WEEK1\_DesignPrinciplesAndPatterns\_Hands0n  
  
**Exercise 1: Implementing the Singleton Pattern**

CODE:  
  
class Logger

{

    public Logger()

    {

        System .out.println("Instance created");

    }

    private static class LoggerHelper

    {

        public static final Logger INSTANCE=new Logger();

    }

    public static Logger getInstance()

    {

        return LoggerHelper.INSTANCE;

    }

}

class Main

{

    public static void main(String[] args)

    {

        Logger l1=Logger.getInstance();

        System.out.println("First instance");

        Logger l2=Logger.getInstance();

        System.out.println("Second instance");

        if(l1==l2)

        {

            System.out.println("Singleton satisfied");

        }

        else{

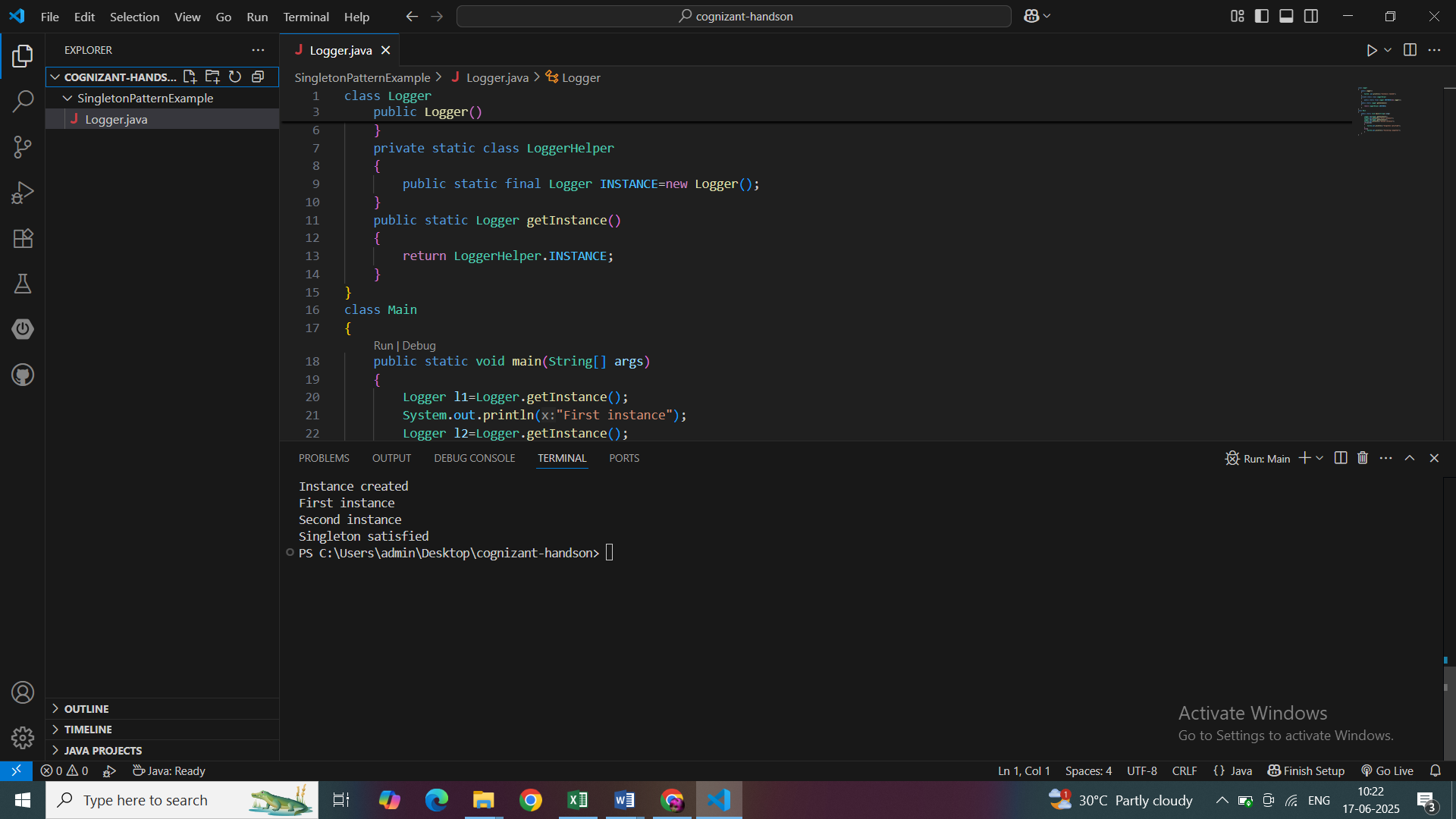
            System.out.println("Violating singleton");

