**Exercise 1: Implementing the Singleton Pattern**

using System;

namespace SingletonPatternExample

{

    public class Logger

    {

        private static Logger \_instance;

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

        private Logger()

        {

            Console.WriteLine("Logger instance created.");

        }

        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();

            logger1.Log("This is the first message.");

            Logger logger2 = Logger.GetInstance();

            logger2.Log("This is the second message.");

            if (logger1 == logger2)

            {

                Console.WriteLine("logger1 and logger2 are the same instance.");

            }

            else

            {

                Console.WriteLine("logger1 and logger2 are different instances.");

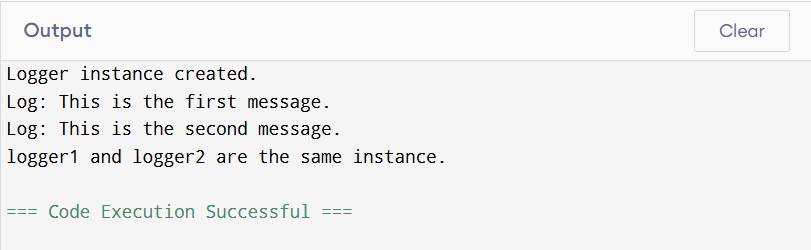
            }

        }

    }

}

Output -



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

using System;

namespace FactoryMethodPatternExample

{

    public abstract class Document

    {

        public abstract void Open();

    }

    public class WordDocument : Document

    {

        public override void Open()

        {

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

        }

    }

    public class PdfDocument : Document

    {

        public override void Open()

        {

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

        }

    }

    public class ExcelDocument : Document

    {

        public override void Open()

        {

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

        }

    }

    public abstract class DocumentFactory

    {

        public abstract Document CreateDocument();

    }

    public class WordDocumentFactory : DocumentFactory

    {

        public override Document CreateDocument()

        {

            return new WordDocument();

        }

    }

    public class PdfDocumentFactory : DocumentFactory

    {

        public override Document CreateDocument()

        {

            return new PdfDocument();

        }

    }

    public class ExcelDocumentFactory : DocumentFactory

    {

        public override Document CreateDocument()

        {

            return new ExcelDocument();

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            DocumentFactory wordFactory = new WordDocumentFactory();

            Document wordDoc = wordFactory.CreateDocument();

            wordDoc.Open();

            DocumentFactory pdfFactory = new PdfDocumentFactory();

            Document pdfDoc = pdfFactory.CreateDocument();

            pdfDoc.Open();

            DocumentFactory excelFactory = new ExcelDocumentFactory();

            Document excelDoc = excelFactory.CreateDocument();

            excelDoc.Open();

            Console.ReadLine();

        }

    }

}

Output -

