Week 1 Hands-On Report

Roll No: 22052918

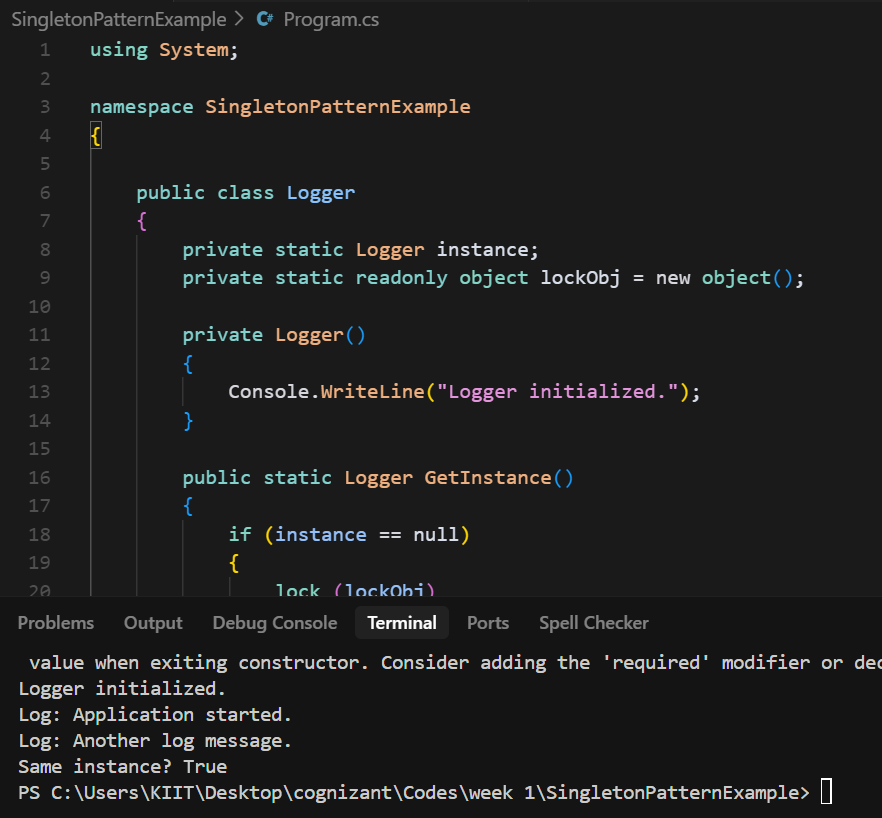
# Exercise 1: Singleton Pattern

Skill Area: Design Principles & Patterns

## Code

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)}");  
 }  
 }  
}

## Output Screenshot



# Exercise 2: Factory Method Pattern

Skill Area: Design Principles & Patterns

## Code

using System;  
  
namespace FactoryMethodPattern  
{  
 public interface IProduct  
 {  
 string GetDetails();  
 }  
  
 public class ConcreteProductA : IProduct  
 {  
 public string GetDetails()  
 {  
 return "This is Product A.";  
 }  
 }  
  
 public class ConcreteProductB : IProduct  
 {  
 public string GetDetails()  
 {  
 return "This is Product B.";  
 }  
 }  
  
 public abstract class Creator  
 {  
 public abstract IProduct FactoryMethod();  
 }  
  
 public class ConcreteCreatorA : Creator  
 {  
 public override IProduct FactoryMethod()  
 {  
 return new ConcreteProductA();  
 }  
 }  
  
 public class ConcreteCreatorB : Creator  
 {  
 public override IProduct FactoryMethod()  
 {  
 return new ConcreteProductB();  
 }  
 }  
  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 Creator creatorA = new ConcreteCreatorA();  
 IProduct productA = creatorA.FactoryMethod();  
 Console.WriteLine(productA.GetDetails());  
  
 Creator creatorB = new ConcreteCreatorB();  
 IProduct productB = creatorB.FactoryMethod();  
 Console.WriteLine(productB.GetDetails());  
 }  
 }  
}

## Output Screenshot

