**DESIGN PRINCIPAL AND PATTERN**

**Exercise 1: Implementing the Singleton Pattern**

using System;

public class Logger

{

private static Logger? instance;

private Logger()

{

Console.WriteLine("Logger Initialized.");

}

public static Logger GetInstance()

{

if (instance == null)

{

instance = new Logger();

}

return instance;

}

public void Log(string message)

{

Console.WriteLine("[LOG]: " + message);

}

}

class Program

{

public static void Main(string[] args)

{

Logger logger1 = Logger.GetInstance();

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

Logger logger2 = Logger.GetInstance();

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

if (logger1 == logger2)

{

Console.WriteLine("Both logger instances are the same (Singleton working).");

}

else

{

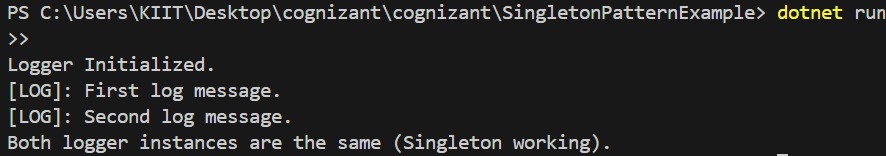
Console.WriteLine("Different logger instances (Singleton failed).");

}

}

}

**OUTPUT:-**



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

**IDocument.cs**

public interface IDocument

{

void Open();

}

**WordDocument.cs**

public class WordDocument : IDocument

{

public void Open()

{

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

}

}

**PdfDocument.cs**

public class PdfDocument : IDocument

{

public void Open()

{

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

}

}

**ExcelDocument.cs**

public class ExcelDocument : IDocument

{

public void Open()

{

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

}

}

**DocumentFactory.cs**

public abstract class DocumentFactory

{

public abstract IDocument CreateDocument();

}

**WordDocumentFactory.cs**

public class WordDocumentFactory : DocumentFactory

{

public override IDocument CreateDocument()

{

return new WordDocument();

}

}

**PdfDocumentFactory.cs**

public class PdfDocumentFactory : DocumentFactory

{

public override IDocument CreateDocument()

{

return new PdfDocument();

}

}

**ExcelDocumentFactory.cs**

public class ExcelDocumentFactory : DocumentFactory

{

public override IDocument CreateDocument()

{

return new ExcelDocument();

}

}

**Program.cs**

using System;

using 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();

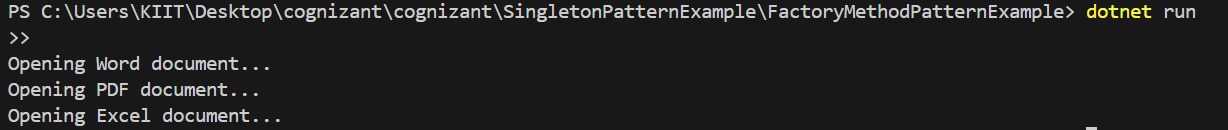
IDocument excel = excelFactory.CreateDocument();

excel.Open();

}

}

**OUTPUT:-**

****