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

**Document.java**

package javapro;

public interface Document {

void open();

}

**WordDocument.java**

package javapro;

public class WordDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening Word Document.");

}

}

**PdfDocument.java**

package javapro;

public class PdfDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening PDF Document.");

}

}

**ExcelDocument.java**

package javapro;

public class ExcelDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening Excel Document.");

}

}

**DocumentFactory.java**

package javapro;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

**WordDocumentFactory.java**

package javapro;

public class WordDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new WordDocument();

}

}

**PdfDocumentFactory.java**

package javapro;

public class PdfDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new PdfDocument();

}

}

**ExcelDocumentFactory.java**

package javapro;

public class ExcelDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new ExcelDocument();

}

}

**Main.java**

package javapro;

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**OUTPUT:**

Opening Word Document.

Opening PDF Document.

Opening Excel Document.