**Project Title: SingletonPatternExample**

**Objective:** Implement the Singleton Design Pattern to ensure only one instance of a Logger utility class exists and is used throughout the application.

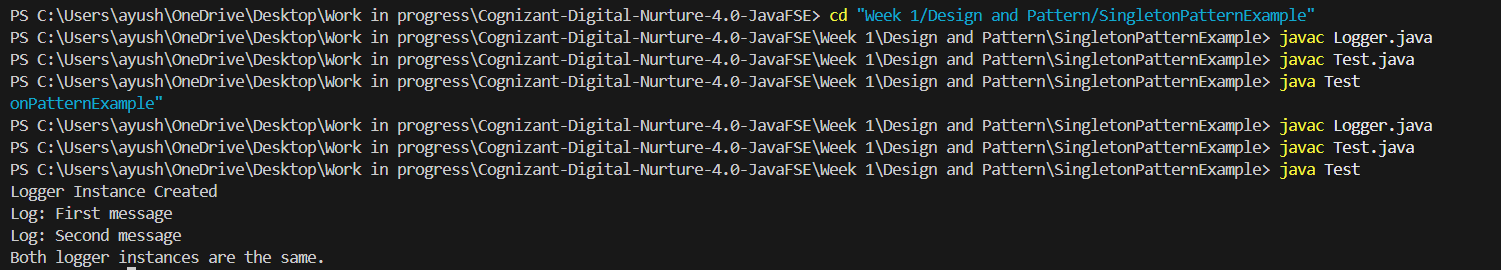
**Design Pattern Used:**

Singleton Pattern — ensures a class has only one instance and provides a global point of access to it.

**Implementation Details:**

1. **Logger Class:**
   * Private static instance: private static Logger instance;
   * Private constructor to prevent instantiation from outside.
   * Public static method getInstance() to provide access to the single instance.
   * log(String message) method to print log messages.
2. **Test Class (Main.java):**
   * Calls Logger.getInstance() multiple times.
   * Confirms both variables (logger1, logger2) point to the same object.
3. **Navigate to the file and then compile and run it.**

**Output Demonstration:**

****

**Conclusion:**

This implementation confirms the Logger class successfully follows the Singleton Design Pattern — only one instance is created and reused.