**Exercise 1: Implementing the Singleton Pattern**

**CODE:-**

**Logger.cs:-**

**using System;**

**public sealed class Logger**

**{**

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

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

**private Logger()**

**{**

**Console.WriteLine("Logger instance created.");**

**}**

**public static Logger Instance**

**{**

**get**

**{**

**lock (\_lock)**

**{**

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

**{**

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

**}**

**return \_instance;**

**}**

**}**

**}**

**public void Log(string message)**

**{**

**Console.WriteLine($"[Log] {DateTime.Now}: {message}");**

**}**

**}**

**CODE:-**

**Program.cs:-**

**using System;**

**class Program**

**{**

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

**{**

**Console.WriteLine("Testing Singleton Logger\n");**

**Logger logger1 = Logger.Instance;**

**logger1.Log("This is the first log message.");**

**Logger logger2 = Logger.Instance;**

**logger2.Log("This is the second log message.");**

**if (ReferenceEquals(logger1, logger2))**

**{**

**Console.WriteLine("\nlogger1 and logger2 are the same instance.");**

**}**

**else**

**{**

**Console.WriteLine("\nlogger1 and logger2 are different instances.");**

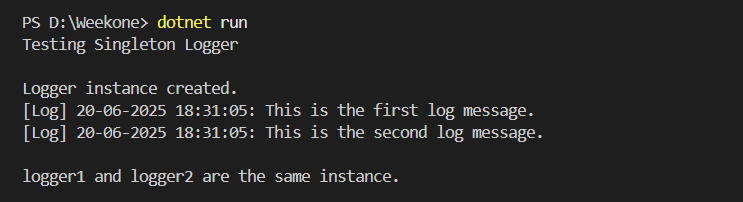
**}**

**Console.ReadLine();**

**}**

**}**

**OUTPUT SCREENSHOTS:-**



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

**CODE:-**

**Program.cs:-**

**using System;**

**namespace FactoryMethodPatternExample**

**{**

**class Program**

**{**

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

**{**

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

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

**word.Open();**

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

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

**pdf.Open();**

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

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

**excel.Open();**

**Console.ReadLine();**

**}**

**}**

**}**

**IDocument.cs:-**

**namespace FactoryMethodPatternExample**

**{**

**public interface IDocument**

**{**

**void Open();**

**}**

**}**

**WordDocument.cs:-**

**using System;**

**namespace FactoryMethodPatternExample**

**{**

**public class WordDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening Word Document.");**

**}**

**}**

**}**

**PdfDocument.cs:-**

**using System;**

**namespace FactoryMethodPatternExample**

**{**

**public class PdfDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening PDF Document.");**

**}**

**}**

**}**

**ExcelDocument.cs:-**

**using System;**

**namespace FactoryMethodPatternExample**

**{**

**public class ExcelDocument : IDocument**

**{**

**public void Open()**

**{**

**Console.WriteLine("Opening Excel Document.");**

**}**

**}**

**}**

**DocumentFactory.cs:-**

**namespace FactoryMethodPatternExample**

**{**

**public abstract class DocumentFactory**

**{**

**public abstract IDocument CreateDocument();**

**}**

**}**

**WordDocumentFactory.cs:-**

**namespace FactoryMethodPatternExample**

**{**

**public class WordDocumentFactory : DocumentFactory**

**{**

**public override IDocument CreateDocument()**

**{**

**return new WordDocument();**

**}**

**}**

**}**

**PdfDocumentFactory.cs:-**

**namespace FactoryMethodPatternExample**

**{**

**public class PdfDocumentFactory : DocumentFactory**

**{**

**public override IDocument CreateDocument()**

**{**

**return new PdfDocument();**

**}**

**}**

**}**

**ExcelDocumentFactory.cs:-**

**namespace FactoryMethodPatternExample**

**{**

**public class ExcelDocumentFactory : DocumentFactory**

**{**

**public override IDocument CreateDocument()**

**{**

**return new ExcelDocument();**

**}**

**}**

**}**

**OUTPUT SCREENSHOT:-**

