WEEK-1 Design principles & Patterns HandsON

Q1.**Implementing the Singleton Pattern**

using System;

public class Logger

{

    private static Logger \_instance;

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

    private Logger()

    {

        Console.WriteLine("Logger initialized.");

    }

    public static Logger GetInstance()

    {

        if (\_instance == null)

        {

            lock (\_lock)

            {

                if (\_instance == null)

                {

                    \_instance = new Logger();

                }

            }

        }

        return \_instance;

    }

    public void Log(string message)

    {

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

    }

}

class Program

{

    static void Main(string[] args)

    {

        Logger logger1 = Logger.GetInstance();

        Logger logger2 = Logger.GetInstance();

        logger1.Log("This is a test log message.");

        Console.WriteLine(object.ReferenceEquals(logger1, logger2)

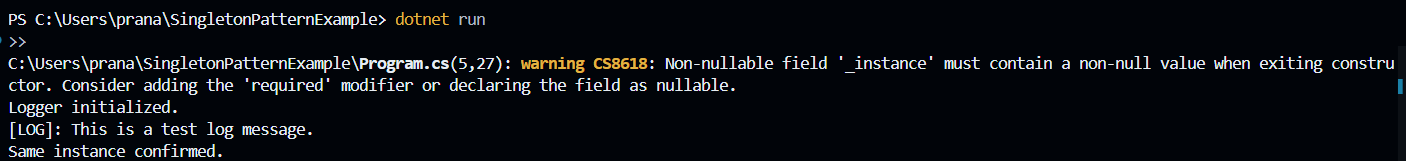
            ? "Same instance confirmed."

            : "Different instances detected!");

    }

}

**Question 1 OUTPUT:**



Q2.**Implementing the Factory Method Pattern**

using System;

public interface IDocument

{

    void Open();

}

public class WordDocument : IDocument

{

    public void Open()

    {

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

    }

}

public class PdfDocument : IDocument

{

    public void Open()

    {

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

    }

}

public class ExcelDocument : IDocument

{

    public void Open()

    {

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

    }

}

public abstract class DocumentFactory

{

    public abstract IDocument CreateDocument();

}

public class WordDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new WordDocument();

    }

}

public class PdfDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new PdfDocument();

    }

}

public class ExcelDocumentFactory : DocumentFactory

{

    public override IDocument CreateDocument()

    {

        return new ExcelDocument();

    }

}

class Program

{

    static void Main(string[] args)

    {

        DocumentFactory wordFactory = new WordDocumentFactory();

        IDocument wordDoc = wordFactory.CreateDocument();

        wordDoc.Open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        IDocument pdfDoc = pdfFactory.CreateDocument();

        pdfDoc.Open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        IDocument excelDoc = excelFactory.CreateDocument();

        excelDoc.Open();

    }

}

