**Exercise 1: Implementing the Singleton Pattern**

**Solution**

**Filename - Logger.cs**

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 entry: {message}");

        }

    }

}

**Filename – Program.cs**

using System;

namespace SingletonPatternExample

{

    class Program

    {

        static void Main(string[] args)

        {

            Logger logger1 = Logger.GetInstance();

            Logger logger2 = Logger.GetInstance();

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

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

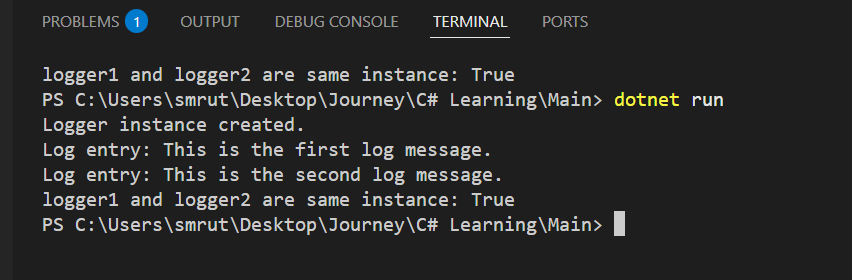
            Console.WriteLine($"logger1 and logger2 are same instance: {object.ReferenceEquals(logger1, logger2)}");

        }

    }

}

**OUTPUT**

****

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

**Solution**

**Filename – Exercise\_2.cs**

using System;

namespace FactoryMethodPatternExample

{

    // Document interface

    public interface IDocument

    {

        void Open();

    }

    // Concrete document types

    public class WordDocument : IDocument

    {

        public void Open() => Console.WriteLine("Opening Word Document...");

    }

    public class PdfDocument : IDocument

    {

        public void Open() => Console.WriteLine("Opening PDF Document...");

    }

    public class ExcelDocument : IDocument

    {

        public void Open() => Console.WriteLine("Opening Excel Document...");

    }

    // Abstract factory

    public abstract class DocumentFactory

    {

        public abstract IDocument CreateDocument();

    }

    // Concrete factories

    public class WordFactory : DocumentFactory

    {

        public override IDocument CreateDocument() => new WordDocument();

    }

    public class PdfFactory : DocumentFactory

    {

        public override IDocument CreateDocument() => new PdfDocument();

    }

    public class ExcelFactory : DocumentFactory

    {

        public override IDocument CreateDocument() => new ExcelDocument();

    }

    // Main class

    class Program

    {

        static void Main(string[] args)

        {

            DocumentFactory wordFactory = new WordFactory();

            IDocument word = wordFactory.CreateDocument();

            word.Open();

            DocumentFactory pdfFactory = new PdfFactory();

            IDocument pdf = pdfFactory.CreateDocument();

            pdf.Open();

            DocumentFactory excelFactory = new ExcelFactory();

            IDocument excel = excelFactory.CreateDocument();

            excel.Open();

        }

    }

}

**OUTPUT**

