**DESIGN PATTERNS AND PRINCIPLES:**

**Exercise 1: Implementing the Singleton Pattern**

**Step 1: Create a Java Project**

* Project Name: SingletonPatternExample

**Step 2: Define the Singleton Logger Class**

**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;

}

// Sample logging method

public void log(String message) {

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

}

}

**Step 3: Test the Singleton Implementation:**

**TestSingleton.java**

public class TestSingleton {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log message");

Logger logger2 = Logger.getInstance();

logger2.log("Second log message");

// Checking if both instances are the same

if (logger1 == logger2) {

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

} else {

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

}

}

}

**OUTPUT:**

