**Digital Nurture 4.0 – Week 1**

**1.Design principles & Patterns**

**Exercise 1:** Implementing the Singleton Pattern

**Program:**

I created a project named SingletonPatternExample with a Logger class using the Singleton design pattern. The constructor is private, and a static getInstance() method ensures only one instance is created. I added a log() method to display messages. Then, I created a separate Main class to verify that the same Logger instance is used throughout the application**.**

**1. Logger.java – Singleton Class**

package singleton;

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);

}

}

**2.TestSingleton.java – To Test the Singleton Behavior**

package singleton;

public class TestSingleton {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log message");

Logger logger2 = Logger.getInstance();

logger2.log("Second log message");

if (logger1 == logger2) {

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

} else {

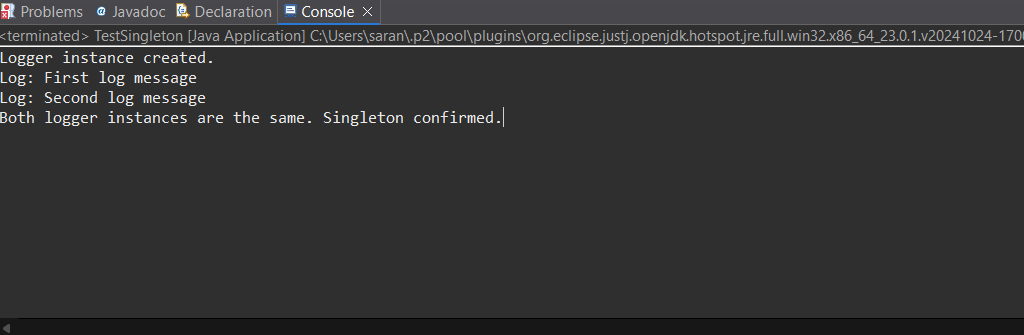
System.out.println("Logger instances are different. Singleton failed.");

}

}

}

**Output:**



Exercise 2: Implementing the Factory Method Pattern

**Program:**

I created a document system using the **Factory Method Pattern**.  
Each document type (Word, PDF, Excel) has its own class and factory.  
The main class uses these factories to create and open documents.

**Factorymethod.java**

package factorypattern;

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

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

}

}

class PdfDocument implements Document {

public void open() {

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

}

}

class ExcelDocument implements Document {

public void open() {

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

}

}

abstract class DocumentFactory {

public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

public class FactoryMethodDemo {

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();

}

}

**Output:**

