**SKILLS : DESIGN PATTERNS AND PRINCIPLES**

**Qn 1 : Implementing the Singleton Pattern:**

**Code :**

**SingletonPatternExample.java**

class Logger {

    private static Logger instance;

    private Logger() {

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

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

        System.out.println("[LOGGER]: " + message);

    }

}

public class SingletonPatternExample {

    public static void main(String[] args) {

        Logger log1 = Logger.getInstance();

        log1.log("First log Instance");

        Logger log2 = Logger.getInstance();

        log2.log("Second log Instance");

        if (log1 == log2) {

            System.out.println("Both have same instance.");

        } else {

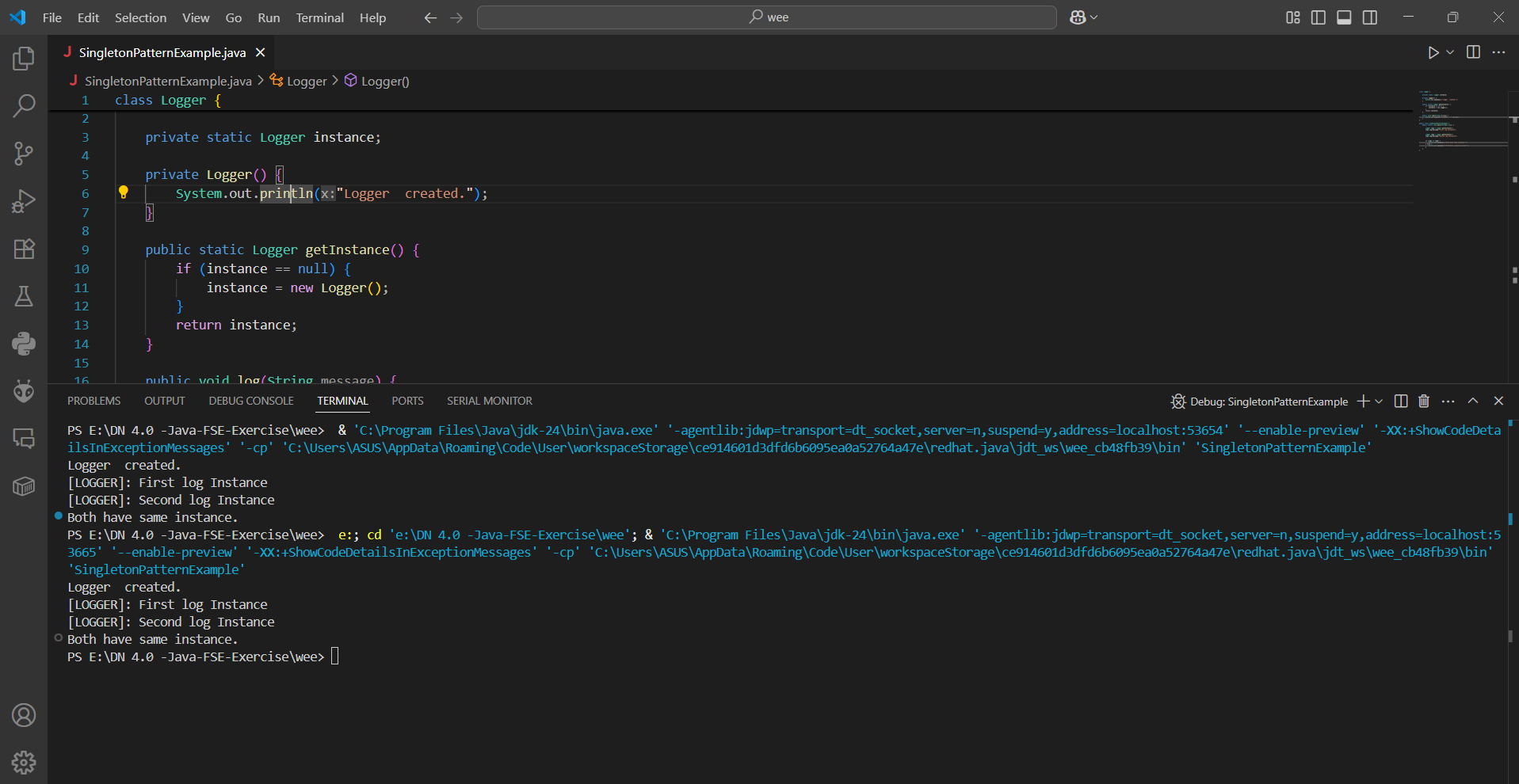
            System.out.println("Different instances exist!");

        }

    }

}

**Output:**

****

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

**Code:**

**FactoryMethodPatternExample.java**

interface Doc {

    void open();

