**DESIGN PATTERN AND PRINCIPLES**

**Exercise 1: Implementing the Singleton Pattern**

**1. Logger.java:**

public class Logger {

// Private static instance of class Logger

private static Logger instance;

// Private constructor

private Logger() {

System.out.println("Initialized");

}

// Public static method to get the singleton instance

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

// Method to log messages

public void log(String message) {

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

}

}

**2. Test.java**

public class Test {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First message");

Logger logger2 = Logger.getInstance();

logger2.log("Second message");

// Check if both logger instances are the same

if (logger1 == logger2) {

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

} else {

System.out.println("Singleton failed.");

}

}

}

