**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:**

**LoggerClass:**

package designpattern;

public class Logger {

private static volatile Logger *inst*;

private Logger() {

System.*out*.println("Private Contructor");

}

public static Logger getInstance() {

if (*inst* == null) {

synchronized (Logger.class) {

if (*inst* == null) {

*inst* = new Logger();

}

}

}

return *inst*;

}

}

**TestClass:**

package designpattern;

public class LoggerTest {

public static void main(String[] args) {

Logger obj1 = Logger.*getInstance*();

System.*out*.println("Instance 1:"+obj1);

Logger obj2 = Logger.*getInstance*();

System.*out*.println("Instance 2:"+obj2);

if (obj1 == obj2) {

System.*out*.println("Both Objects are same .");

} else {

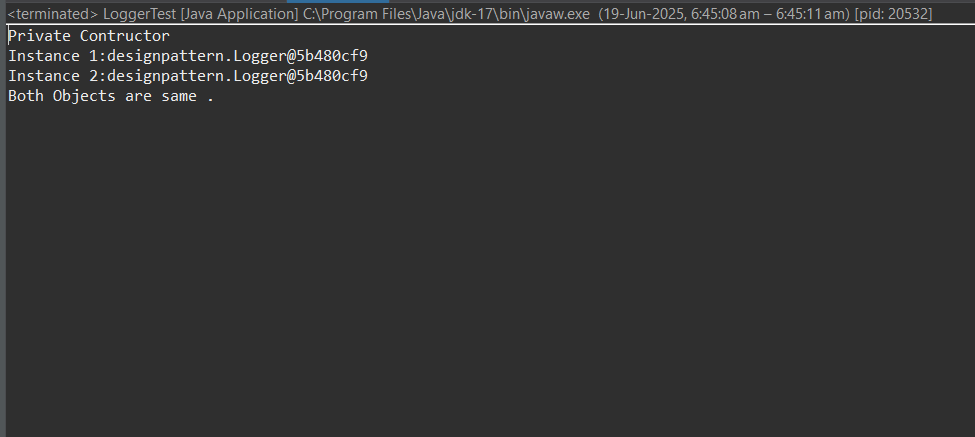
System.*out*.println("Different");

}

}

}

**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.

**Code:**

package FactoryExample;

interface Document {

String open();

}

class WordDocument implements Document {

public String open() {

return"Opening Word document.";

}

}

class PdfDocument implements Document {

public String open() {

return"Opening PDF document.";

}

}

class ExcelDocument implements Document {

public String open() {

return"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 DocumentFactoryExample {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument();

System.***out***.println( wordDoc.open());

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.createDocument();

System.***out***.println(pdf.open());

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.createDocument();

System.***out***.println(excel.open()); } }

Output:

