**Cognizant Deep Skilling Week-1**

1.Design Principles and Patterns

Exercise 1:**Implementing the Singleton Pattern**

CODE:

using System;

namespace DesignPrinciples.Exercise1

{

    public class Logger

    {

        private static Logger instance;

        private Logger()

        {

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

        }

        public static Logger GetInstance()

        {

            if (instance == null)

            {

                instance = new Logger();

            }

            return instance;

        }

        public void Log(string message)

        {

            Console.WriteLine("LOG: " + message);

        }

    }

}

using System;

namespace DesignPrinciples.Exercise1

{

    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.");

            if (logger1 == logger2)

            {

                Console.WriteLine("Both logger1 and logger2 refer to the same instance.");

            }

            else

            {

                Console.WriteLine("Different instances exist (not a singleton).");

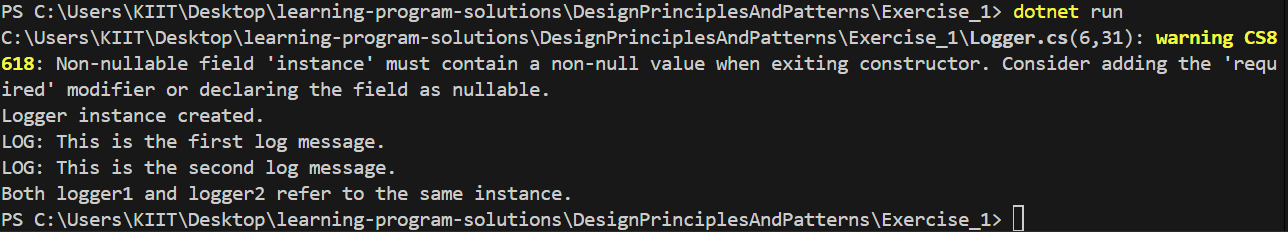
            }

        }

    }

}

OUTPUT:



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

CODE:

namespace FactoryMethodPatternExample

{

    public interface IDocument

    {

        void Open();

    }

}

using System;

namespace FactoryMethodPatternExample

{

    public class PdfDocument : IDocument

    {

        public void Open()

        {

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

        }

    }

}

using System;

namespace FactoryMethodPatternExample

{

    public class WordDocument : IDocument

    {

        public void Open()

        {

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

        }

    }

}

using System;

namespace FactoryMethodPatternExample

{

    public class ExcelDocument : IDocument

    {

        public void Open()

        {

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

        }

    }

}

amespace FactoryMethodPatternExample

{

    public class PdfFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new PdfDocument();

        }

    }

}

amespace FactoryMethodPatternExample

{

    public class WordFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new WordDocument();

        }

    }

}

namespace FactoryMethodPatternExample

{

    public class ExcelFactory : DocumentFactory

    {

        public override IDocument CreateDocument()

        {

            return new ExcelDocument();

        }

    }

}

namespace FactoryMethodPatternExample

{

    public abstract class DocumentFactory

    {

        public abstract IDocument CreateDocument();

    }

}

OUTPUT:

