

Singleton Pattern in C#

Project Structure

```
SingletonPatternExample/  
├── Logger.cs           // Singleton Logger class  
└── Program.cs         // Test class (Main method)
```

1. Singleton Class: `Logger.cs`

```
using System;  
  
namespace SingletonPatternExample  
{  
    public class Logger  
    {  
        private static Logger? instance;  
        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);  
        }  
    }  
}
```

2. Test Class: `Program.cs`

```
// Program.cs  
using System;  
  
namespace SingletonPatternExample  
{  
    class Program  
    {  

```

```

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)
    {
        Console.WriteLine("Both instances are the same. Singleton
confirmed.");
    }
    else
    {
        Console.WriteLine("Instances are different. Singleton
broken.");
    }
}
}

```

3. Output

The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code is the same as shown in the previous block. The 'TERMINAL' pane at the bottom displays the output of the program:

```

PS D:\Cognizant\Week 1\Hand_On> cd SingletonPatternExample
PS D:\Cognizant\Week 1\Hand_On\SingletonPatternExample> dotnet run
>>
Logger Initialized.
Log: First log message.
Log: Second log message.
Both instances are the same. Singleton confirmed.
PS D:\Cognizant\Week 1\Hand_On\SingletonPatternExample>

```

The output confirms that both instances of the Logger are the same, as expected for a Singleton pattern.