        }

    }

}

OUTPUT:



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

**CODE:**package FactoryMethodPatternExample;

public class Factory {

    public static void main(String[] args)

    {

        DocFactory df=new WordFactory();

        Document d=df.createDoc();

        System.out.println(d.create());

        DocFactory df1=new PdfFactory();

        Document d1=df1.createDoc();

        System.out.println(d1.create());

        DocFactory df2=new ExcelFactory();

        Document d2=df2.createDoc();

        System.out.println(d2.create());

    }

}

interface Document{

    String create();

}

class Word implements Document{

    public String create()

    {

        return "WordDocument";

    }

}

class Pdf implements Document{

    public String create()

    {

        return "PdfDocument";

    }

}

class Excel implements Document{

    public String create()

    {

        return "ExcelDocument";

    }

}

abstract class DocFactory

{

    abstract Document createDoc();

}

class WordFactory extends DocFactory{

    public Document createDoc()

    {

        return new Word();

    }

}

class PdfFactory extends DocFactory{

    public Document createDoc(){

        return new Pdf();

    }

}

class ExcelFactory extends DocFactory{

    public Document createDoc()

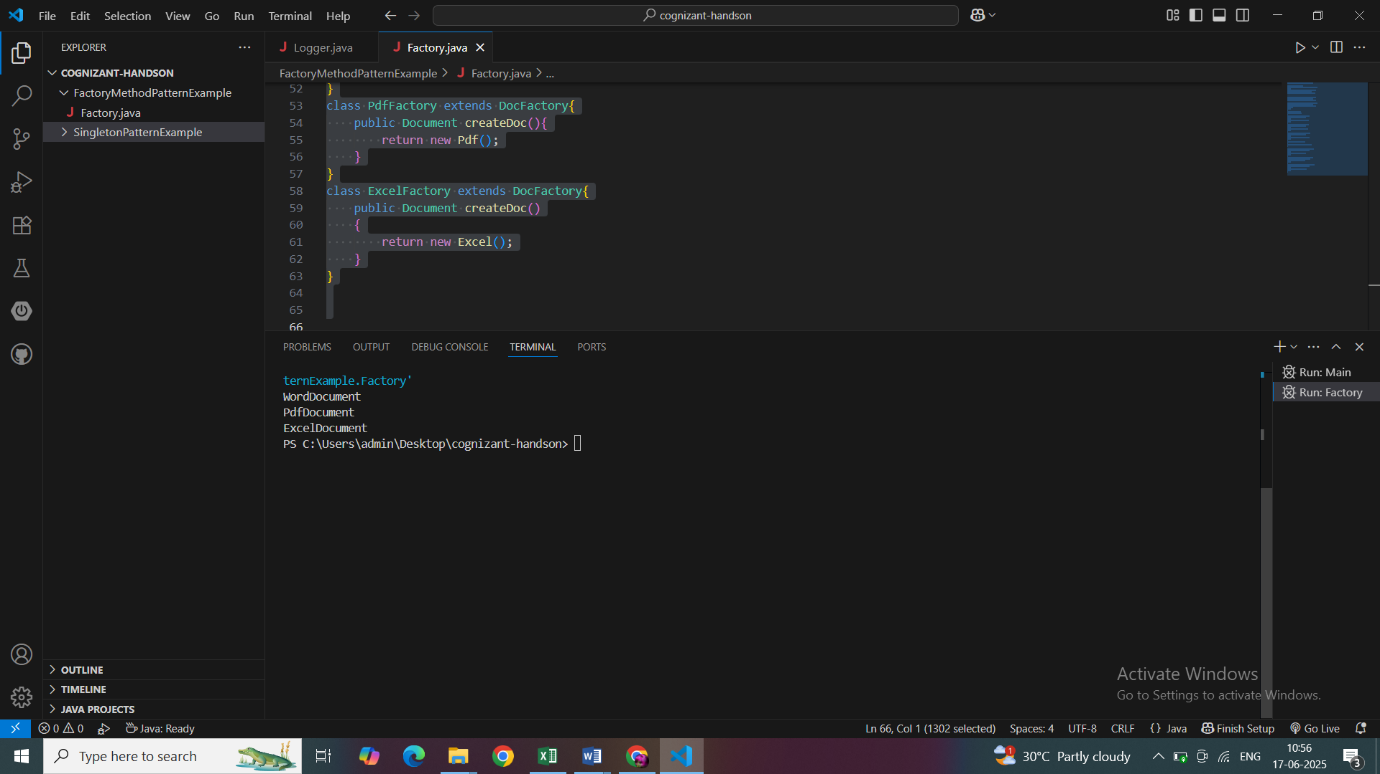
    {

        return new Excel();

    }

}

**OUTPUT:**

****