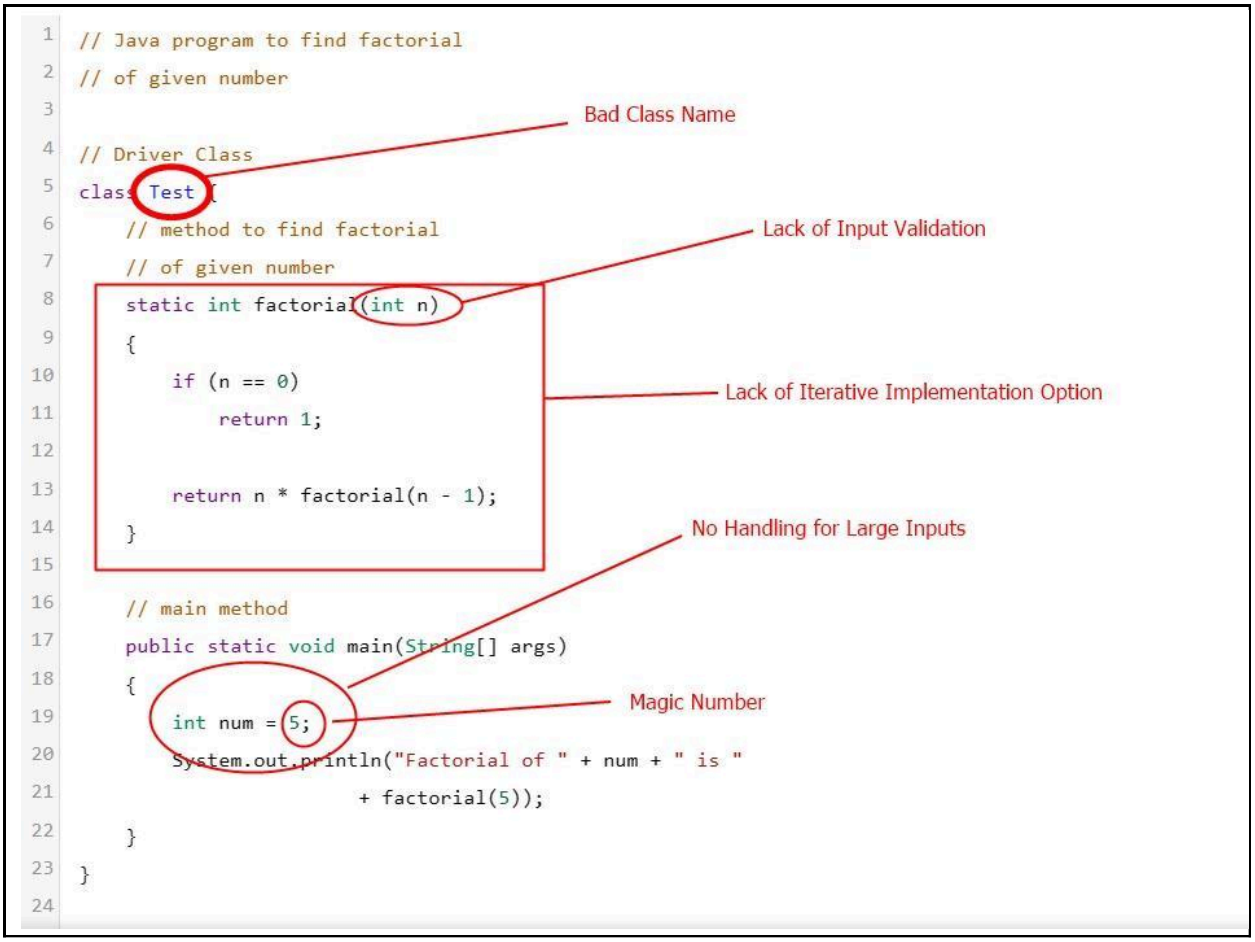
Search for source code on the internet, identify any errors, highlight them, and explain why they are incorrect. Then, rewrite the entire program to improve the source code.

This is an example

The source code originates from:
https://www.geeksforgeeks.org/java-program-for-factorial-of-a-number/



```
// Java program to find factorial
// of given number

// Driver Class
class Test {
    // method to find factorial
    // of given number
    static int factorial(int n)
    {
        if (n == 0)
            return 1;

        return n * factorial(n - 1);
    }

    // main method
    public static void main(String[] args)
    {
        int num = 5;
        System.out.println("Factorial of " + num + " is "
                           + factorial(5));
    }
}
```

