**Week-1 – Engineering Concepts**

**Module - 1 Design Pattens & Principles**

**Exercise 1: Implementing the Singleton Pattern**

**1. Logger.java:**

public class Logger {

// class names Logger with a private static instance of itself

private static Logger loggerInstance;

// private constructor

private Logger() {

}

// public static method to get instance of the class

public static Logger getInstance() {

if (loggerInstance == null) {

synchronized (Logger.class) {

if (loggerInstance == null) {

loggerInstance = new Logger();

}

}

}

return loggerInstance;

}

// sample logging method

public void log(String msg) {

System.out.println("LOG: " + msg);

}

}

**2. SingletonTest.java:**

public class SingletonTest {

public static void main(String[] args) {

// get instances of logger

Logger log1 = Logger.getInstance();

Logger log2 = Logger.getInstance();

// log messages using references

log1.log("This is test message 1");

log2.log("This is test message 2");

// check if they point to the same instance

if (log1 == log2) {

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

} else {

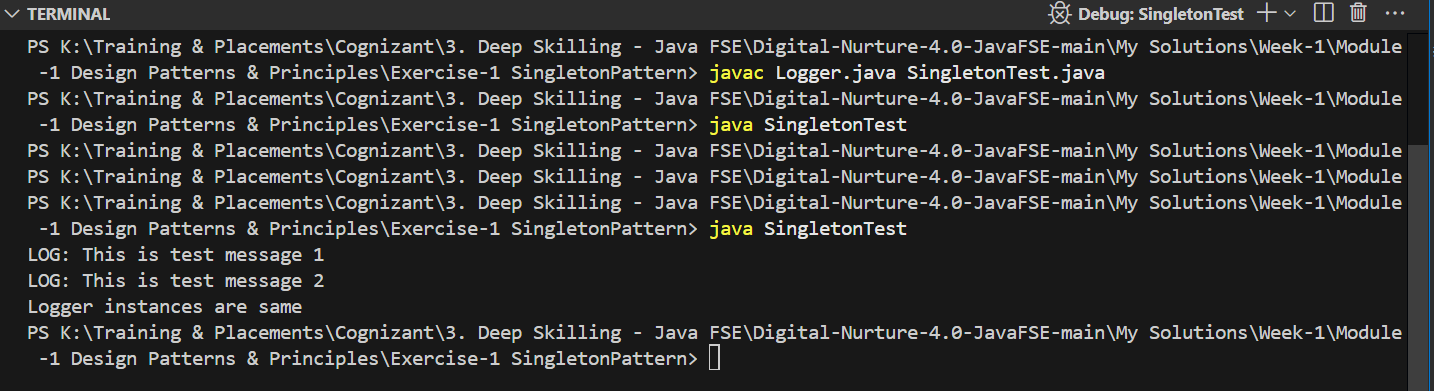
System.out.println("Logger instances are not same");

}

}

}

Output –



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

**1. Document.java:**

public interface Document {

    void open();

    void close();

}

**2. DocumentFactory.java:**

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**3. ExcelDocument.java:**

public class ExcelDocument implements Document {

public void open() {

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

}

public void close() {

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

}

}

**4. ExcelDocumentFactory.java:**

public class ExcelDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**5. FactoryTest.java:**

public class FactoryTest {

    public static void main(String[] args) {

        // Create Word document

        DocumentFactory word = new WordDocumentFactory();

        Document wordDoc = word.createDocument();

        wordDoc.open();

        wordDoc.close();

        // Create PDF document

        DocumentFactory pdf = new PdfDocumentFactory();

        Document pdfDoc = pdf.createDocument();

        pdfDoc.open();

        pdfDoc.close();

        // Create Excel document

        DocumentFactory excel = new ExcelDocumentFactory();

        Document excelDoc = excel.createDocument();

        excelDoc.open();

        excelDoc.close();

    }

}

**6. PdfDocument.java:**

public class PdfDocument implements Document {

    public void open() {

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

    }

    public void close() {

        System.out.println("Closing PDF Document.");

    }

}

**7. PdfDocumentFactory.java:**

public class PdfDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**8. WordDocument.java:**

public class WordDocument implements Document {

    public void open() {

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

    }

    public void close() {

        System.out.println("Closing Word Document");

    }

}

**9. WordDocumentFactory.java:**

public class WordDocumentFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

Output:

