Design Patterns and Principles

Exercise 1: Implementing the Singleton Pattern

**Logger.java**

package Singleton;

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*;

}

public void log(String message) {

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

}

}

**Main.java**

package Singleton;

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

logger1.log("First log message.");

logger2.log("Second log message.");

if (logger1 == logger2) {

System.*out*.println("Both logger1 and logger2 are the same instance.");

} else {

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

}

}

}

OUTPUT:

