# Singleton Pattern in C#

This document explains and demonstrates the Singleton Design Pattern using a Logger class in C#.

## Logger.cs

namespace SingletonPatternExample  
{  
 public class Logger  
 {  
 private static Logger instance = null;  
  
 private Logger()  
 {  
 Console.WriteLine("Logger Initialized.");  
 }  
  
 public static Logger GetInstance()  
 {  
 if (instance == null)  
 {  
 instance = new Logger();  
 }  
 return instance;  
 }  
  
 public void Log(string message)  
 {  
 Console.WriteLine("[LOG] " + message);  
 }  
 }  
}

## Program.cs

using System;  
  
namespace SingletonPatternExample  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 Logger logger1 = Logger.GetInstance();  
 logger1.Log("Starting the application...");  
  
 Logger logger2 = Logger.GetInstance();  
 logger2.Log("Continuing the application...");  
  
 if (logger1 == logger2)  
 {  
 Console.WriteLine("✅ Only one instance of Logger is used.");  
 }  
 else  
 {  
 Console.WriteLine("❌ Different instances found. Singleton failed!");  
 }  
 }  
 }  
}



