using System;

namespace SingletonPatternExample {

public class Logger {

private static Logger?\_instance;

private static readonly object \_lock=new();

private Logger() {

Console.WriteLine("Logger started");

}

public static Logger GetInstance() {

if (\_instance==null) {

lock (\_lock) {

if (\_instance==null) {

\_instance=new Logger();

}

}

}

return \_instance;

}

public void Log(string msg) {

Console.WriteLine("LOG: "+msg);

}

}

class Program {

static void Main(string[] args) {

var logger1=Logger.GetInstance();

logger1.Log("logger1 initialized");

var logger2=Logger.GetInstance();

logger2.Log("logger2 initialized");

if (logger1==logger2) {

Console.WriteLine("Same instance is initialized");

} else {

Console.WriteLine("Different instances");

}

}

}

}