**Week 1**: **Exercise 1: Implementing the Singleton Pattern**

**Superset Id:6362819  
  
  
  
1.**Project Name: SingletonPatternExample

**2,3:**

public class Logger {

    // 2.1: Private static instance of the class

    private static Logger instance;

    // 2.2: Private constructor to restrict direct instantiation

    private Logger() {

        System.out.println("Logger started.");

    }

    // 2.3: Public static method to get the single instance

    public static Logger getInstance() {

        if (instance == null) {

            synchronized (Logger.class) {

                if (instance == null) {

                    instance = new Logger();

                }

            }

        }

        return instance;

    }

    // Optional utility method to simulate logging

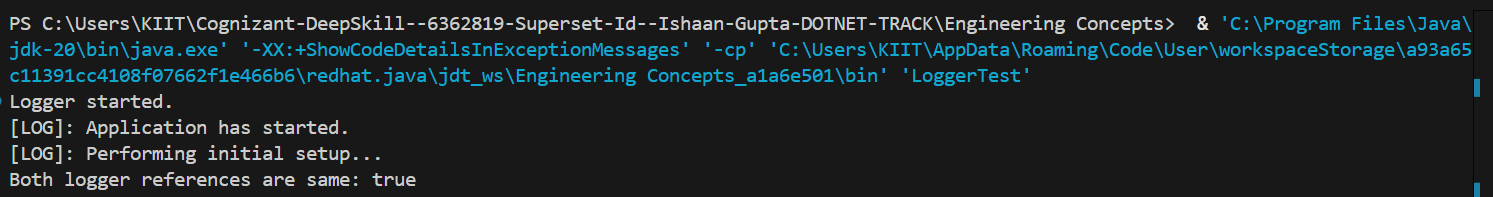
    public void log(String message) {

        System.out.println("[LOG]: " + message);

    }

}

Output:



**4:**

public class LoggerTest {

    public static void main(String[] args) {

        // 4.1: Get first logger reference

        Logger logger1 = Logger.getInstance();

        logger1.log("Application has started.");

        // 4.2: Get second logger reference

        Logger logger2 = Logger.getInstance();

        logger2.log("Performing initial setup...");

        // 4.3: Verify if both logger references point to the same object

        boolean sameInstance = logger1 == logger2;

        System.out.println("Both logger references are same: " + sameInstance);

    }

}

Output: