**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 method to provide access to the instance

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger();

}

return *instance*;

}

// Log method

public void log(String message) {

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

}

}

Main.java

package Singleton;

public class Main {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

logger1.log("This is the first log message.");

logger2.log("This is the second log message.");

// Check if both references are the same

if (logger1 == logger2) {

System.***out***.println("Both logger instances are the same and Singleton is verified.");

} else {

System.***out***.println("Different logger instances and Singleton is failed");

}

}

}

Output:

Logger Instance Created

Log: This is the first log message.

Log: This is the second log message.

Both logger instances are the same and Singleton is verified.