***WEEK 1\_2: MANDATORY HANDSON EXERCISE (DESIGN PATTERN AND PRINCIPLE)***

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

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Steps:**

1. **Create a New c# Project:**
   * Create a new c# project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **Implement the Factory Method:**
   * Create an abstract class **DocumentFactory** with a method **createDocument()**.
   * Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.

***CODE:***

// IDocument.cs

public interface IDocument

{

    void Open();

    void Save();

}

public class WordDocument : IDocument

{

    public void Open() => Console.WriteLine("Opening Word document...");

    public void Save() => Console.WriteLine("Saving Word document...");

}

public class PdfDocument : IDocument

{

    public void Open() => Console.WriteLine("Opening PDF document...");

    public void Save() => Console.WriteLine("Saving PDF document...");

}

public class ExcelDocument : IDocument

{

    public void Open() => Console.WriteLine("Opening Excel document...");

    public void Save() => Console.WriteLine("Saving Excel document...");

}

// DocumentFactory.cs

public abstract class DocumentFactory

{

    public abstract IDocument CreateDocument();

}

public class WordDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument() => new WordDocument();

}

public class PdfDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument() => new PdfDocument();

}

public class ExcelDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument() => new ExcelDocument();

}

//program.cs

class Program

{

    static void Main()

    {

        // Create factories

        DocumentFactory wordFactory = new WordDocumentFactory();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        // Create documents using factories

        IDocument wordDoc = wordFactory.CreateDocument();

        IDocument pdfDoc = pdfFactory.CreateDocument();

        IDocument excelDoc = excelFactory.CreateDocument();

        // Test document operations

        Console.WriteLine("Testing documents:");

        TestDocument(wordDoc);

        TestDocument(pdfDoc);

        TestDocument(excelDoc);

    }

    static void TestDocument(IDocument doc)

    {

        doc.Open();

        doc.Save();

        Console.WriteLine("-------------------");

    }

}

***OUTPUT:***

