Cognizant Deep Skilling - Week 1

# 22052918-Pranjal Yadav

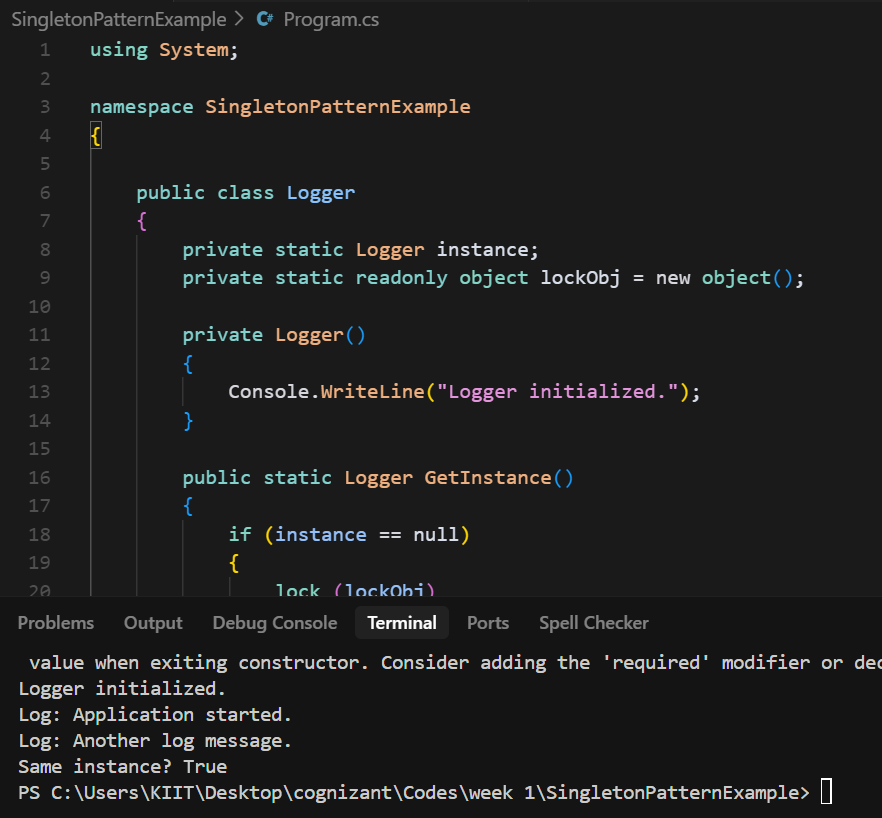
# Exercise 1: Implementing the Singleton Pattern

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

## Code Implementation

using System;  
  
namespace SingletonPatternExample  
{  
 public class Logger  
 {  
 private static Logger instance;  
 private static readonly object lockObj = new object();  
  
 private Logger()  
 {  
 Console.WriteLine("Logger initialized.");  
 }  
  
 public static Logger GetInstance()  
 {  
 if (instance == null)  
 {  
 lock (lockObj)  
 {  
 if (instance == null)  
 {  
 instance = new Logger();  
 }  
 }  
 }  
 return instance;  
 }  
  
 public void Log(string message)  
 {  
 Console.WriteLine("Log: " + message);  
 }  
 }  
  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 Logger logger1 = Logger.GetInstance();  
 logger1.Log("Application started.");  
  
 Logger logger2 = Logger.GetInstance();  
 logger2.Log("Another log message.");  
  
 Console.WriteLine($"Same instance? {ReferenceEquals(logger1, logger2)}");  
 }  
 }  
}

## Execution Output



Screenshot showing successful execution of Singleton Pattern implementation.