**HANDS-ON 2:**

**DESIGN PATTERNS AND PRINCIPLES:**

**IMPLEMENTING FACTORY DESIGN PATTERN:**

**CODE:**

interface Document{

void create();

}

class WordDocument implements Document{

*@Override*

public void create() {

System.***out***.println("Word Document is created");

}

}

class PdfDocument implements Document{

*@Override*

public void create() {

System.***out***.println("PDF Document is created");

}

}

class ExcelDocument implements Document{

*@Override*

public void create() {

System.***out***.println("Excel Document is created");

}

}

class DocumentFactory{

public Document createDocument(String type) {

if(type.equals("Word")) {

return new WordDocument();

}

else if(type.equals("Pdf")) {

return new PdfDocument();

}

else if(type.equals("Excel")) {

return new ExcelDocument();

}return null;

}

}

public class Main {

public static void main(String[] args) {

DocumentFactory Factory = new DocumentFactory();

Document d1 = Factory.createDocument("Word");

d1.create();

Document d2=Factory.createDocument("Pdf");

d2.create();

Document d3=Factory.createDocument("Excel");

d3.create();

}

}

**OUTPUT:**

Word Document is created

PDF Document is created

Excel Document is created