**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

**You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.**

**Code for logger.java:**

package com.singleton.example;

public class logger {

private static logger *instance*;

private logger() {

System.***out***.println("logger initialized.");

}

public static logger getInstance() {

if (*instance* == null) {

*instance* = new logger(); // Lazy initialization

}

return *instance*;

}

public void log(String message) {

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

}

}

**Code for TestSingleton.java:**

package com.singleton.example;

public class TestSingleton {

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.");

// Check if both references point to the same object

if (logger1 == logger2) {

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

} else {

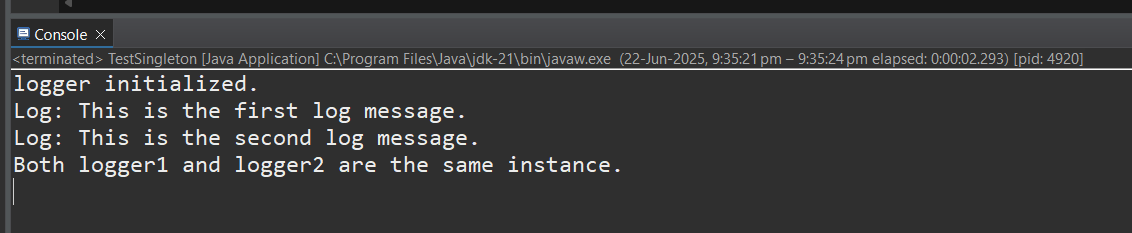
System.***out***.println("Different instances were created!");

}

}

}

**OUTPUT:**



**Exercise 2: Implementing the Factory Method Pattern Scenario: You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.**

**DocumentFactoryExample.java**

package com.factorymethod.example;

public class DocumentFactoryExample {

public static void main(String[] args) {

DocumentFactory factory;

factory = new WordDocumentFactory();

Document word = factory.createDocument();

word.open();

factory = new PdfDocumentFactory();

Document pdf = factory.createDocument();

pdf.open();

factory = new ExcelDocumentFactory();

Document excel = factory.createDocument();

excel.open();

}

}

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();

}

}

**OUTPUT:**

