WEEK 1 – Design Patterns and Principles

HANDS-ON 1 :- Implementing the Singleton Pattern

**CODE OF SINGLETON PATTERN :- Logger.java**

package singleton;

public 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("Log: " + message);

    }

}

**Main.java**

package singleton;

public class Main {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("This is the first log message.");

        Logger logger2 = Logger.getInstance();

        logger2.log("This is the second log message.");

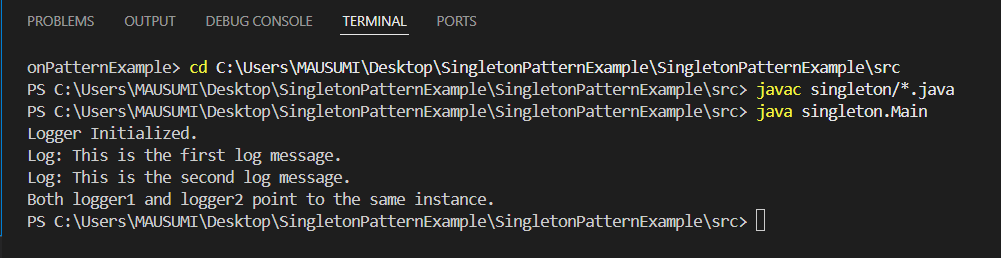
        if (logger1 == logger2) {

            System.out.println("Both logger1 and logger2 point to the same instance.");

        } else {

            System.out.println("Different Logger instances.");

        }

**OUTPUT OF SINGLETON PATTERN**

HANDS-ON 2 :- Implementing the Factory Method Pattern

**CODE OF FACTORY METHOD PATTERN:-**

**public class FactoryMethodPatternExample {**

**interface Document {**

**void open();**

**}**

**static class WordDocument implements Document {**

**public void open() {**

**System.out.println("Opening Word Document...");**

**}**

**}**

**static class PdfDocument implements Document {**

**public void open() {**

**System.out.println("Opening PDF Document...");**

**}**

**}**

**static class ExcelDocument implements Document {**

**public void open() {**

**System.out.println("Opening 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();**

**}**

**}**

**// Main method**

**public static void main(String[] args) {**

**DocumentFactory wordFactory = new WordDocumentFactory();**

**Document word = wordFactory.createDocument();**

**word.open();**

**DocumentFactory pdfFactory = new PdfDocumentFactory();**

**Document pdf = pdfFactory.createDocument();**

**pdf.open();**

**DocumentFactory excelFactory = new ExcelDocumentFactory();**

**Document excel = excelFactory.createDocument();**

**excel.open();**

**}**

**}**

**OUTPUT OF FACTORY METHOD PATTERN**