}

class WordDocument implements Doc {

    public void open() {

        System.out.println("Word Document is now open.");

    }

}

class PdfDocument implements Doc {

    public void open() {

        System.out.println("PDF Document is opened.");

    }

}

class ExcelDocument implements Doc {

    public void open() {

        System.out.println("Excel Document is opened.");

    }

}

abstract class DocumentFactory {

    public abstract Doc createDocument();

}

class WordDocumentFactory extends DocumentFactory {

    public Doc createDocument() {

        return new WordDocument();

    }

}

class PdfDocumentFactory extends DocumentFactory {

    public Doc createDocument() {

        return new PdfDocument();

    }

}

class ExcelDocumentFactory extends DocumentFactory {

    public Doc createDocument() {

        return new ExcelDocument();

    }

}

public class FactoryMethodPatternExample {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Doc wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Doc pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

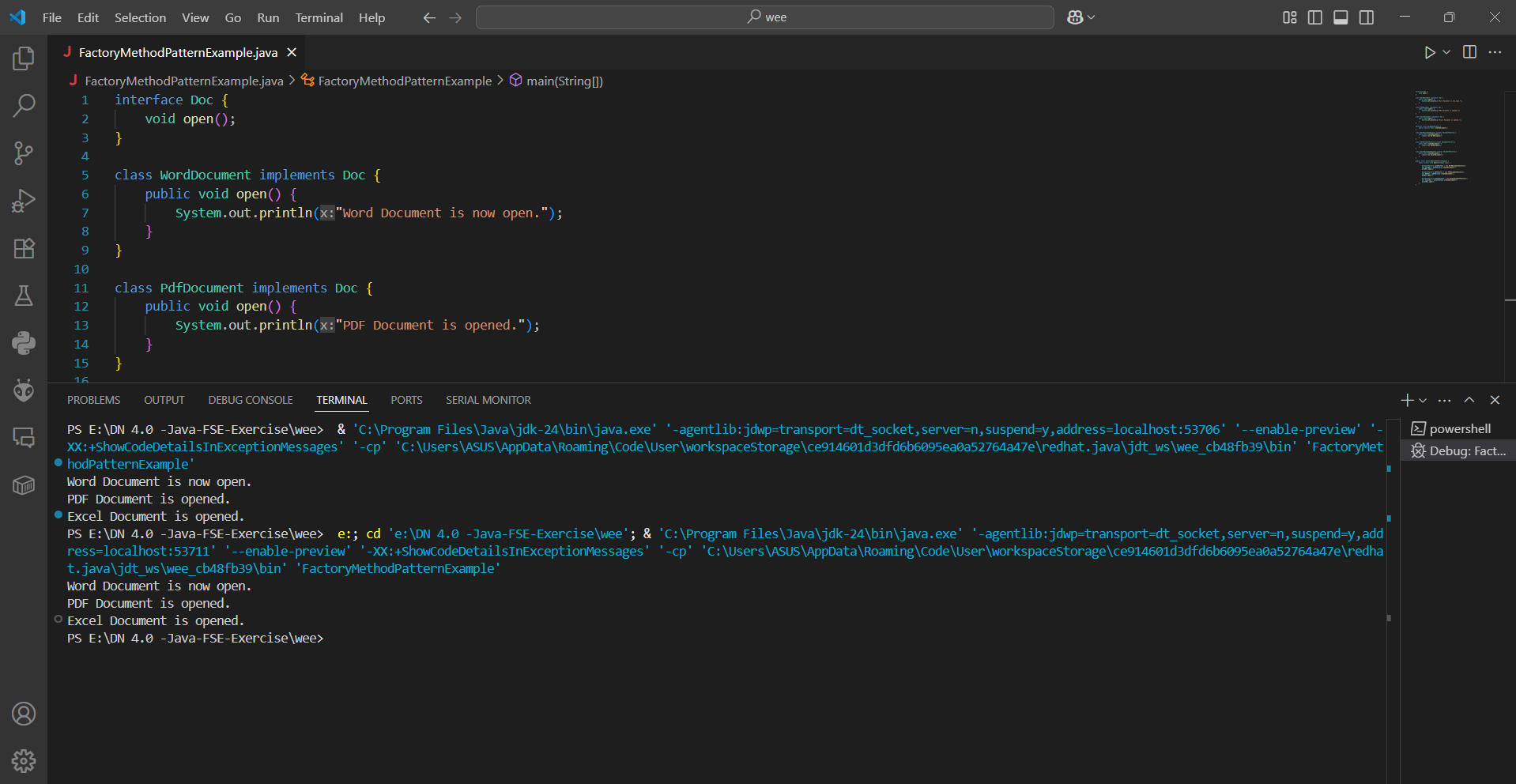
        Doc excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

**Output:**

****