**Exercise 1: Implementing the Singleton Pattern**

* **Step-1:** Created a Java project named **SingletonPatternExample**
* **Step-2**:

**Logger.java**

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger initialized through constructor");

    }

    public static Logger getInstanceOf\_Logger() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log\_Msg(String message) {

        System.out.println(message);

    }

}

**Singleton\_Test.java**

public class Singleton\_Test {

    public static void main(String[] args) {

        Logger log1 = Logger.getInstanceOf\_Logger();

        Logger log2 = Logger.getInstanceOf\_Logger();

        log1.log\_Msg("First logger instance message.");

        log2.log\_Msg("Second loger instance  message.");

        if (log1 == log2) {

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

            System.out.println("Singleton pattern is working.");

        } else {

            System.out.println("Different logger instances.");

            System.out.println("Not singleton class");

        }

    }

}

**Output:**

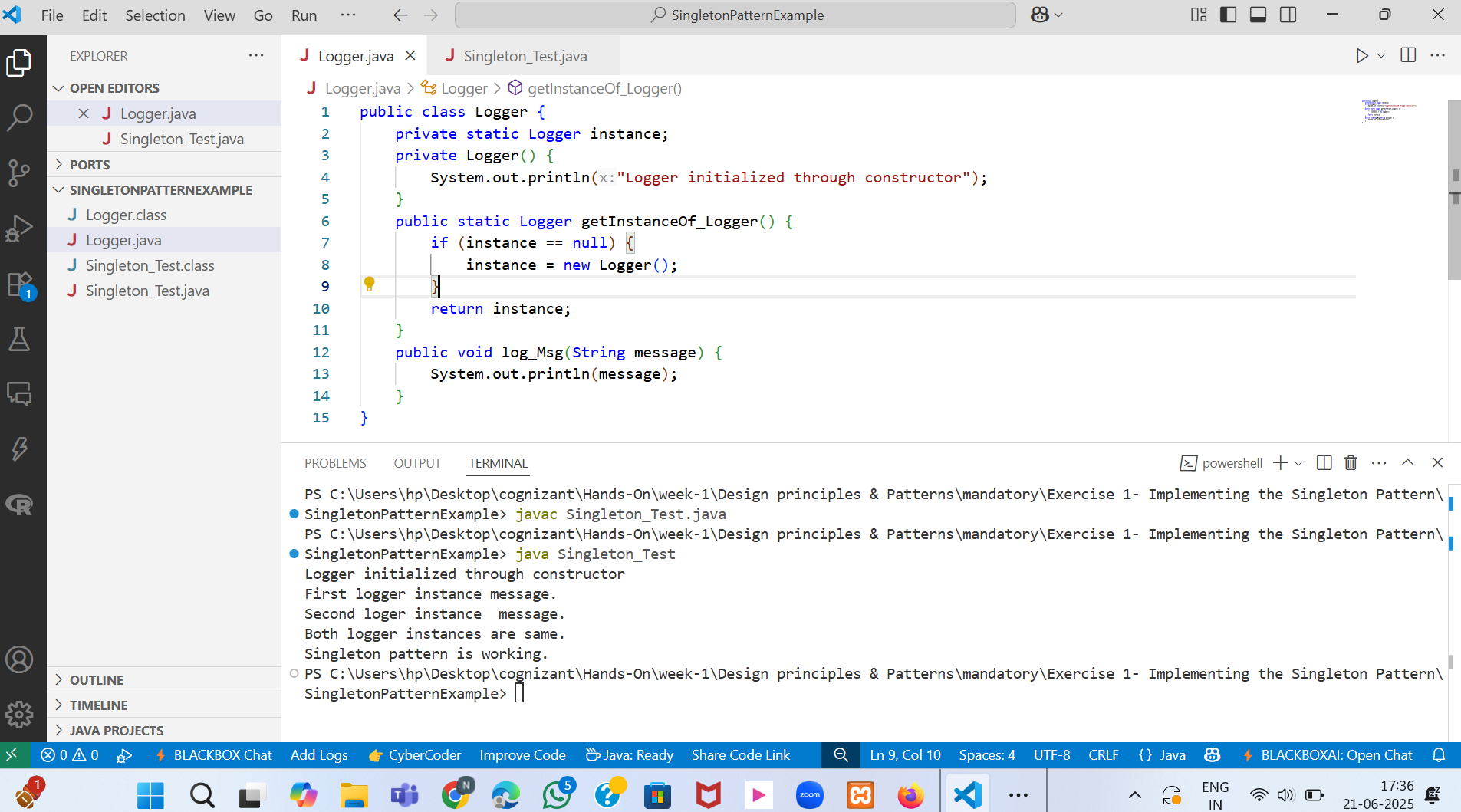
****

Fig-1 Logger.java

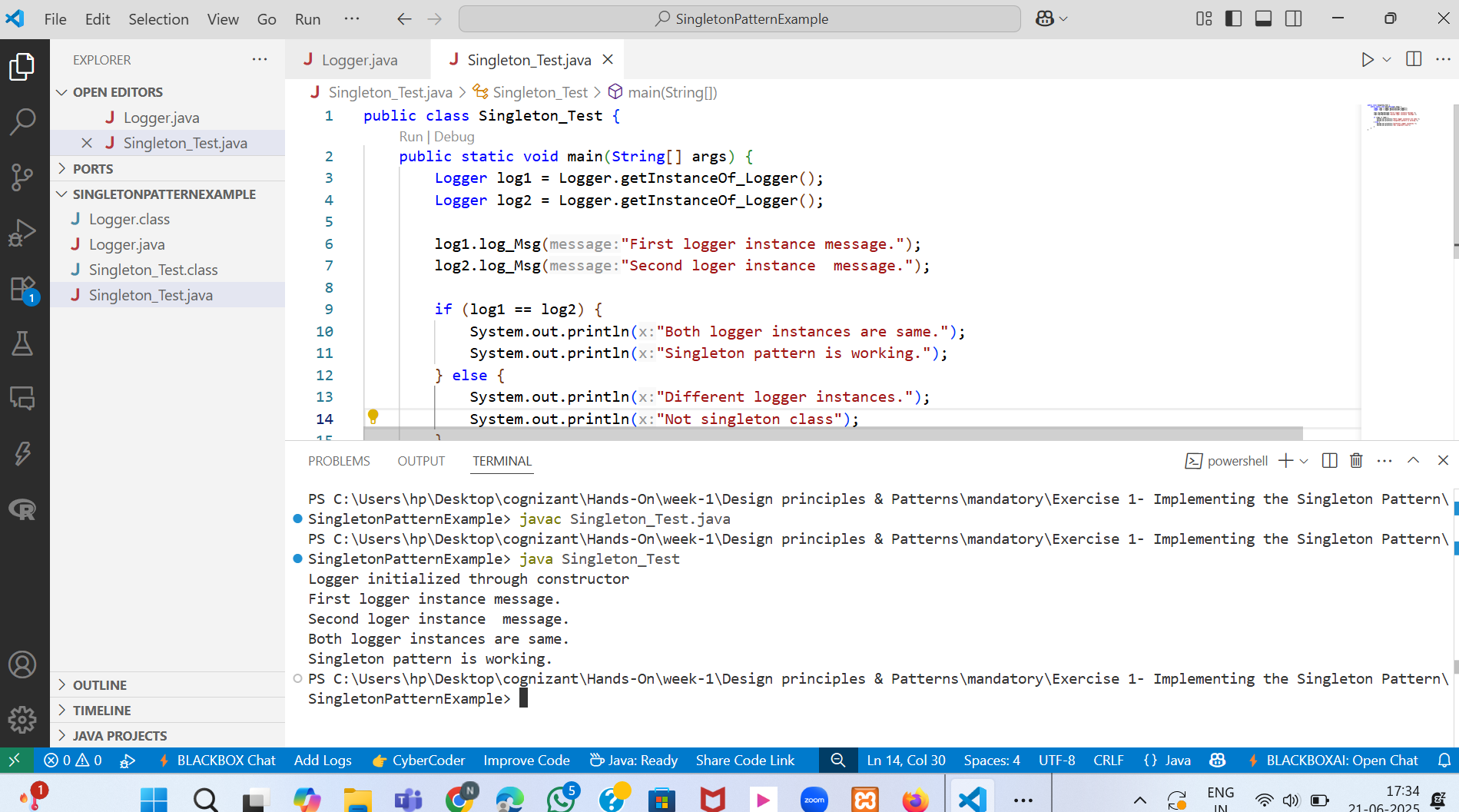


Fig-2 Singleton\_Test.java with program output