**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

using System;

class Logger {

private static Logger \_instance;

private static readonly object \_lock = new object();

// Private constructor to prevent external instantiation

private Logger() {

Console.WriteLine("Logger created");

}

// Public static method to get the single instance

public static Logger GetInstance() {

if (\_instance == null) {

lock (\_lock) // Thread-safe lazy initialization

{

if (\_instance == null) {

\_instance = new Logger();

}

}

}

return \_instance;

}

public void Log(string message) {

Console.WriteLine($"[LOG]: {message}");

}

}

class SingletonDemo {

public static void Main() {

Logger logger1 = Logger.GetInstance();

Logger logger2 = Logger.GetInstance();

logger1.Log("Hello from logger1");

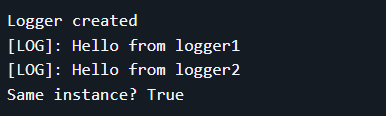
logger2.Log("Hello from logger2");

Console.WriteLine($"Same instance? {object.ReferenceEquals(logger1, logger2)}");

}

}

**OUTPUT:**



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

**CODE:**

using System;

// Step 2: Define abstract base class

abstract class Document {

public abstract void Open();

}

// Step 3: Concrete document classes

class WordDocument : Document {

public override void Open() {

Console.WriteLine("Opening a Word document.");

}

}

class PdfDocument : Document {

public override void Open() {

Console.WriteLine("Opening a PDF document.");

}

}

class ExcelDocument : Document {

public override void Open() {

Console.WriteLine("Opening an Excel document.");

}

}

// Step 4: Factory abstract class

abstract class DocumentFactory {

public abstract Document CreateDocument();

}

// Step 4: Concrete factories

class WordDocumentFactory : DocumentFactory {

public override Document CreateDocument() {

return new WordDocument();

}

}

class PdfDocumentFactory : DocumentFactory {

public override Document CreateDocument() {

return new PdfDocument();

}

}

class ExcelDocumentFactory : DocumentFactory {

public override Document CreateDocument() {

return new ExcelDocument();

}

}

// Step 5: Test class

class FactoryMethodDemo {

public static void Main() {

DocumentFactory wordFactory = new WordDocumentFactory();

Document word = wordFactory.CreateDocument();

word.Open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdf = pdfFactory.CreateDocument();

pdf.Open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excel = excelFactory.CreateDocument();

excel.Open();

}

}

**OUTPUT:**

