**Design Patterns and Principles**

**Exercise 2: Factory Method Pattern Example**

**SOLUTION**

package com.fse.factory;

interface WordDocument {

    void display();

}

interface PdfDocument {

    void display();

}

interface ExcelDocument {

    void display();

}

class Word implements WordDocument {

    public void display() {

        System.out.println("This is a Word Document.");

    }

}

class Pdf implements PdfDocument {

    public void display() {

        System.out.println("This is a PDF Document.");

    }

}

class Excel implements ExcelDocument {

    public void display() {

        System.out.println("This is an Excel Document.");

    }

}

abstract class DocumentFactory {

    public abstract Object createDocument();

}

class WordFactory extends DocumentFactory {

    public WordDocument createDocument() {

        return new Word();

    }

}

class PdfFactory extends DocumentFactory {

    public PdfDocument createDocument() {

        return new Pdf();

    }

}

class ExcelFactory extends DocumentFactory {

    public ExcelDocument createDocument() {

        return new Excel();

    }

}

public class FactoryMethodPatternExample {

    public static void main(String[] args) {

        WordFactory wordFactory = new WordFactory();

        PdfFactory pdfFactory = new PdfFactory();

        ExcelFactory excelFactory = new ExcelFactory();

        WordDocument word = wordFactory.createDocument();

        word.display();

        PdfDocument pdf = pdfFactory.createDocument();

        pdf.display();

        ExcelDocument excel = excelFactory.createDocument();

        excel.display();

    }

}

**OUTPUT:**

