**WEEK 1  
  
Design principles & Patterns  
  
Exercise 1: Implementing the Singleton Pattern  
  
CODE  
  
-Logger.cs  
  
 using System;**

**public class Logger**

**{**

**private static Logger? instance = null; // ✅**

**private static readonly object padlock = new object();**

**private Logger()**

**{**

**Console.WriteLine("Logger Initialized.");**

**}**

**public static Logger Instance**

**{**

**get**

**{**

**lock (padlock)**

**{**

**if (instance == null)**

**{**

**instance = new Logger();**

**}**

**return instance;**

**}**

**}**

**}**

**public void Log(string message)**

**{**

**Console.WriteLine($"[LOG]: {message}");**

**}**

**}**

**-Program.cs  
  
  
using System;**

**class Program**

**{**

**static void Main()**

**{**

**Logger logger1 = Logger.Instance;**

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

**Logger logger2 = Logger.Instance;**

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

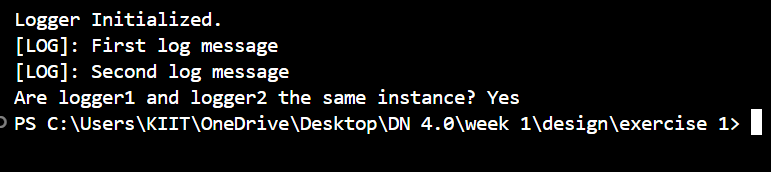
**Console.WriteLine("Are logger1 and logger2 the same instance? " +**

**(object.ReferenceEquals(logger1, logger2) ? "Yes" : "No"));**

**}**

**}**

**OUTPUT**

****

**Exercise 2: Implementing the Factory Method Pattern  
  
CODE-  
  
-DocumentFactory.cs  
  
 public abstract class DocumentFactory**

**{**

**public abstract IDocument CreateDocument();**

**}**

**-ExcelDocument.cs  
   
 using System;**

**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();**

**}**

**}**

**-IDocument.cs  
  
 public interface IDocument**

**{**

**void Open();}**

**-pdfDocument.cs  
  
 using System;**

**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();**

**}**

**}**

**-Program.cs  
  
 using System;**

**class Program**

**{**

**static void Main()**

**{**

**DocumentFactory wordFactory = new WordDocumentFactory();**

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

**wordDoc.Open();**

**DocumentFactory pdfFactory = new PdfDocumentFactory();**

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

**pdfDoc.Open();**

**DocumentFactory excelFactory = new ExcelDocumentFactory();**

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

**excelDoc.Open();**

**}**

**}**

**-WordDocument.cs  
  
 using System;**

**public class WordDocument : IDocument**

**{**

**public void Open()**

**{**

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

**}**

**}**

**-WordDocumentFactor.cs**

**public class WordDocumentFactory : DocumentFactory**

**{**

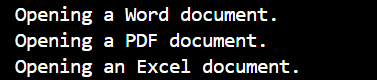
**public override IDocument CreateDocument()**

**{**

**return new WordDocument();**

**}**

**}**

**OUTPUT-  
  
**