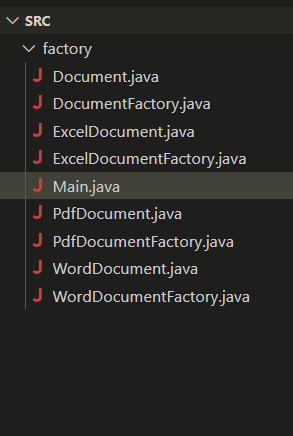
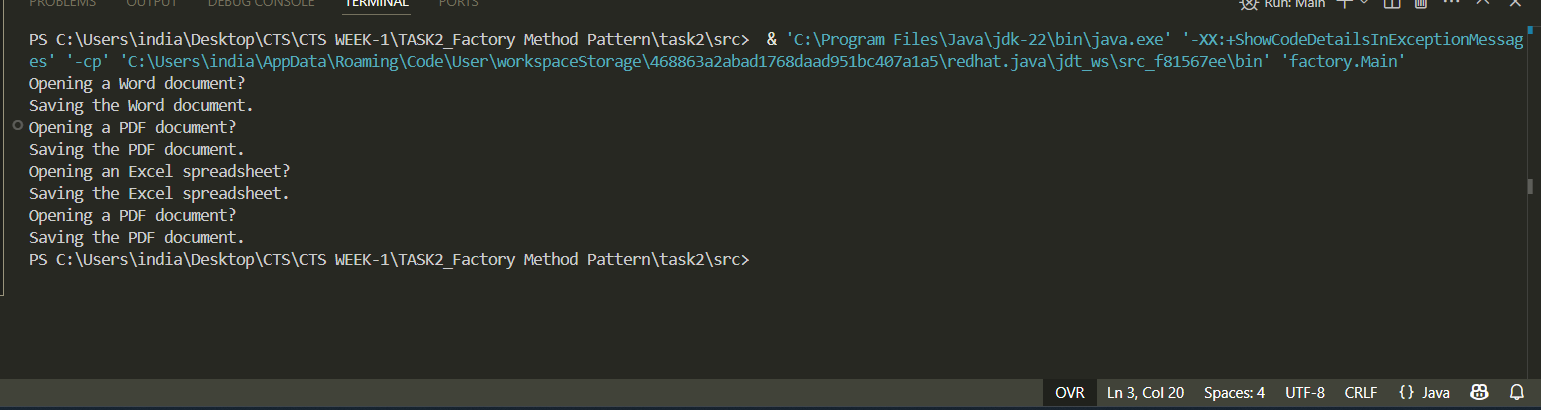
2200030216 **Exercise 2: Implementing the Factory Method Pattern  
here in total we have to create 9 file**

**Open you vs code  
create a java project and name your file  
I named it as the task-2  
in this create a “factory” folder inside the “src” folder**

**Now we can do our operations**You are developing a document management system that needs to create different types of documents

****

**Output  
**

Main.java  
package factory;

public class Main {

    public static *void* main(String[] *args*) {

        DocumentFactory wordFactory  = new WordDocumentFactory();

        DocumentFactory pdfFactory   = new PdfDocumentFactory();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document w = wordFactory.createDocument();

        Document p = pdfFactory.createDocument();

        Document e = excelFactory.createDocument();

        w.open(); w.save();

        p.open(); p.save();

        e.open(); e.save();

        Document doc2 = DocumentFactory.of("pdf").createDocument();

        doc2.open();

        doc2.save();

    }

}

Document

package factory;

public interface Document {

*void* open();

*void* save();

}

Documentfactory  
package factory;

public abstract class DocumentFactory {

    public abstract Document createDocument();

    /\* Optional helper that clients can call instead of “new ConcreteFactory()” \*/

    public static DocumentFactory of(String *type*) {

        return switch (*type*.toLowerCase()) {

            case "word"   *->* new WordDocumentFactory();

            case "pdf"    *->* new PdfDocumentFactory();

            case "excel"  *->* new ExcelDocumentFactory();

            default       *->* throw new IllegalArgumentException("Unknown type: " + *type*);

        };

    }

}

Excel.java  
package factory;

public class ExcelDocument implements Document {

    @*Override*

    public *void* open() { System.out.println("Opening an Excel spreadsheet…"); }

    @*Override*

    public *void* save() { System.out.println("Saving the Excel spreadsheet."); }

}

Exceldocument factory  
  
package factory;

public class ExcelDocumentFactory extends DocumentFactory {

    @*Override*

    public Document createDocument() { return new ExcelDocument(); }

}

Pdfdocument  
package factory;

public class PdfDocument implements Document {

    @*Override*

    public *void* open() { System.out.println("Opening a PDF document…"); }

    @*Override*

    public *void* save() { System.out.println("Saving the PDF document."); }

}

Pdfdocument factory  
  
package factory;

public class PdfDocumentFactory extends DocumentFactory {

    @*Override*

    public Document createDocument() { return new PdfDocument(); }

}