**Design Patterns and Principles**

Exercise 1: Implementing the Singleton Pattern

**Code:**

* Logger class

public class Logger {

private static Logger singleInstance = null;

private Logger() {

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

}

public static Logger getInstance() {

if (singleInstance == null) {

singleInstance = new Logger();

}

return singleInstance;

}

public void log(String message) {

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

}

}

* Test the Singleton Implementation:

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

logger1.log("First message");

logger2.log("Second message");

if (logger1 == logger2) {

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

} else {

System.out.println("Different instances exist (Singleton failed!)");

}

}

}

* OUTPUT