Bad Class Name

Lack of Input Validation

Lack of Iterative Implementation Option

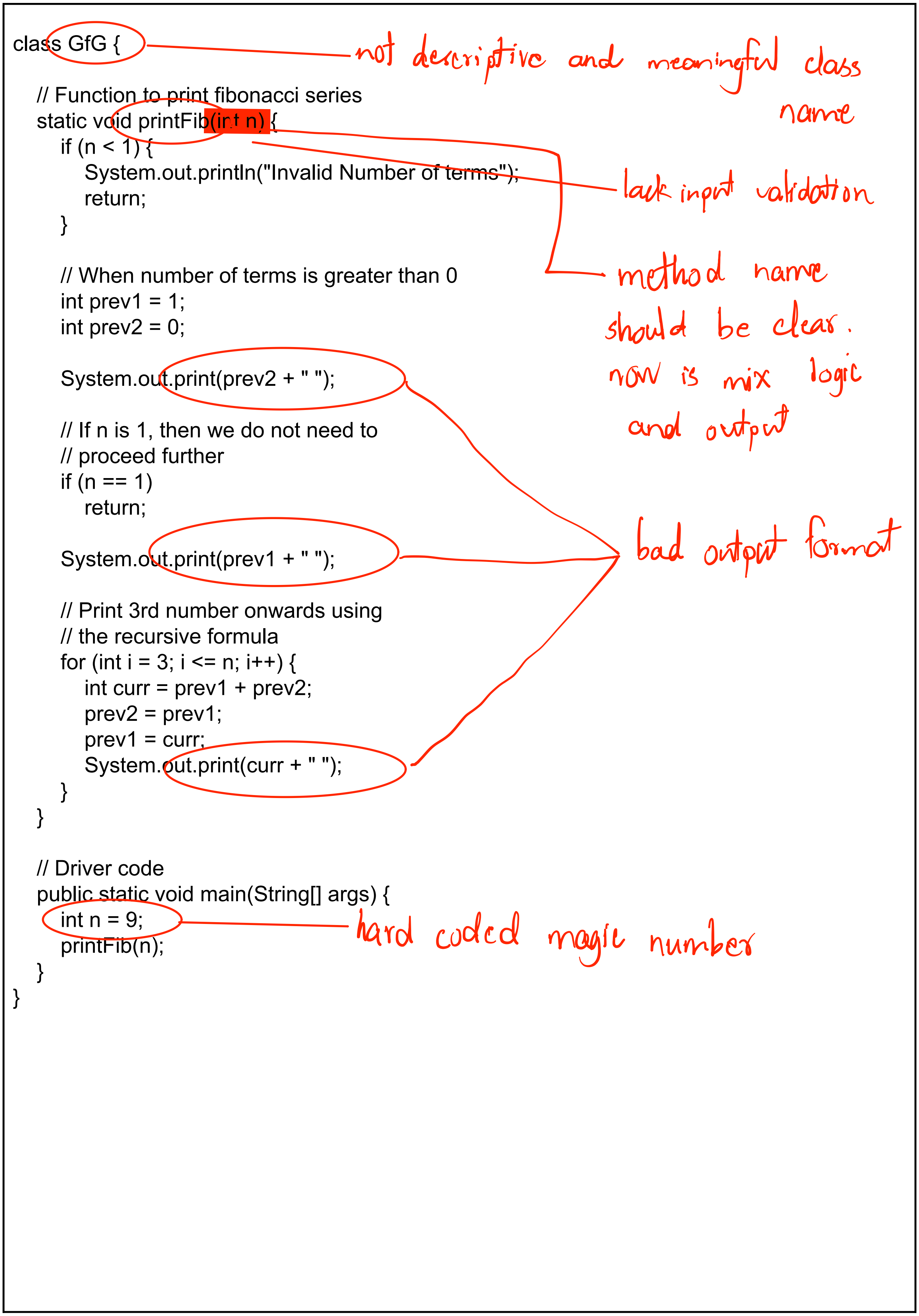
No Handling for Large Inputs

Magic Number

The program after improvement.

```
// Improved Java program to calculate factorial

import java.math.BigInteger; // Required for handling large factorials

// Class name updated for better readability
class FactorialCalculator {

    // Factorial method with input validation and BigInteger for large numbers
    static BigInteger factorial(int number) {
        if (number < 0) {
            throw new IllegalArgumentException("Factorial is not defined for negative numbers.");
        }

        BigInteger resultFactorials = BigInteger.ONE;

            int round = 1
 for (round <= number; round++) {
            resultFactorials = resultFactorials.multiply(BigInteger.valueOf(i));
        }
        return resultFactorials;
    }

    // Main method
    public static void main(String[] args) {
        int number = 5; // Avoid magic numbers, make it user-defined if necessary
        System.out.println("Factorial of " + number + " is " + factorial(number));
    }
}
```

The source code originates from: ***https://www.geeksforgeeks.org/fibonacci-series/***



```
class GfG {

    // Function to print fibonacci series
    static void printFib(int n) {
        if (n < 1) {
            System.out.println("Invalid Number of terms");
            return;
        }

        // When number of terms is greater than 0
        int prev1 = 1;
        int prev2 = 0;

        System.out.print(prev2 + " ");

        // If n is 1, then we do not need to
        // proceed further
        if (n == 1)
            return;

        System.out.print(prev1 + " ");

        // Print 3rd number onwards using
        // the recursive formula
        for (int i = 3; i <= n; i++) {
            int curr = prev1 + prev2;
            prev2 = prev1;
            prev1 = curr;
            System.out.print(curr + " ");
        }
    }

    // Driver code
    public static void main(String[] args) {
        int n = 9;
        printFib(n);
    }
}
```

not descriptive and meaningful class name

lack input validation

method name should be clear. now is mix logic and output

bad output format

hard coded magic number

The program after improvement.

```java
import java.util.Scanner;

public class FibonacciPrinter {

    // Generates the Fibonacci sequence up to n terms and returns an array
    public static int[] generateFibonacci(int n) {
        if (n <= 0) {
            throw new IllegalArgumentException("Number of terms must be positive.");
        }

        int[] fibSeries = new int[n];
        fibSeries[0] = 0;

        if (n > 1) {
            fibSeries[1] = 1;
        }

        for (int i = 2; i < n; i++) {
            fibSeries[i] = fibSeries[i - 1] + fibSeries[i - 2];
        }

        return fibSeries;
    }

    // Prints the Fibonacci series
    public static void printFibonacciSeries(int[] series) {
        System.out.print("Fibonacci Series: ");
        for (int num : series) {
            System.out.print(num + " ");
        }
        System.out.println(); // newline for cleanliness
    }

    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.print("Enter number of terms to generate Fibonacci Series: ");
            if (scanner.hasNextInt()) {
                int n = scanner.nextInt();
                int[] fib = generateFibonacci(n);
                printFibonacciSeries(fib);
            } else {
                System.out.println("Invalid input. Please enter a positive integer.");
            }
        } catch (Exception e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
    }
}
```