**Exercise 1:** Implementing the Singleton Patterns

**Program** :

import java.util.\*;

public class SingletonPatternExample

{

public static class Logger

{

private static volatile Logger instance;

// Private constructor

private Logger() {

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

}

public static Logger getInstance() {

if (instance == null) {

synchronized (Logger.class) {

if (instance == null) {

instance = new Logger();

}

}

}

return instance;

}

// Logging method

public void log(String message) {

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

}

}

// 4. Test in main method

public static void main(String[] args) {

Logger log1 = Logger.getInstance();

Logger log2 = Logger.getInstance();

log1.log("Starting application");

log2.log("Processing data");

// Verify that both references refer to the same instance

System.out.println("Same instance? " + (log1 == log2));

}

}