**Week-1**

**Design Pattern and Principles**

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

**Main.java**

public class Main {

public static void main(String[] args) {

test(new WordDocumentFactory());

test(new PdfDocumentFactory());

test(new ExcelDocumentFactory());

}

public static void test(DocumentFactory f) {

concretedoc doc = f.createDocument();

doc.open();

}

}

**concretedoc.java**

public interface concretedoc{

void open();

}

**DocumentFactory.java**

public abstract class DocumentFactory {

public abstract concretedoc createDocument();

}

**ExcelDocument.java**

class ExcelDocument implements concretedoc{

public void open(){

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

}

}

**ExcelDocumentFactory.java**

public class ExcelDocumentFactory extends DocumentFactory{

@Override

public concretedoc createDocument(){

return new ExcelDocument();

}

}

**PdfDocument.java**

class PdfDocument implements concretedoc {

public void open(){

System.out.println("Opening Pdf");

}

}

**PdfDocumentFactory.java**

public class PdfDocumentFactory extends DocumentFactory{

@Override

public concretedoc createDocument(){

return new PdfDocument();

}

}

**WordDocumentFactory.java**

class WordDocument implements concretedoc{

public void open(){

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

}

}

**WordDocumentFactory.java**

public class WordDocumentFactory extends DocumentFactory{

@Override

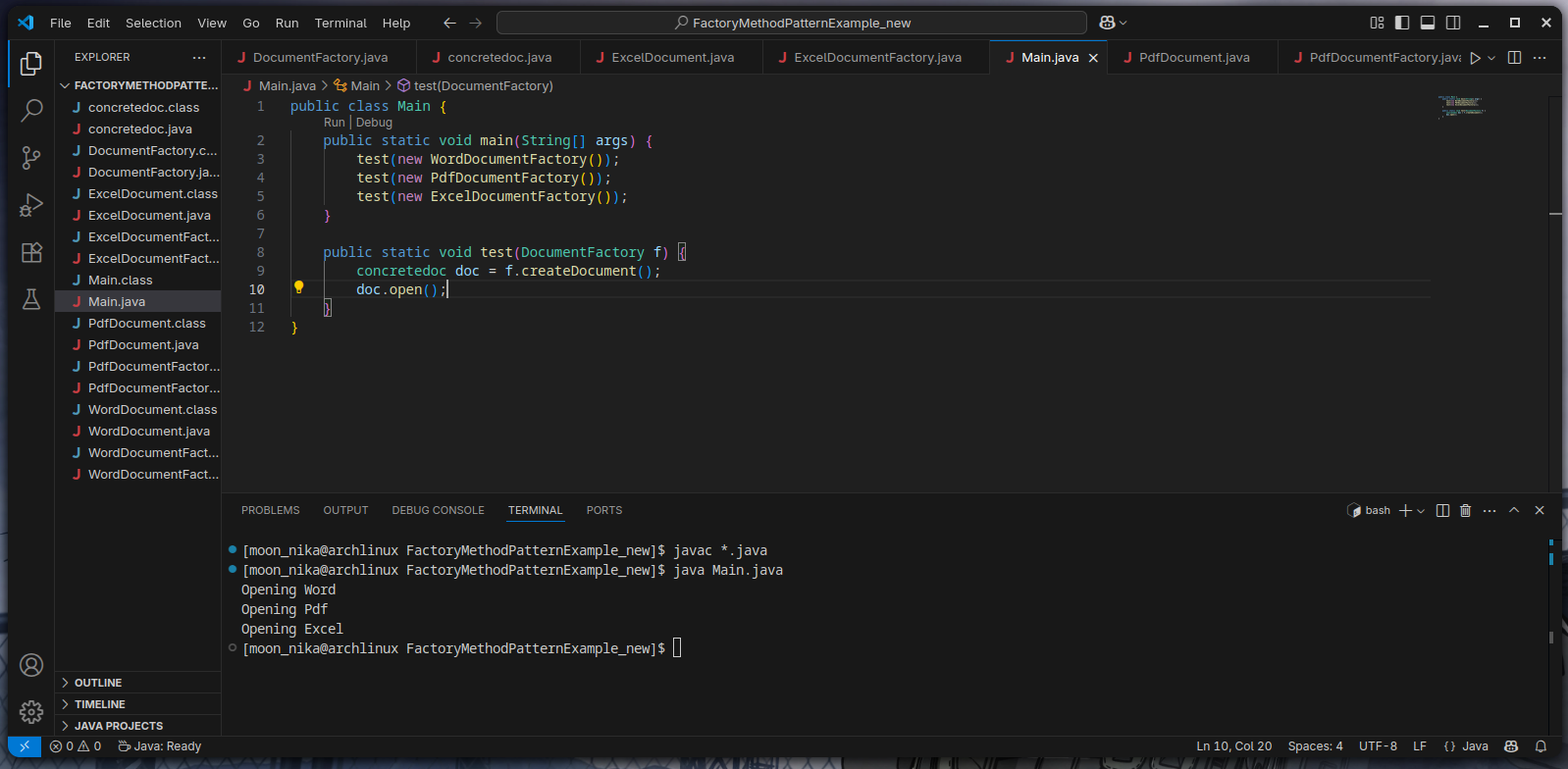
public concretedoc createDocument(){

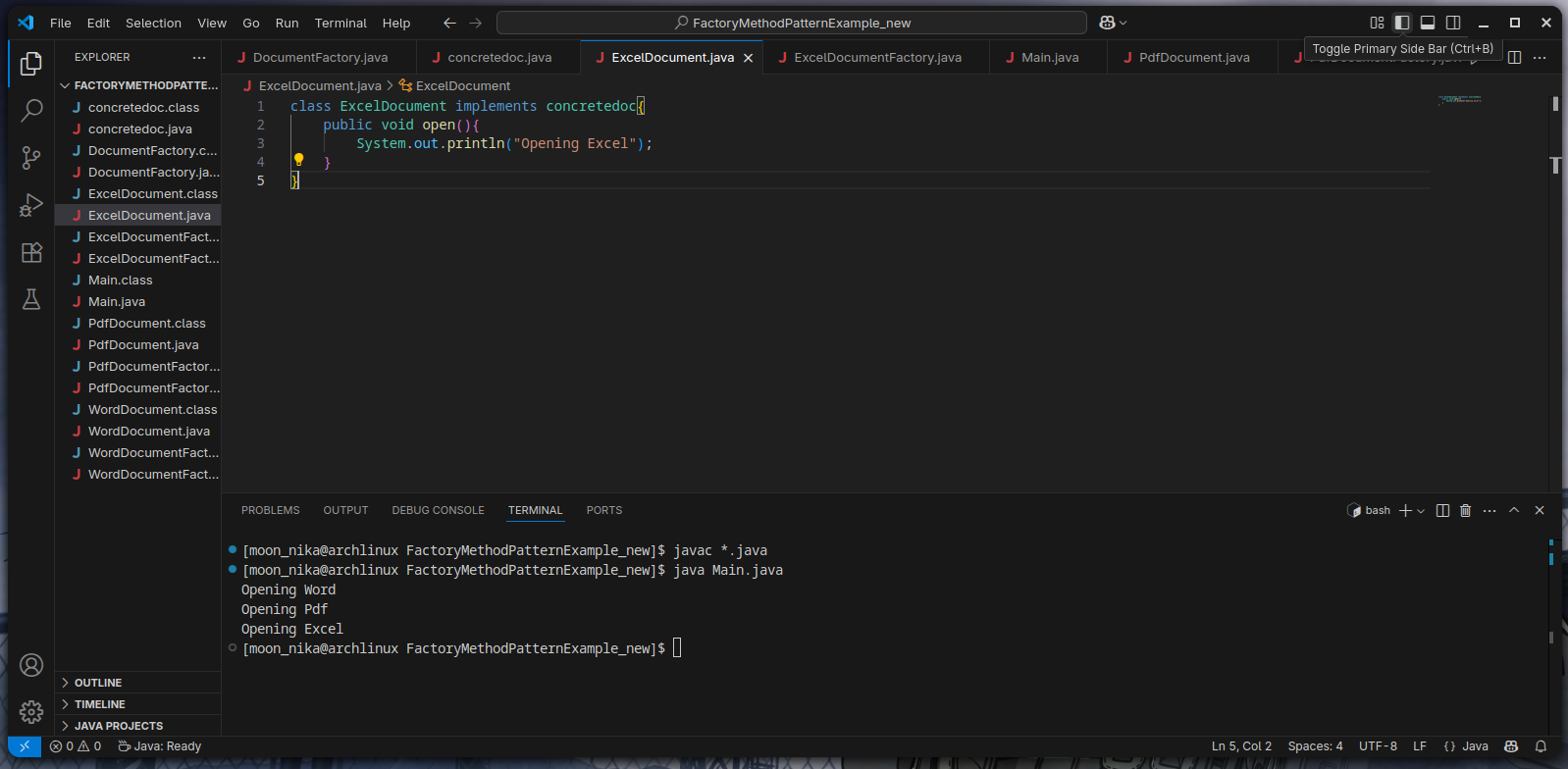
return new WordDocument();

