**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.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **SingletonPatternExample**.
2. **Define a Singleton Class:**
   * Create a class named Logger that has a private static instance of itself.
   * Ensure the constructor of Logger is private.
   * Provide a public static method to get the instance of the Logger class.
3. **Implement the Singleton Pattern:**
   * Write code to ensure that the Logger class follows the Singleton design pattern.
4. **Test the Singleton Implementation:**
   * Create a test class to verify that only one instance of Logger is created and used across the application.

**Answer :**

**Logger.java   
  
public class Logger {**

**// Step 1: Static variable to hold the only instance**

**private static Logger instance;**

**// Step 2: Private constructor to prevent other objects**

**private Logger() {**

**System.out.println("Logger Initialized");**

**}**

**// Step 3: Public method to provide access to the instance**

**public static Logger getInstance() {**

**if (instance == null) {**

**instance = new Logger();**

**}**

**return instance;**

**}**

**// Example method to show logging**

**public void log(String message) {**

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

**}**

**}  
  
Main.Java**

**public class Main {**

**public static void main(String[] args) {**

**// Get the singleton instance**

**Logger logger1 = Logger.getInstance();**

**logger1.log("Logging the first message.");**

**Logger logger2 = Logger.getInstance();**

**logger2.log("Logging the second message.");**

**// Test if both instances are same**

**if (logger1 == logger2) {**

**System.out.println("Logger is Singleton - Both instances are same.");**

**} else {**

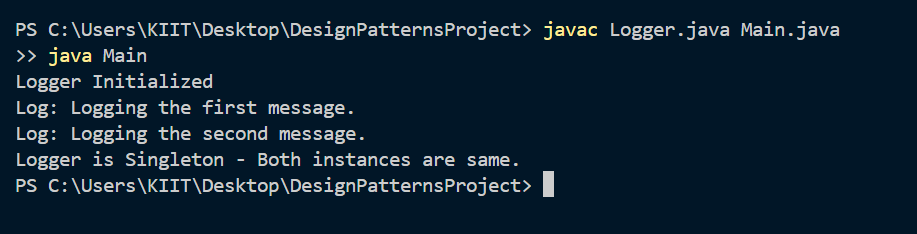
**System.out.println("Logger is NOT Singleton - Different instances.");**

**}**

**}**

**}**

**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.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **Implement the Factory Method:**
   * Create an abstract class **DocumentFactory** with a method **createDocument()**.
   * Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.

**Answer :**

**interface Document {**

**void open();**

**}**

**class WordDocument implements Document {**

**public void open() {**

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

**}**

**}**

**class PdfDocument implements Document {**

**public void open() {**

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

**}**

**}**

**class ExcelDocument implements Document {**

**public void open() {**

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

**}**

**}**

**class DocumentFactory {**

**public static Document createDocument(String type) {**

**switch (type.toLowerCase()) {**

**case "word":**

**return new WordDocument();**

**case "pdf":**

**return new PdfDocument();**

**case "excel":**

**return new ExcelDocument();**

**default:**

**throw new IllegalArgumentException("Unknown document type: " + type);**

**}**

**}**

**}**

**Main.java :**

**public class Main {**

**public static void main(String[] args) {**

**Document doc1 = DocumentFactory.createDocument("pdf");**

**doc1.open();**

**Document doc2 = DocumentFactory.createDocument("word");**

**doc2.open();**

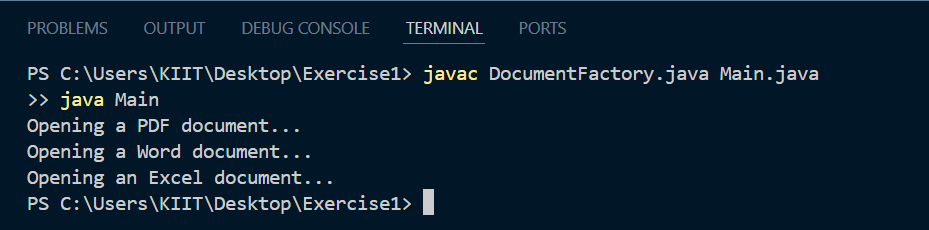
**Document doc3 = DocumentFactory.createDocument("excel");**

**doc3.open();**

**}**

**}**

**Output :**

****