**Exercise 2: FactoryMethodPatternExample**

**Code:**

***Document.java***

package assignments.cts;

public interface Document {

void open();

}

***DocumentFactory.java***

package assignments.cts;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

***WordDocument.java***

package assignments.cts;

public class WordDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening a Word document.");

}

}

***WordDocumentFactory.java***

package assignments.cts;

public class WordDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new WordDocument();

}

}

***PdfDocument.java***

package assignments.cts;

public class PdfDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening a PDF document.");

}

}

***PdfDocumentFactory.java***

package assignments.cts;

public class PdfDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new PdfDocument();

}

}

***ExcelDocument.java***

package assignments.cts;

public class ExcelDocument implements Document {

*@Override*

public void open() {

System.***out***.println("Opening an Excel document.");

}

}

***ExcelDocumentFactory.java***

package assignments.cts;

public class ExcelDocumentFactory extends DocumentFactory {

*@Override*

public Document createDocument() {

return new ExcelDocument();

}

}

***Main.java***

package assignments.cts;

public class App

{

public static void main( String[] args )

{

DocumentFactory wf = new WordDocumentFactory();

Document wd = wf.createDocument();

wd.open();

DocumentFactory pf = new PdfDocumentFactory();

Document pd = pf.createDocument();

pd.open();

DocumentFactory ef = new ExcelDocumentFactory();

Document ed = ef.createDocument();

ed.open();

}

}

**Output:**

