**Module-1**

**Design patterns and principles**

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

**Document.java:**

public interface Document {

    void open();

}

**DocumentFactory.java:**

public abstract class DocumentFactory {

    public abstract Document createDocument();

**}**

**ExcelDocument.java:**

public class ExcelDocument implements Document {

    public void open() {

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

    }

}

**ExcelDocumentFactory.java:**

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**Main.java:**

public class Main {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word = wordFactory.createDocument();

        word.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf = pdfFactory.createDocument();

        pdf.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel = excelFactory.createDocument();

        excel.open();

    }

}

**PdfDocument.java:**

public class PdfDocument implements Document {

    public void open() {

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

    }

}

**PdfDocumentFactory.java:**

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordDocument.java:**

public class WordDocument implements Document {

    public void open() {

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

    }

}

**WordDocumentFactory.java:**

public class WordDocumentFactory extends DocumentFactory {

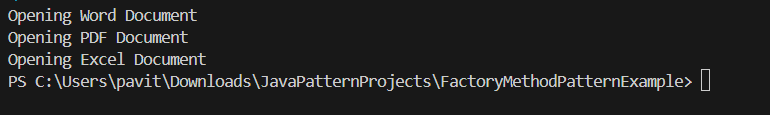
    public Document createDocument() {

        return new WordDocument();

    }

}

**Output:**

****