**Exercise 1: Implementing the Singleton Pattern**

**Code:**

using System;

*public* class Logger

{

*private* *static* Logger instance;

*private* *static* *readonly* object lockObj = new object();

*private* Logger()

    {

        Console.WriteLine("Logger initialized.");

    }

*public* *static* Logger GetInstance()

    {

        if (instance == null)

        {

            lock (lockObj)

            {

                if (instance == null)

                {

                    instance = new Logger();

                }

            }

        }

        return instance;

    }

*public* void Log(string message)

    {

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

    }

}

class Test

{

*static* void Main(string[] args)

    {

        Logger logger1 = Logger.GetInstance();

        Logger logger2 = Logger.GetInstance();

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

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

        if (logger1 == logger2)

        {

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

        }

        else

        {

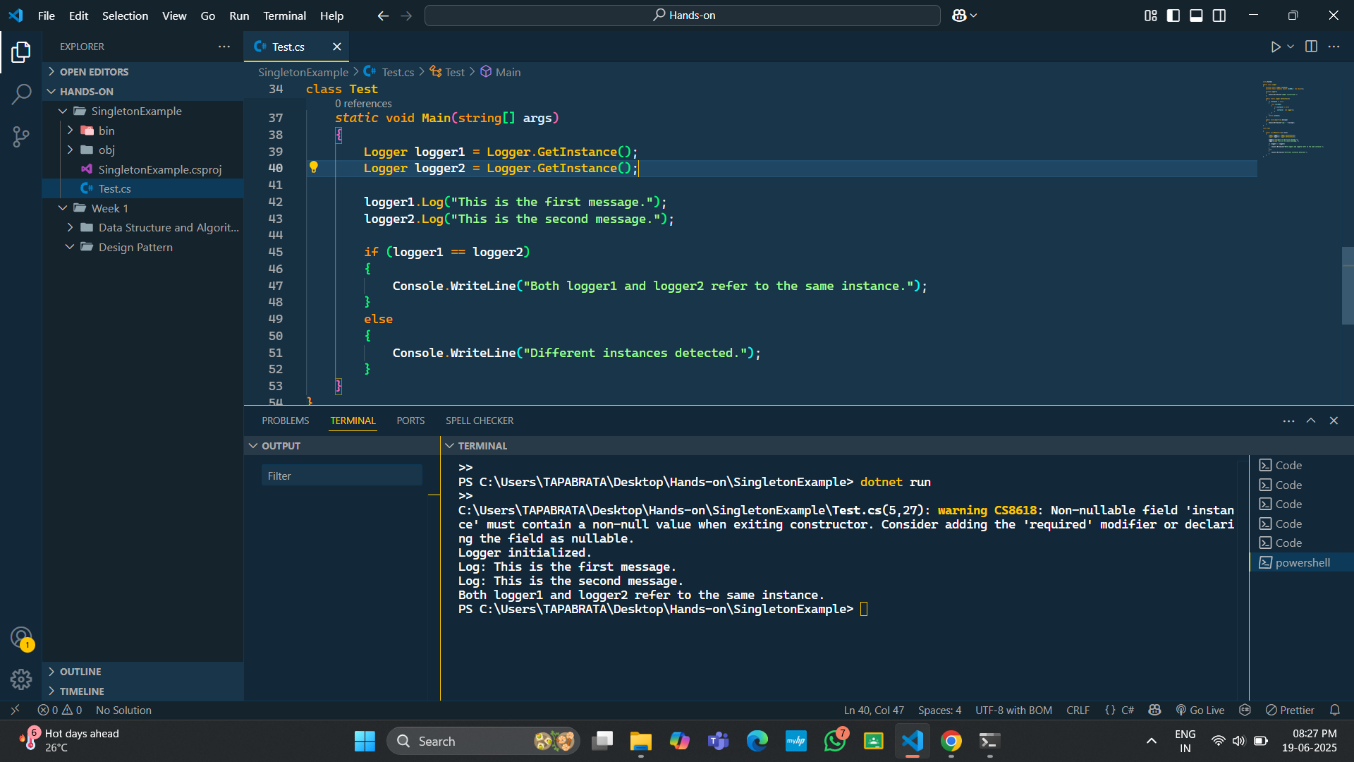
            Console.WriteLine("Different instances detected.");

