**Exercise 1: Implementing the Singleton Pattern**

Codes :

Logger.cs:

using System;

*namespace* SingletonPatternExample

{

    public *class* Logger

    {

        private static Logger instance = null;

        private static readonly object padlock = new object();

private Logger()

        {

            Console.WriteLine("Logger Initialized.");

        }

public static Logger GetInstance()

        {

            if (instance == null)

            {

                lock (padlock)

                {

                    if (instance == null)

                    {

                        instance = new Logger();

                    }

                }

            }

            return instance;

        }

        public void Log(string *message*)

        {

            Console.WriteLine("Log Message: " + *message*);

        }

    }

}

Program.cs:

using System;

*namespace* SingletonPatternExample

{

*class* Program

    {

        static void Main(string[] *args*)

        {

            Logger logger1 = Logger.GetInstance();

            Logger logger2 = Logger.GetInstance();

logger1.Log("First log message.");

            logger2.Log("Second log message.");

if (logger1 == logger2)

            {

Console.WriteLine("Both logger instances are the same (Singleton works).");

            }

            else

            {

Console.WriteLine("Logger instances are different (Singleton failed).");

            }

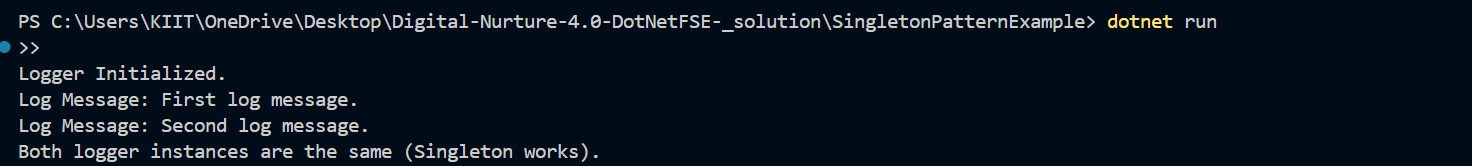
Console.ReadLine(); // Keep the console open

        }

    }

}

Output:



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

Code:

Program.cs:

using System;

*namespace* FactoryMethodPatternExample

{

    // Document Interface

public *interface* IDocument

    {

        void Open();

    }

// Word Document

    public *class* WordDocument : IDocument

    {

public void Open()

        {

            Console.WriteLine("Opening Word Document...");

        }

    }

// PDF Document

    public *class* PdfDocument : IDocument

    {

        public void Open()

        {

            Console.WriteLine("Opening PDF Document...");

        }

    }

// Excel Document

    public *class* ExcelDocument : IDocument

    {

        public void Open()

        {

            Console.WriteLine("Opening Excel Document...");

        }

    }

// Abstract Factory

    public abstract *class* DocumentFactory

    {

        public abstract IDocument CreateDocument();

    }

// Word Factory

    public *class* WordFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new WordDocument();

        }

    }

// PDF Factory

    public *class* PdfFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new PdfDocument();

        }

    }

// Excel Factory

    public *class* ExcelFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new ExcelDocument();

        }

    }

// Test Program

*class* Program

    {

        static void Main(string[] *args*)

        {

            // Create Word Document using Factory

            DocumentFactory wordFactory = new WordFactory();

            IDocument wordDoc = wordFactory.CreateDocument();

            wordDoc.Open();

// Create PDF Document using Factory

            DocumentFactory pdfFactory = new PdfFactory();

            IDocument pdfDoc = pdfFactory.CreateDocument();

            pdfDoc.Open();

// Create Excel Document using Factory

            DocumentFactory excelFactory = new ExcelFactory();

            IDocument excelDoc = excelFactory.CreateDocument();

            excelDoc.Open();

Console.ReadLine(); // Keep the console window open

        }

    }

}

Output:

