Design Pattern and Principles

Exercise 1 Code

class Logger{

private static Logger instance;

private Logger(){}

public static Logger getInstance(){

if(instance == null){

instance = new Logger();

}

return instance;

}

}

public class Exercise1 {

public static void main(String[] args){

Logger log = Logger.getInstance();

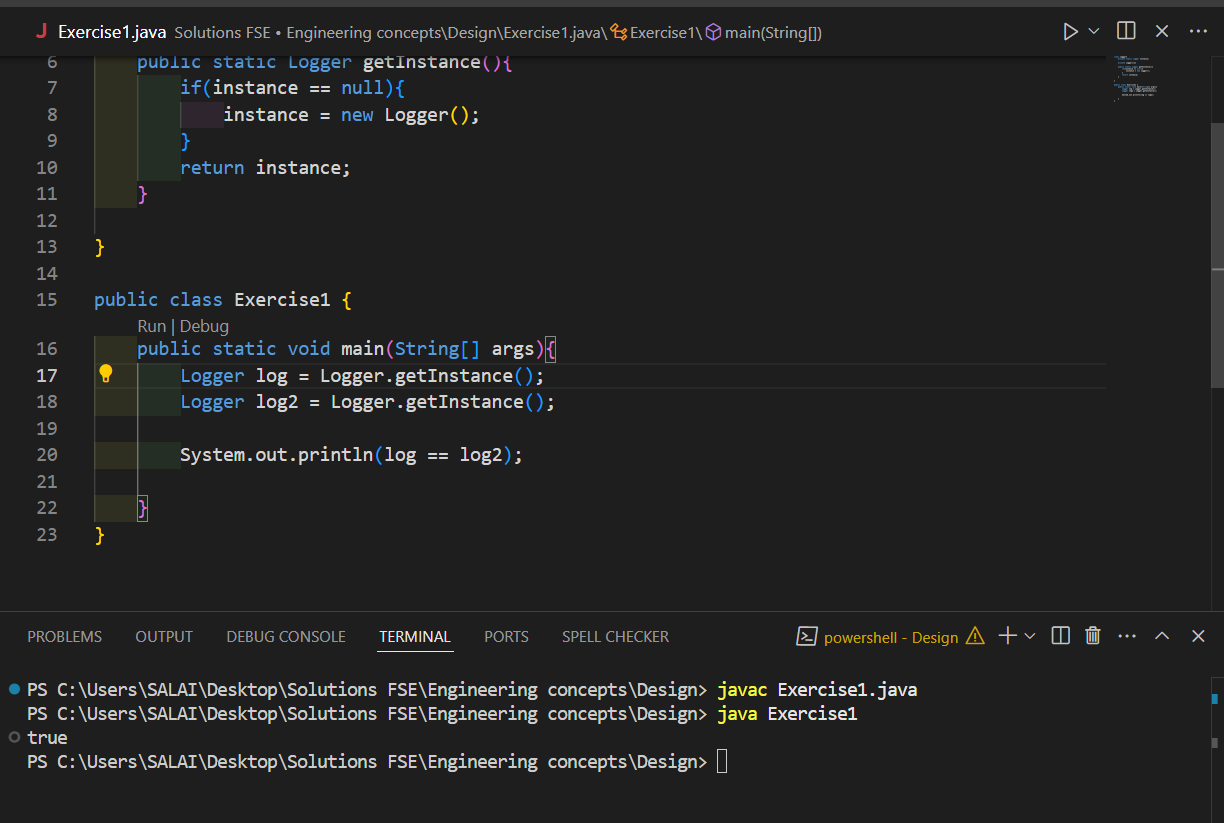
Logger log2 = Logger.getInstance();

System.out.println(log == log2);

}

}

Output



Exercise 2 Code

abstract class document{

    void open(){};

}

class wordDocument extends document{

    void open(){

        System.out.println("Opening a word doc");

    }

}

class pdfDocument extends document{

    void open(){

        System.out.println("Opening a pdf doc");

    }

}

class excelDocument extends document{

    void open(){

        System.out.println("Opening an excel doc");

    }

}

abstract class documentFactory{

    public abstract document create();

}

class wordDocumentFactory extends documentFactory{

    public document create(){

        return new wordDocument();

    }

}

class pdfDocumentFactory extends documentFactory{

    public document create(){

        return new pdfDocument();

    }

}

class excelDocumentFactory extends documentFactory{

 public document create(){

        return new excelDocument();

    }

}

public class Exercise2{

    public static void main(String[] args){

        documentFactory wordFactory = new wordDocumentFactory();

        document word = wordFactory.create();

        documentFactory pdfFactory = new pdfDocumentFactory();

        document pdf = pdfFactory.create();

        documentFactory excelFactory = new excelDocumentFactory();

        document excel = excelFactory.create();

        word.open();

        pdf.open();

        excel.open();

    }

}

Output