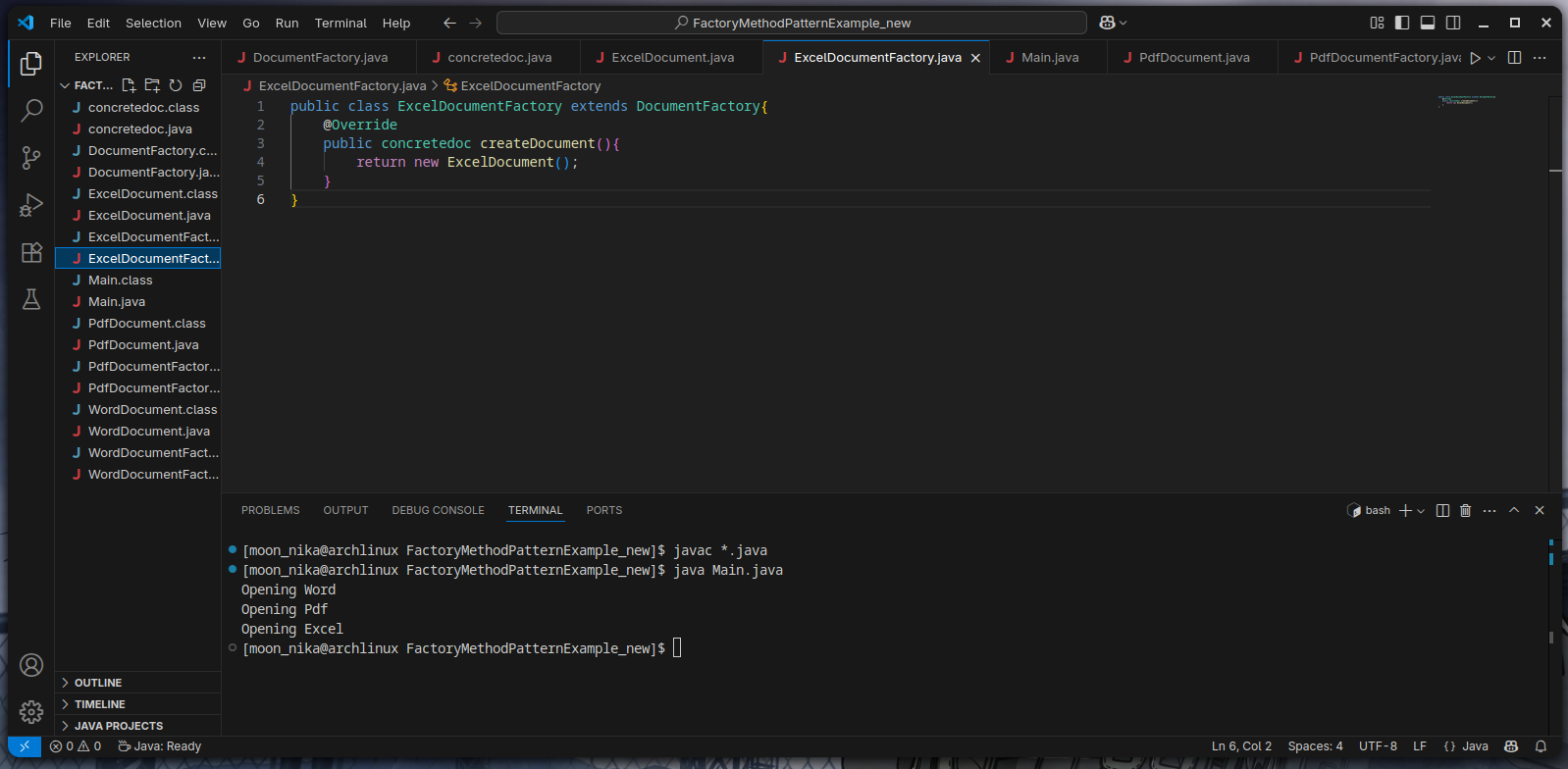
}

}

**Output:**





****