# Exercise 2 – Factory Method Pattern

## 1. Objective

To implement a document creation system using the Factory Method Pattern, which supports different types of documents such as Word, PDF, and Excel.

## 2. Problem Statement / 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.

## 3. Approach / Steps

1. Define a `Document` interface with a method `open()`.  
2. Implement `WordDocument`, `PdfDocument`, and `ExcelDocument` classes that implement `Document`.  
3. Create an abstract class `DocumentFactory` with the method `createDocument()`.  
4. Implement `WordDocumentFactory`, `PdfDocumentFactory`, and `ExcelDocumentFactory`.  
5. Create a `Main` class to test the document creation using each factory.

## 4. Code (Simplified)

Document.java

public interface Document {  
 void open();  
}

WordDocument.java

public class WordDocument implements Document {  
 public void open() {  
 System.out.println("Opening a Word document...");  
 }  
}

DocumentFactory.java

public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}

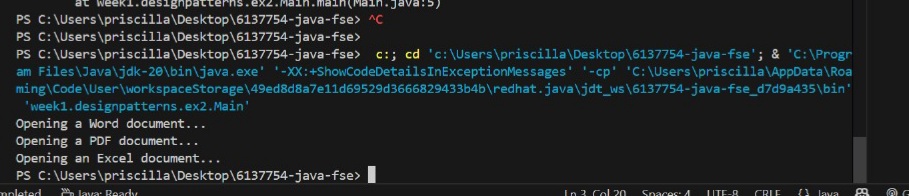
WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}

Main.java

public class Main {  
 public static void main(String[] args) {  
 DocumentFactory factory = new WordDocumentFactory();  
 Document doc = factory.createDocument();  
 doc.open();  
 }  
}

## 5. Output (Screenshot)



## 6. Conclusion

The Factory Method Pattern helps create objects without specifying the exact class. It supports the open/closed principle and makes the system more extensible and maintainable.