Exercise 1: Implementing the Singleton Pattern

Code:

Program.cs

```
using System;

public class Program
{
    public static void Main(string[] args)
    {
        Logger log1 = Logger.GetInstance();
        log1.Print("This is the First message");

        Logger log2 = Logger.GetInstance();
        log2.Print("This is the Second Message");
    }
}
```

Logger.cs

```
using System;

public class Logger
{
    private static Logger instance = null;

    private Logger()
    {
```

```csharp
        Console.WriteLine("Instance Created");

    }


    public static Logger GetInstance()

    {

        if (instance == null)

        {

            instance = new Logger();

        }

        return instance;

    }


    public void Print(string message)

    {

        Console.WriteLine(message);

    }

}
```
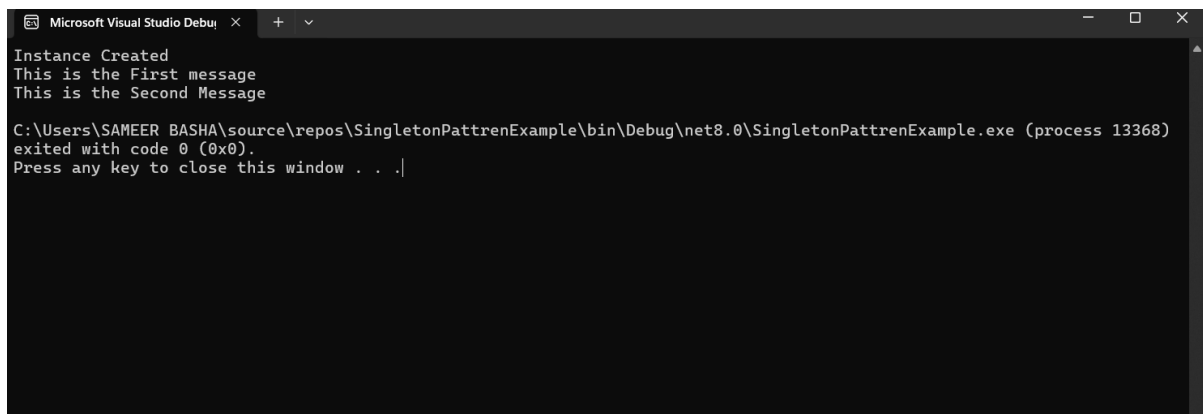
```
Microsoft Visual Studio Debug    ×    +    ∨                                                                              —    □    ×

Instance Created
This is the First message
This is the Second Message

C:\Users\SAMEER BASHA\source\repos\SingletonPattrenExample\bin\Debug\net8.0\SingletonPattrenExample.exe (process 13368)
exited with code 0 (0x0).
Press any key to close this window . . .
```

Exercise 2: Implementing the Factory Method Pattern

IDocument.cs :

```
public interface IDocument
{
    void Open();
}
```

WordDocument.cs:

```
using System;
public class WordDocument : IDocument
{
    public void Open()
    {
        Console.WriteLine("Opening a Word Document.");
    }
}
```

PdfDocument.cs:

```
using System;
public class PdfDocument : IDocument
{
    public void Open()
    {
        Console.WriteLine("Opening a PDF Document.");
    }
}
```

ExcelDocument.cs:

```
using System;
public class ExcelDocument : IDocument
{
```

```csharp
    public void Open()
    {
        Console.WriteLine("Opening an Excel Document.");
    }
}
```

DocumentFactory.cs:

```csharp
public abstract class DocumentFactory
{
    public abstract IDocument CreateDocument();
}
```

WordFactory.cs:

```csharp
public class WordFactory : DocumentFactory
{
    public override IDocument CreateDocument()
    {
        return new WordDocument();
    }
}
```

PdfFactory.cs:

```csharp
public class PdfFactory : DocumentFactory
{
    public override IDocument CreateDocument()
    {
        return new PdfDocument();
    }
}
```

ExcelFactory.cs:

```csharp
public class ExcelFactory : DocumentFactory
{
```

```csharp
    public override IDocument CreateDocument()

    {

        return new ExcelDocument();

    }

}
```

Program.cs:

```csharp
using System;

public class Program

{

    public static void Main(string[] args)

    {

        DocumentFactory wordFactory = new WordFactory();

        IDocument wordDoc = wordFactory.CreateDocument();

        wordDoc.Open();


        DocumentFactory pdfFactory = new PdfFactory();

        IDocument pdfDoc = pdfFactory.CreateDocument();

        pdfDoc.Open();


        DocumentFactory excelFactory = new ExcelFactory();

        IDocument excelDoc = excelFactory.CreateDocument();

        excelDoc.Open();

    }

}
```
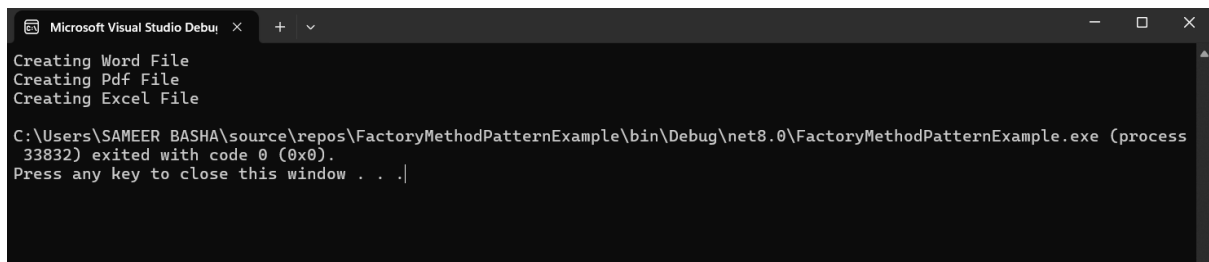
```
Creating Word File
Creating Pdf File
Creating Excel File

C:\Users\SAMEER BASHA\source\repos\FactoryMethodPatternExample\bin\Debug\net8.0\FactoryMethodPatternExample.exe (process
 33832) exited with code 0 (0x0).
Press any key to close this window . . .
```