WEEK 1 DESIGN AND PATTERN AND PRINCIPLES HANDSON

**Exercise 1: Implementing the Singleton Pattern:**

**Code**

using System;

class Logger

{

    private static readonly Logger instance = new Logger();

    private Logger()

    {

        Console.WriteLine("Logger initialized");

    }

    public static Logger GetInstance()

    {

        return instance;

    }

    public void Log(string message)

    {

        Console.WriteLine($"LOG: {message}");

    }

}

class Program

{

    static void Main()

    {

        Logger logger1 = Logger.GetInstance();

        Logger logger2 = Logger.GetInstance();

        logger1.Log("Application started");

        logger2.Log("Performing some operations");

        if (ReferenceEquals(logger1, logger2))

        {

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

        }

        else

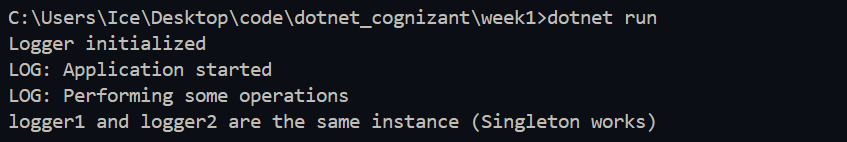
        {

            Console.WriteLine("Different instances detected (Singleton failed)");

        }

    }

}



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

**Code:**

using System;

interface Document

{

    void Open();

}

class WordDocument : Document

{

    public void Open()

    {

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

    }

}

class PdfDocument : Document

{

    public void Open()

    {

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

    }

}

class ExcelDocument : Document

{

    public void Open()

    {

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

    }

}

abstract class DocumentFactory

{

    public abstract Document CreateDocument();

}

class WordDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new WordDocument();

    }

}

class PdfDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new PdfDocument();

    }

}

class ExcelDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new ExcelDocument();

    }

}

class Program

{

    static void Main()

    {

        DocumentFactory wordFactory = new WordDocumentFactory();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document wordDoc = wordFactory.CreateDocument();

        Document pdfDoc = pdfFactory.CreateDocument();

        Document excelDoc = excelFactory.CreateDocument();

        wordDoc.Open();

        pdfDoc.Open();

        excelDoc.Open();

    }

}

