**Week 1 – Hands-on : Design Pattern and Principles**

Exercise 1 : Implementing the Singleton Pattern

**File**: Main.java

import com.SingletonClass.Logger;

class Main {

    public static void main(String[] args) {

        Logger log1 = Logger.getInstance();

        log1.log("First time");

        Logger log2 = Logger.getInstance();

        log2.log("Second time");

        if (log1 == log2) {

            System.out.println("Both are of same instance, Singleton class verified");

        } else {

            System.out.println("Both are different instance, Singleton class failed");

        }

    }

}

**File:** Logger.java

package com.SingletonClass;

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Initialized..");

    }

    public static synchronized Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

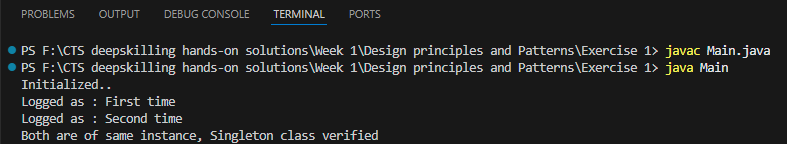
    public void log(String msg) {

        System.out.println("Logged as : " + msg);

    }

}

OUTPUT :



Exercise 2 : Implementing the Factory Method Pattern

**File** : Main.java

import com.Factory.Document;

import com.Factory.DocumentFactory;

class Main {

    public static void main(String[] args) {

        DocumentFactory df = new DocumentFactory();

        Document doc1 = df.create("PDF");

        doc1.createDocument();

        Document doc2 = df.create("Excel");

        doc2.createDocument();

        Document doc3 = df.create("word");

        doc3.createDocument();

        Document doc4 = df.create("Csv");

        doc4.createDocument();

    }

}

**File** : Document.java

package com.Factory;

public interface Document {

    public void createDocument();

}

**File** : PdfDoc.java

package com.Factory;

public class PdfDoc implements Document {

    public void createDocument() {

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

    }

}

**File** : WordDoc.java

package com.Factory;

public class WordDoc implements Document {

    public void createDocument() {

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

    }

}

**File** : ExcelDoc.java

package com.Factory;

public class ExcelDoc implements Document {

    public void createDocument() {

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

    }

}

**File** : Error.java

package com.Factory;

public class Error implements Document {

    public void createDocument() {

        System.out.println("Unsupported document..");

    }

}

**File** : DocumentFactory.java

package com.Factory;

public class DocumentFactory {

    public Document create(String file) {

        String documentType = file.trim().toLowerCase();

        switch (documentType) {

            case "pdf":

                return new PdfDoc();

            case "excel":

                return new ExcelDoc();

            case "word":

                return new WordDoc();

            default:

                return new Error();

        }

    }

}

OUTPUT:

