**Design Pattern And Principles**

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

**Source Code**

// Document.java

interface Document {

void open();

}

// WordDocument.java

class WordDocument implements Document {

public void open() {

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

}

}

// PdfDocument.java

class PdfDocument implements Document {

public void open() {

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

}

}

// ExcelDocument.java

class ExcelDocument implements Document {

public void open() {

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

}

}

// DocumentFactory.java

abstract class DocumentFactory {

public abstract Document createDocument();

}

// WordFactory.java

class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

// PdfFactory.java

class PdfFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

// ExcelFactory.java

class ExcelFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

// Main.java (Test class)

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelFactory();

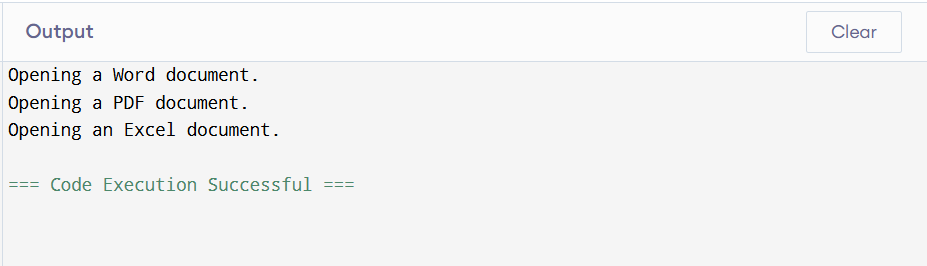
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output**

****