**Digital Nurture 4.0**

Week 1-Design Patterns and Principles

**Exercise 1: Implementing the Singleton Pattern**:

**Program:**

Implemented and tested a Logger class using the Singleton design pattern to ensure only one instance exists throughout the application. Verified consistent logging through a test class by confirming the same instance is used across the application.

**Logger.java**:

public class Logger {

private static Logger *instance*;

private Logger() {

System.***out***.println("Logger Instance created successfully");

}

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger();

}

return *instance*;

}

public void log(String message) {

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

}

}

**Main.java**:

package SingletonPatternExample;

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

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

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

if (logger1 == logger2) {

System.***out***.println("Same Instances!");

} else {

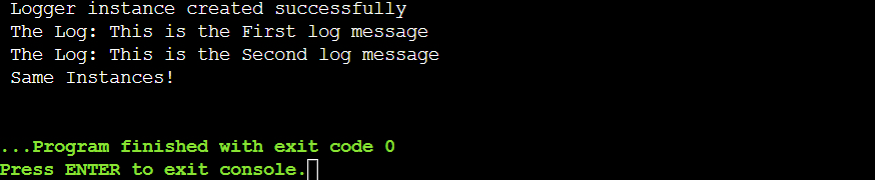
System.***out***.println("Different Instances!");

}

}

}

**Output**:



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

**Program:**

In my program, I created a class named FactoryMethodPatternExample and implemented the Factory Method Pattern within it. Including the interface, document types, and factory classes, were defined in a single Java file. The main method tests the creation of Word, PDF, and Excel documents through their respective factories.

**FactoryMethodPatternExample.java:**

public class FactoryMethodPatternExample {

interface Document {

void open();

}

static class WordDocument implements Document {

public void open() {

System.***out***.println("Displaying Word Document..!");

}

}

static class PdfDocument implements Document {

public void open() {

System.***out***.println("Displaying PDF Document..!");

}

}

static class ExcelDocument implements Document {

public void open() {

System.***out***.println("Displaying Excel Document..!");

}

}

abstract static class DocumentFactory {

abstract Document createDocument();

}

static class WordDocumentFactory extends DocumentFactory {

Document createDocument() {

return new WordDocument();

}

}

static class PdfDocumentFactory extends DocumentFactory {

Document createDocument() {

return new PdfDocument();

}

}

static class ExcelDocumentFactory extends DocumentFactory {

Document createDocument() {

return new ExcelDocument();

}

}

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:**

****