Design Patterns and Principles

Exercise 1: Implementing the Singleton Pattern

**Program code:**

package mypack;

public class SingletonPatternExample {

static class Logger {

        private static Logger instance;

        private Logger() {

            System.out.println("Logger Instance Created");

        }

        public static Logger getInstance() {

            if (instance == null) {

                instance = new Logger();

            }

            return instance;

        }

public void log(String message) {

            System.out.println("Log: " + message);

        }

    }

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

  logger1.log("hello logger!");

        logger2.log("i am here!");

if (logger1 == logger2) {

            System.out.println("Both logger instances are the same.")} else {

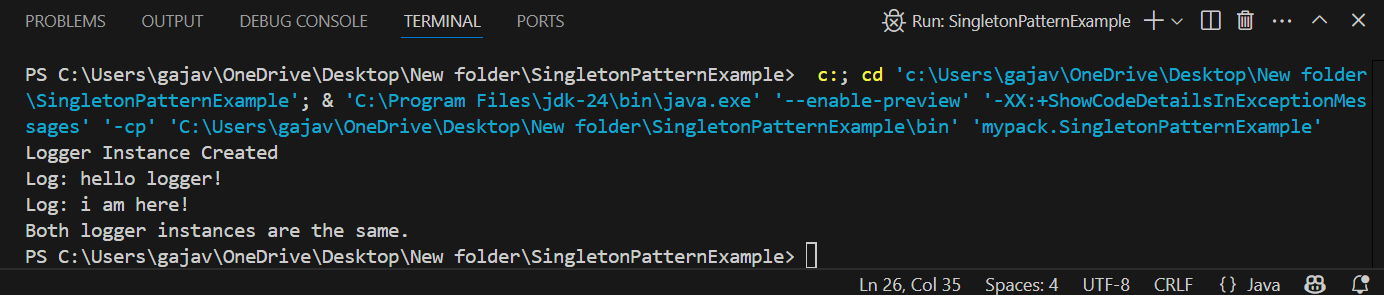
            System.out.println("Logger instances are different.");

