**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 Java Project:**
   * Create a new Java 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:**

using System;

interface IDocument

{

void Open();

void Save();

}

class WordDocument : IDocument

{

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

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

}

class PdfDocument : IDocument

{

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

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

}

abstract class DocumentFactory

{

public abstract IDocument CreateDocument();

}

class WordFactory : DocumentFactory

{

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

}

class PdfFactory : DocumentFactory

{

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

}

class Program

{

static void Main()

{

DocumentFactory factory = new WordFactory();

IDocument doc = factory.CreateDocument();

doc.Open();

doc.Save();

}

}

**OUTPUT:**