        }

    }

}

Output:



**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 a Word document.");

    }

}

*public* class PdfDocument : *IDocument*

{

*public* void Open()

    {

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

    }

}

*public* class ExcelDocument : *IDocument*

{

*public* void Open()

    {

        Console.WriteLine("Opening an 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();

    }

}

class Test

{

*static* void Main(string[] args)

    {

        DocumentFactory wordFactory = new WordDocumentFactory();

*IDocument* wordDoc = wordFactory.CreateDocument();

        wordDoc.Open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

*IDocument* pdfDoc = pdfFactory.CreateDocument();

        pdfDoc.Open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

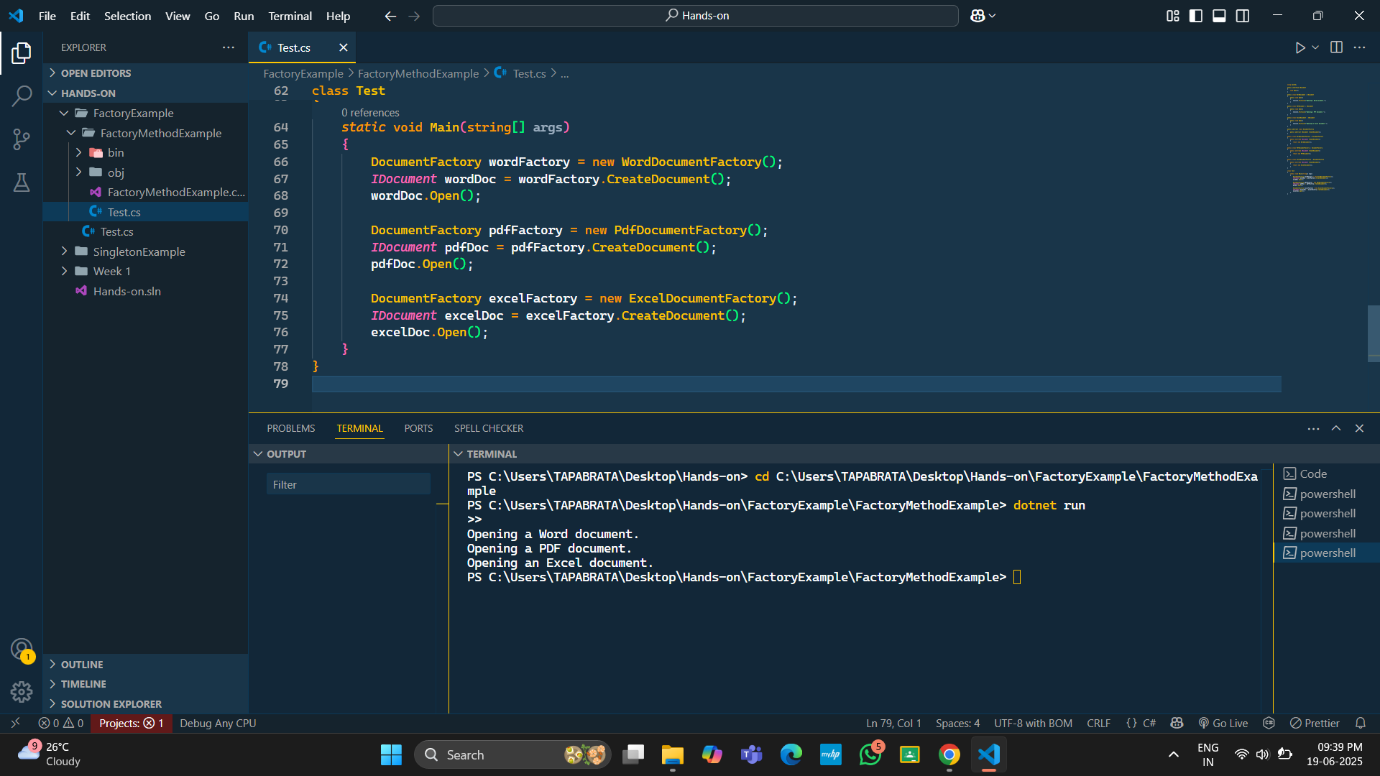
        IDocument excelDoc = excelFactory.CreateDocument();

        excelDoc.Open();

    }

}

Output:

****