**Name:** Paarthivi D

**Superset ID:** 6410961

**DN 4.0 - Java FSE Mandatory Hands-on**

**WEEK – 1**

**Design Principles and Patterns**

**Exercise 1: Implementing the Singleton Pattern**

**Code:**

package singletonPatternExample;

public class SingletonPatternExample {

static class Logger {

// Step 1: Create a private static instance of Logger

private static Logger *singleInstance*;

// Step 2: Make the constructor private

private Logger() {

System.*out*.println("Logger instance created");

}

// Step 3: Provide a public static method to get the instance

public static Logger getInstance() {

if (*singleInstance* == null) {

*singleInstance* = new Logger();

}

return *singleInstance*;

}

// Logging method

public void log(String message) {

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

}

}

// Main method to test the Singleton

public static void main(String[] args) {

// Get instances

Logger logger1 = Logger.*getInstance*();

Logger logger2 = Logger.*getInstance*();

// Use the logger

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

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

// Test if both are the same instance

if (logger1 == logger2) {

System.*out*.println("Both logger1 and logger2 refer to the same instance.");

} else {

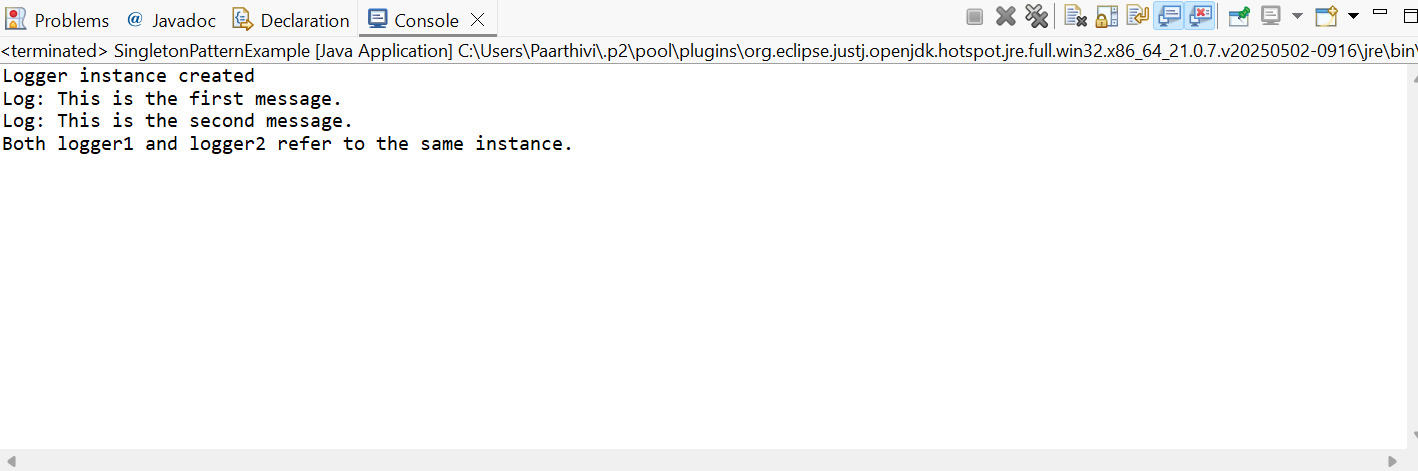
System.*out*.println("Different instances exist!");

}

}

}

**Output Screenshot:**



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

**Code:**

package factoryMethodPatternExample;

public class FactoryMethodDemo {

// Step 1: Document interface

interface Doc {

void open();

}

// Step 2: Concrete Document classes

static class WordDocument implements Doc {

public void open() {

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

}

}

static class PdfDocument implements Doc {

public void open() {

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

}

}

static class ExcelDocument implements Doc {

public void open() {

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

}

}

// Step 3: Abstract Factory

static abstract class DocumentFactory {

public abstract Doc createDocument();

}

// Step 4: Concrete Factories

static class WordDocumentFactory extends DocumentFactory {

public Doc createDocument() {

return new WordDocument();

}

}

static class PdfDocumentFactory extends DocumentFactory {

public Doc createDocument() {

return new PdfDocument();

}

}

static class ExcelDocumentFactory extends DocumentFactory {

public Doc createDocument() {

return new ExcelDocument();

}

}

// Step 5: Main method

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Doc wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Doc pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Doc excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output Screenshot:**

