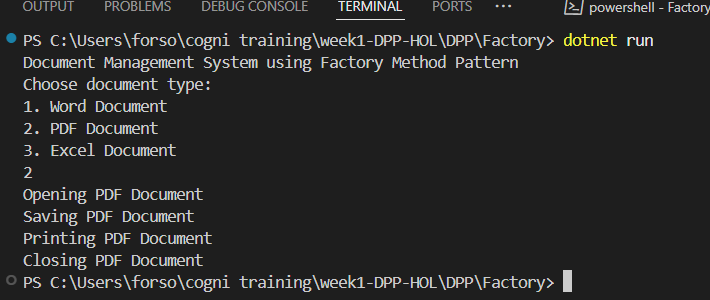
**Exercise 2: Implementing the Factory Method Pattern**

**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.

**OUTPUT:**



**CODE:**

// Exercise 2: Implementing the Factory Method Pattern

// Document Management System

using System;

// Step 2: Define Document Interfaces

public interface IDocument

{

    void Open();

    void Close();

    void Save();

    void Print();

}

// Step 3: Create Concrete Document Classes

public class WordDocument : IDocument

{

    public void Open()

    {

        Console.WriteLine("Opening Word Document");

    }

    public void Close()

    {

        Console.WriteLine("Closing Word Document");

    }

    public void Save()

    {

        Console.WriteLine("Saving Word Document");

    }

    public void Print()

    {

        Console.WriteLine("Printing Word Document");

    }

}

public class PdfDocument : IDocument

{

    public void Open()

    {

        Console.WriteLine("Opening PDF Document");

    }

    public void Close()

    {

        Console.WriteLine("Closing PDF Document");

    }

    public void Save()

    {

        Console.WriteLine("Saving PDF Document");

    }

    public void Print()

    {

        Console.WriteLine("Printing PDF Document");

    }

}

public class ExcelDocument : IDocument

{

    public void Open()

    {

        Console.WriteLine("Opening Excel Document");

    }

    public void Close()

    {

        Console.WriteLine("Closing Excel Document");

    }

    public void Save()

    {

        Console.WriteLine("Saving Excel Document");

    }

    public void Print()

    {

        Console.WriteLine("Printing Excel Document");

    }

}

// Step 4: Implement the Factory Method

public abstract class DocumentFactory

{

    public abstract IDocument CreateDocument();

}

public class WordDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new WordDocument();

    }

}

public class PdfDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new PdfDocument();

    }

}

public class ExcelDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new ExcelDocument();

    }

}

// Step 5: Test the Factory Method Implementation

public class FactoryMethodPatternExample

{

    static void Main(string[] args)

    {

        Console.WriteLine("Document Management System using Factory Method Pattern");

        Console.WriteLine("Choose document type:");

        Console.WriteLine("1. Word Document");

        Console.WriteLine("2. PDF Document");

        Console.WriteLine("3. Excel Document");

        int choice = int.Parse(Console.ReadLine());

        DocumentFactory factory = null;

        switch (choice)

        {

            case 1:

                factory = new WordDocumentFactory();

                break;

            case 2:

                factory = new PdfDocumentFactory();

                break;

            case 3:

                factory = new ExcelDocumentFactory();

                break;

            default:

                Console.WriteLine("Invalid choice!");

                return;

        }

        IDocument document = factory.CreateDocument();

        document.Open();

        document.Save();

        document.Print();

        document.Close();

        Console.ReadKey();

    }

}