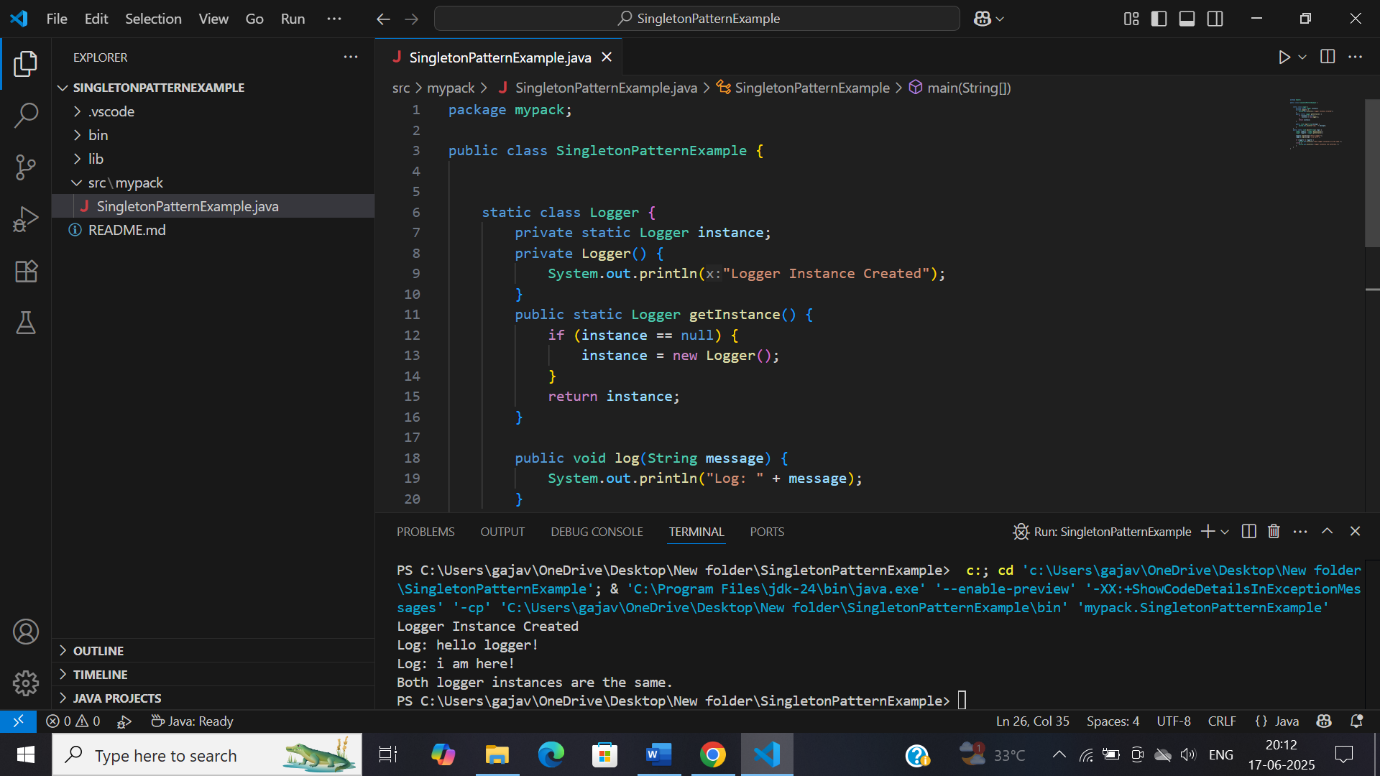
        }

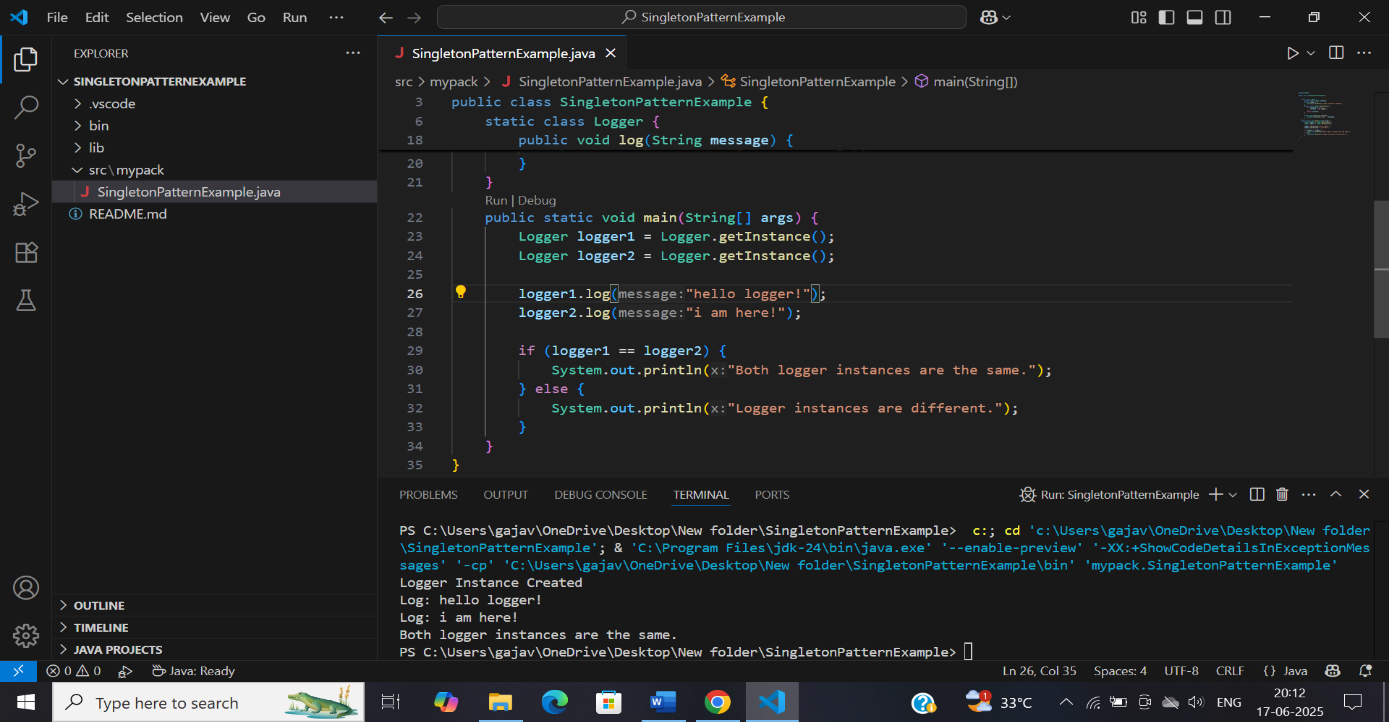
}

}

**OUTPUT:**







Exercise 2: Implementing the Factory Method Pattern

Program code:

package mypack;

public class FactoryMethodPatternExample {

    interface Document {

        void open();

    }

    static class WordDocument implements Document {

        public void open() {

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

        }

    }

    static class PdfDocument implements Document {

        public void open() {

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

        }

    }

    static class ExcelDocument implements Document {

        public void open() {

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

        }

    }

    abstract static class DocumentFactory {

        public abstract Document createDocument();

    }

    static class WordDocumentFactory extends DocumentFactory {

        public Document createDocument() {

            return new WordDocument();

        }

    }

    static class PdfDocumentFactory extends DocumentFactory {

        public Document createDocument() {

            return new PdfDocument();

        }

    }

    static class ExcelDocumentFactory extends DocumentFactory {

        public Document createDocument() {

            return new ExcelDocument();

        }

    }

    public static void main(String[] args) {

        DocumentFactory wf= new WordDocumentFactory();

        Document wordDoc = wf.createDocument();

        wordDoc.open();

        DocumentFactory pf = new PdfDocumentFactory();

        Document pdfDoc = pf.createDocument();

        pdfDoc.open();

        DocumentFactory ef = new ExcelDocumentFactory();

        Document excelDoc = ef.createDocument();

        excelDoc.open();

    }

}

**OUTPUT:**

