**Design Principles & Patterns**

Week 1

Exercise 1 :Implementing the Singleton Pattern

Code:

Logger.java

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger instance created.");

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

        System.out.println("Log: " + message);

    }

}

LoggerTest.java

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("First log message");

        logger2.log("Second log message");

        if (logger1 == logger2) {

            System.out.println("Both logger instances are the same (Singleton works).");

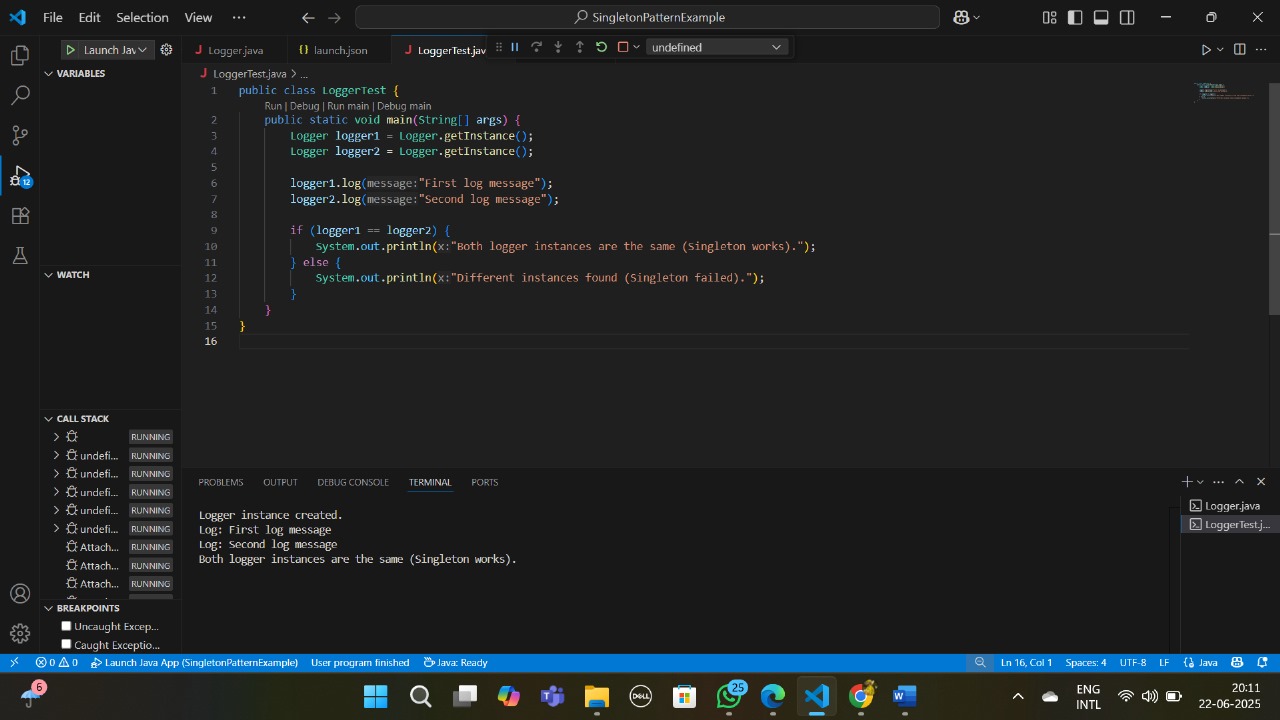
        } else {

            System.out.println("Different instances found (Singleton failed).");

        }

    }

}



Exercise 2 :FactoryMethodPatternExample