**Week-1**

**Design principles & Patterns**

**Exercise 1: Implementing the Singleton Pattern**

Logger.java

package com.example.singleton;

public class Logger {

private static Logger singleInstance;

private Logger() {

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

}

public static Logger getInstance() {

if (singleInstance == null) {

singleInstance = new Logger();

}

return singleInstance;

}

public void log(String message) {

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

}

}

Main.java

package com.example.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 logger instances are the same (Singleton confirmed).");

} else {

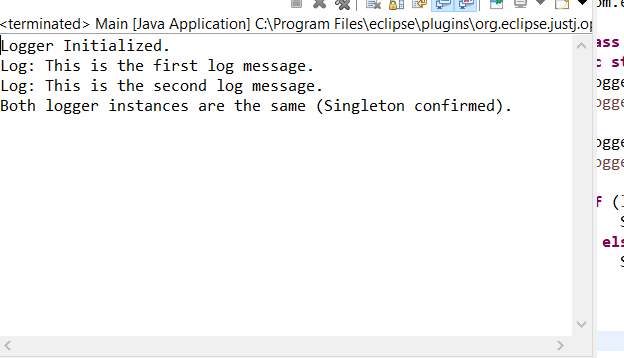
System.out.println("Logger instances are different (Singleton violated).");

}

}

}

**Output:**



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

Document.java

package com.example.factory;

public interface Document {

void open();

}

WordDocument.java

package com.example.factory;

public class WordDocument implements Document {

@Override

public void open() {

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

}

}

PdfDocument.java

package com.example.factory;

public class PdfDocument implements Document {

public void open() {

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

}

}

ExcelDocument.java

package com.example.factory;

public class ExcelDocument implements Document {

public void open() {

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

}

}

WordFactory.java

package com.example.factory;

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

PdfFactory.java

package com.example.factory;

public class PdfFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

ExcelFactory.java

package com.example.factory;

public class ExcelFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

DocumentFactory.java

package com.example.factory;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

Main.java

package com.example.factory;

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordFactory();

Document word = wordFactory.createDocument();

word.open();

DocumentFactory pdfFactory = new PdfFactory();

Document pdf = pdfFactory.createDocument();

pdf.open();

DocumentFactory excelFactory = new ExcelFactory();

Document excel = excelFactory.createDocument();

excel.open();

}

}

**OUTPUT:**

