**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**CODE**

class Logger {

    private static Logger instance;

    private Logger() {

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

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

        System.out.println(message);

    }

}

//our main function

public class SingletonPatternExample {

    public static void main(String[] args) {

        // Getting the singleton instance of Logger

        Logger logger1 = Logger.getInstance();

        logger1.log("First message");

        Logger logger2 = Logger.getInstance();

        logger2.log("Second message");

        // Verifying both references point to the same object

        if (logger1 == logger2) {

            System.out.println("logger1 and logger2 are the same instance.");

        }

else {

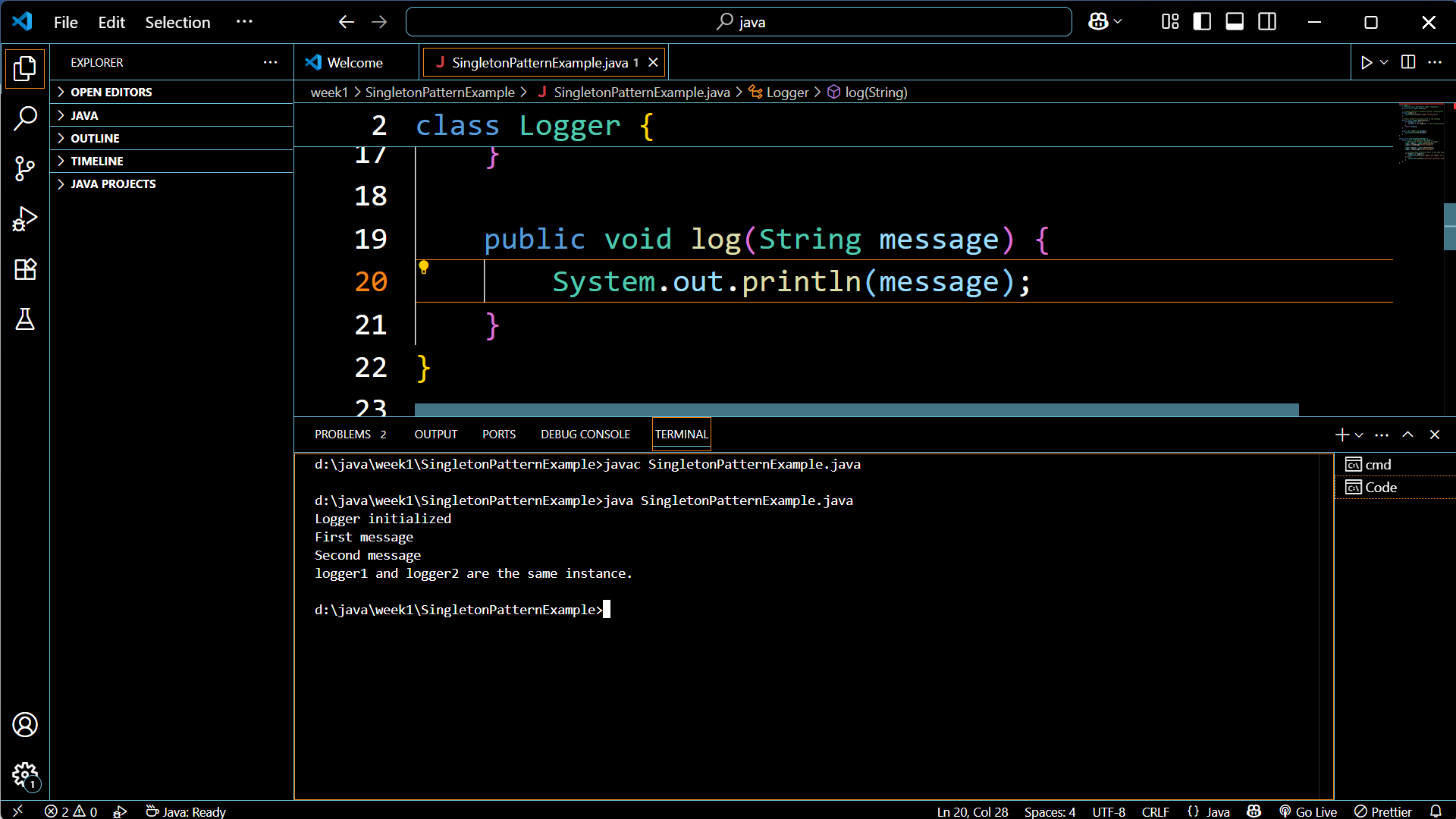
            System.out.println("Different instances! Singleton pattern failed.");

        }

    }

}

**OUTPUT**



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

**CODE**

interface Document {

    void open();

}

class WordDocument implements Document {

    @Override

    public void open() {

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

    }

}

class PdfDocument implements Document {

    @Override

    public void open() {

        System.out.println("Opening a PDF document.");

    }

}

class ExcelDocument implements Document {

    @Override

    public void open() {

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

    }

}

abstract class DocumentFactory {

    public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new WordDocument();

    }

}

class PdfDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new PdfDocument();

    }

}

class ExcelDocumentFactory extends DocumentFactory {

    @Override

    public Document createDocument() {

        return new ExcelDocument();

    }

}

//main file

public class FactoryMethodPatternExample {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

**OUTPUT**

