**Exercise 1: Implementing the Singleton Pattern**

**Program.cs**

**using System;**

**class Program**

**{**

**static void Main()**

**{**

**Logger logger1 = Logger.Instance;**

**Logger logger2 = Logger.Instance;**

**logger1.Log("First log message.");**

**logger2.Log("Second log message.");**

**Console.WriteLine(object.ReferenceEquals(logger1, logger2));**

**}**

**}**

**Logger.cs**

**public sealed class Logger**

**{**

**private static Logger \_instance = null;**

**private static readonly object \_lock = new object();**

**private Logger() { }**

**public static Logger Instance**

**{**

**get**

**{**

**lock (\_lock)**

**{**

**if (\_instance == null)**

**\_instance = new Logger();**

**return \_instance;**

**}**

**}**

**}**

**public void Log(string message)**

**{**

**Console.WriteLine($"Log: {message}");**

**}**

**}**

**SingletonPatternExample.csproj**

**<Project Sdk="Microsoft.NET.Sdk">**

**<PropertyGroup>**

**<OutputType>Exe</OutputType>**

**<TargetFramework>net9.0</TargetFramework>**

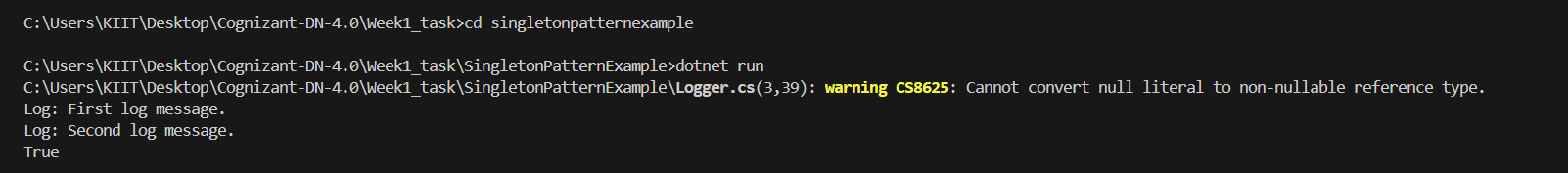
**<ImplicitUsings>enable</ImplicitUsings>**

**<Nullable>enable</Nullable>**

**</PropertyGroup>**

**</Project>**

**Output:-**

****

**Exercise 2: Implementing the Factory Method Pattern**

**Program.cs**

**using System;**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**DocumentFactory factory;**

**factory = new WordDocumentFactory();**

**IDocument wordDoc = factory.CreateDocument();**

**wordDoc.Open();**

**factory = new PdfDocumentFactory();**

**IDocument pdfDoc = factory.CreateDocument();**

**pdfDoc.Open();**

**factory = new ExcelDocumentFactory();**

**IDocument excelDoc = factory.CreateDocument();**

**excelDoc.Open();**

**}**

**}**

**IDocument.cs**

**public interface IDocument**

**{**

**void Open();**

**}**

**PdfDocument.cs**

**public class PdfDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening a PDF document.");**

**}**

**}**

**PdfDocumentFactory.cs**

**public class PdfDocumentFactory : DocumentFactory**

**{**

**public override IDocument CreateDocument()**

**{**

**return new PdfDocument();**

**}**

**}**

**ExcelDocument.cs**

**public class ExcelDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening an Excel document.");**

**}**

**}**

**ExcelDocumentFactory.cs**

**public class ExcelDocumentFactory : DocumentFactory**

**{**

**public override IDocument CreateDocument()**

**{**

**return new ExcelDocument();**

**}**

**}**

**DocumentFactory.cs**

**public abstract class DocumentFactory**

**{**

**public abstract IDocument CreateDocument();**

**}**

**WordDocument.cs**

**public class WordDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening a Word document.");**

**}**

**}**

**WordDocumentFactory.cs**

public class WordDocumentFactory : DocumentFactory

{

public override IDocument CreateDocument()

{

return new WordDocument();

}

}

**FactoryMethodPatternExample.csproj**

**<Project Sdk="Microsoft.NET.Sdk">**

**<PropertyGroup>**

**<OutputType>Exe</OutputType>**

**<TargetFramework>net9.0</TargetFramework>**

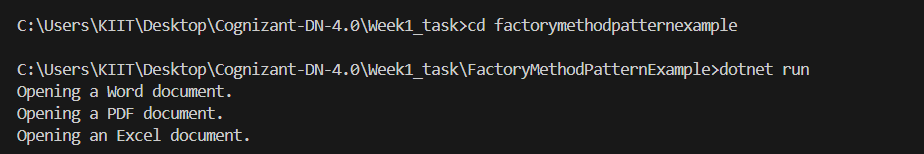
**<ImplicitUsings>enable</ImplicitUsings>**

**<Nullable>enable</Nullable>**

**</PropertyGroup>**

**</Project>**

**Output:-**

****