**Factory Method Pattern in C#**

**Project Structure**

FactoryMethodPatternExample/

├── IDocument.cs

├── WordDocument.cs

├── PdfDocument.cs

├── ExcelDocument.cs

├── DocumentFactory.cs

├── WordDocumentFactory.cs

├── PdfDocumentFactory.cs

├── ExcelDocumentFactory.cs

├── DocumentFactorySelector.cs

└── Program.cs

**1. Interface: IDocument.cs**

public interface IDocument

{

    void Open();

}

**2. Concrete Document Classes**

**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");

    }

}

**3. Abstract Factory: DocumentFactory.cs**

public abstract class DocumentFactory

{

    public abstract IDocument CreateDocument();

}

**4. Concrete Factories**

**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();

    }

}

**5. Factory Selector: DocumentFactorySelector.cs**

public static class DocumentFactorySelector

{

    public static DocumentFactory GetFactory(string type)

    {

        switch (type.ToUpper())

        {

            case "WORD":

                return new WordDocumentFactory();

            case "PDF":

                return new PdfDocumentFactory();

            case "EXCEL":

                return new ExcelDocumentFactory();

            default:

                throw new ArgumentException("Invalid document type: " + type);

        }

    }

}

**6. Main Program: Program.cs**

using System;

class Program

{

    static void Main(string[] args)

    {

        Console.Write("Enter document type (WORD, PDF, EXCEL): ");

        string? type = Console.ReadLine()?.Trim();

        try

        {

            DocumentFactory factory = DocumentFactorySelector.GetFactory(type ?? "");

            IDocument doc = factory.CreateDocument();

            doc.Open();

        }

        catch (Exception ex)

        {

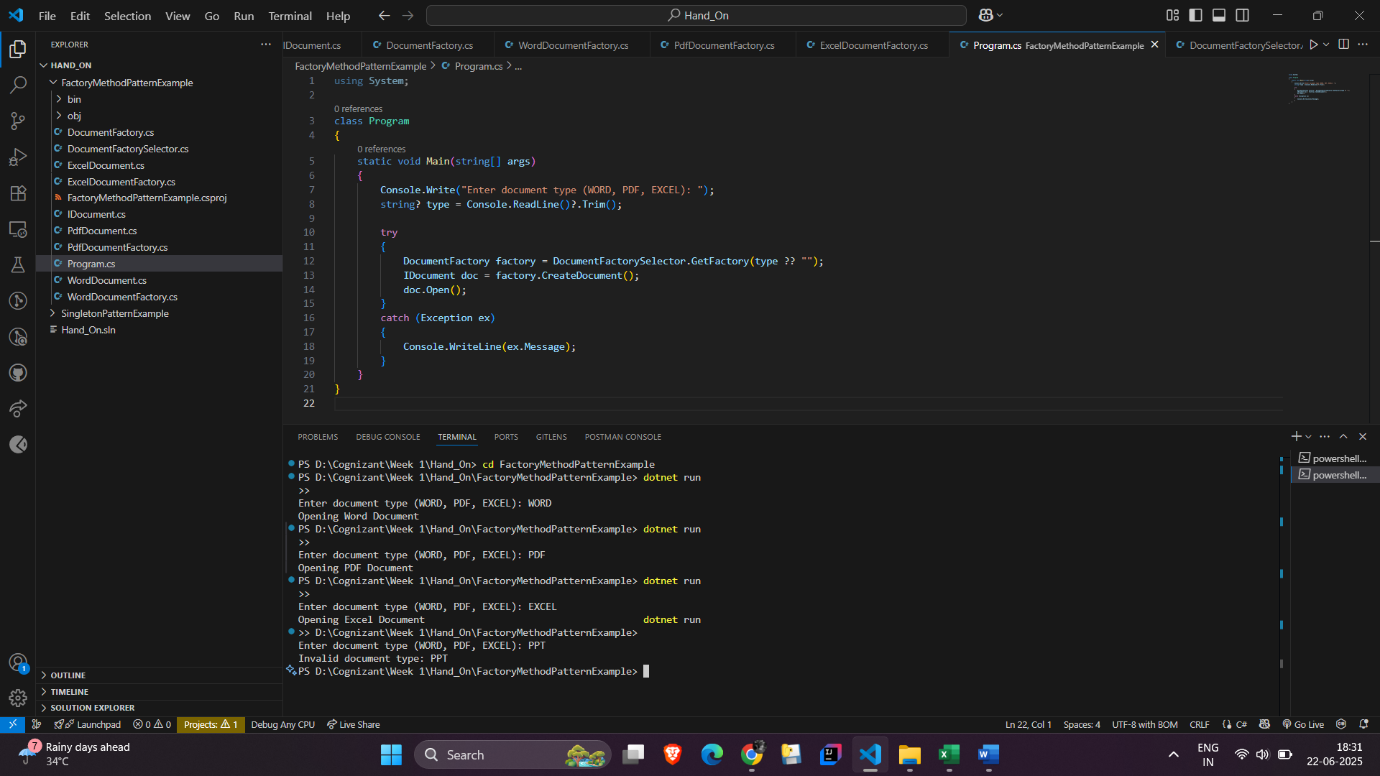
            Console.WriteLine(ex.Message);

        }

    }

}

**7. Output**

****