**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

using System;

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: " + message);

    }

}

public class Program

{

    public static void Main()

    {

        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)

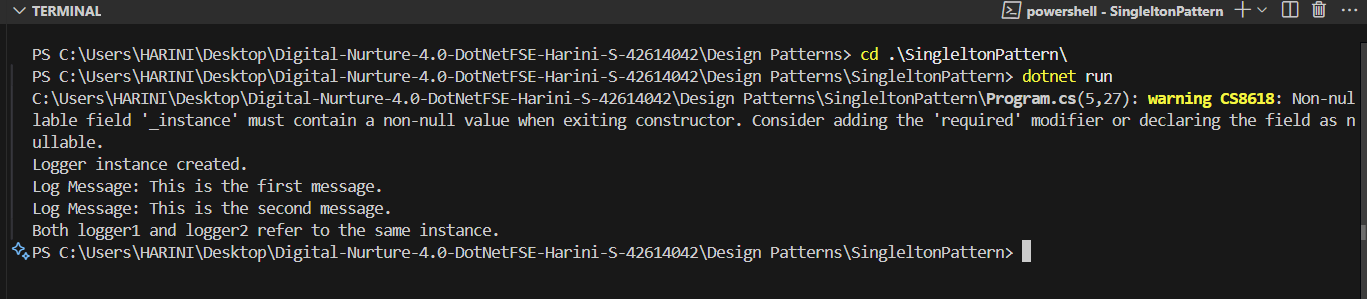
        {

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

        }

    }

}



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

**Code:**

using System;

public interface IDocument

{

    void Open();

}

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...");

    }

}

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();

    }

}

public class Program

{

    public static void Main()

    {

        DocumentFactory wordFactory = new WordDocumentFactory();

        IDocument word = wordFactory.CreateDocument();

        word.Open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        IDocument pdf = pdfFactory.CreateDocument();

        pdf.Open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        IDocument excel = excelFactory.CreateDocument();

        excel.Open();

    }

}

**Output:**

