# Design Principles & Patterns

Week 1

Exercise 1 :Implementing the Singleton Pattern

Code:

Logger.java

public 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);

    }

}

LoggerTest.java

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("First log message");

        logger2.log("Second log message");

        if (logger1 == logger2) {

            System.out.println("Both logger instances are the same (Singleton works).");

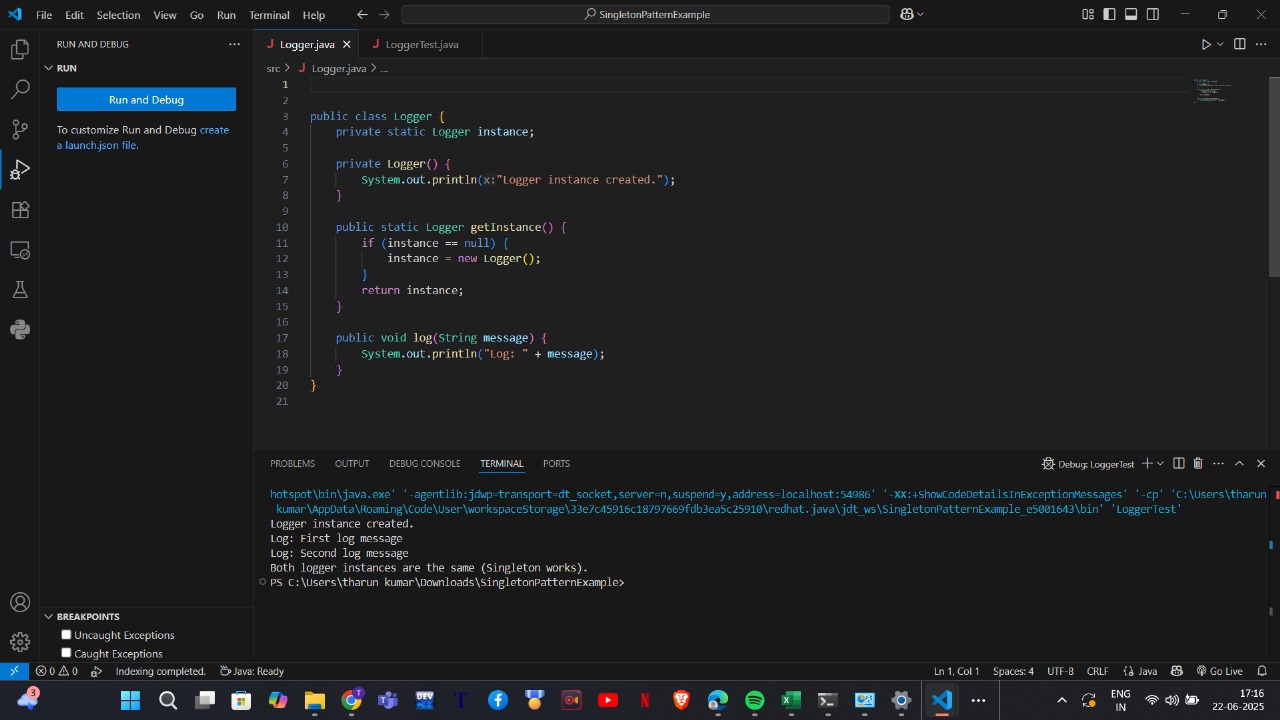
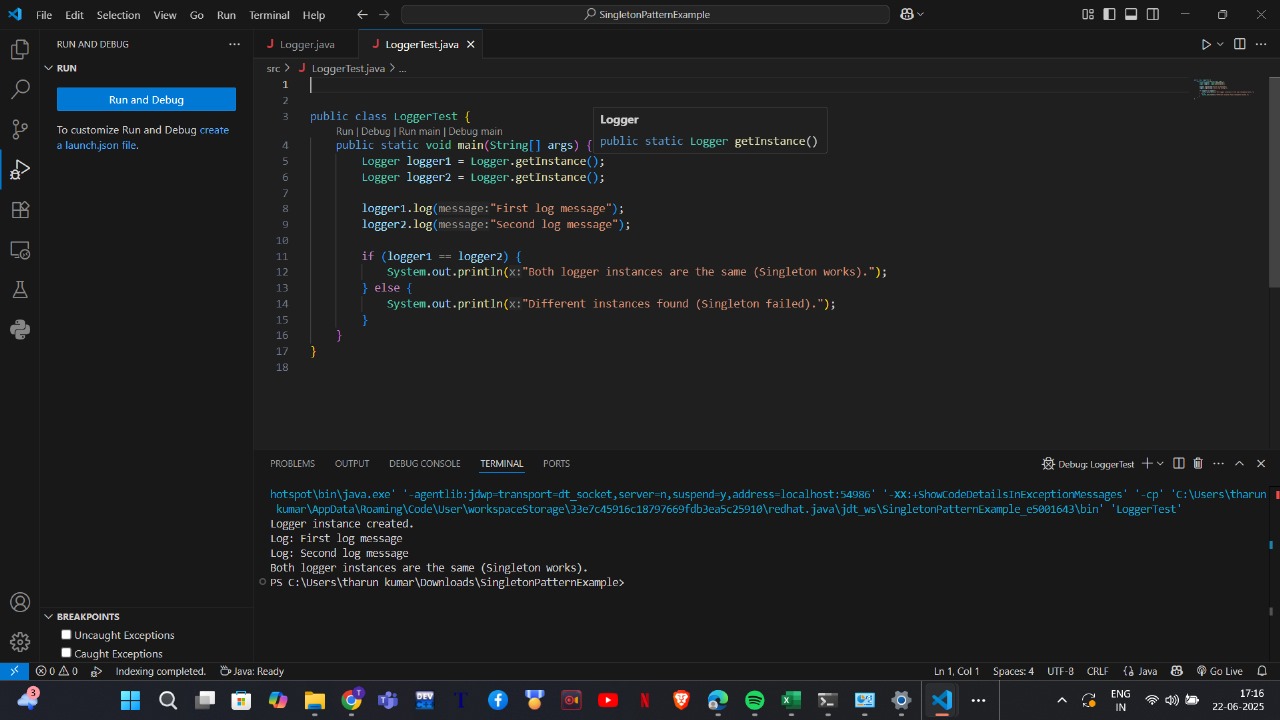
        } else {

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

        }

    }

}



Exercise 2 :FactoryMethodPatternExample

DocumentFactory.java

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

ExcelDocument.java

public class ExcelDocument implements Document {

    public void open() {

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

    }

}

WordDocumentFactory.java

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

PdfDocumentFactory.java

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

ExcelDocumentFactory.java

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

FactoryTest.java

public class FactoryTest {

    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();

    }

}

Document.java

public interface Document {

    void open();

}

WordDocument.java

public class WordDocument implements Document {

    public void open() {

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

    }

}

pdfDocument.java

public class PdfDocument implements Document {

    public void open() {

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

    }

}

ExcelDocument.java

public class PdfDocument implements Document {

    public void open() {

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

    }

}

