Exercise 1: Implementing the Singleton Pattern

using System;

public class Logger

{

    private static Logger \_instance;

    private static readonly object \_lock = new object();

    private Logger()

    {

        Console.WriteLine("Logger initialized.");

    }

    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: " + message);

    }

}

class Program

{

    static void Main(string[] args)

    {

        Logger logger1 = Logger.GetInstance();

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

        Logger logger2 = Logger.GetInstance();

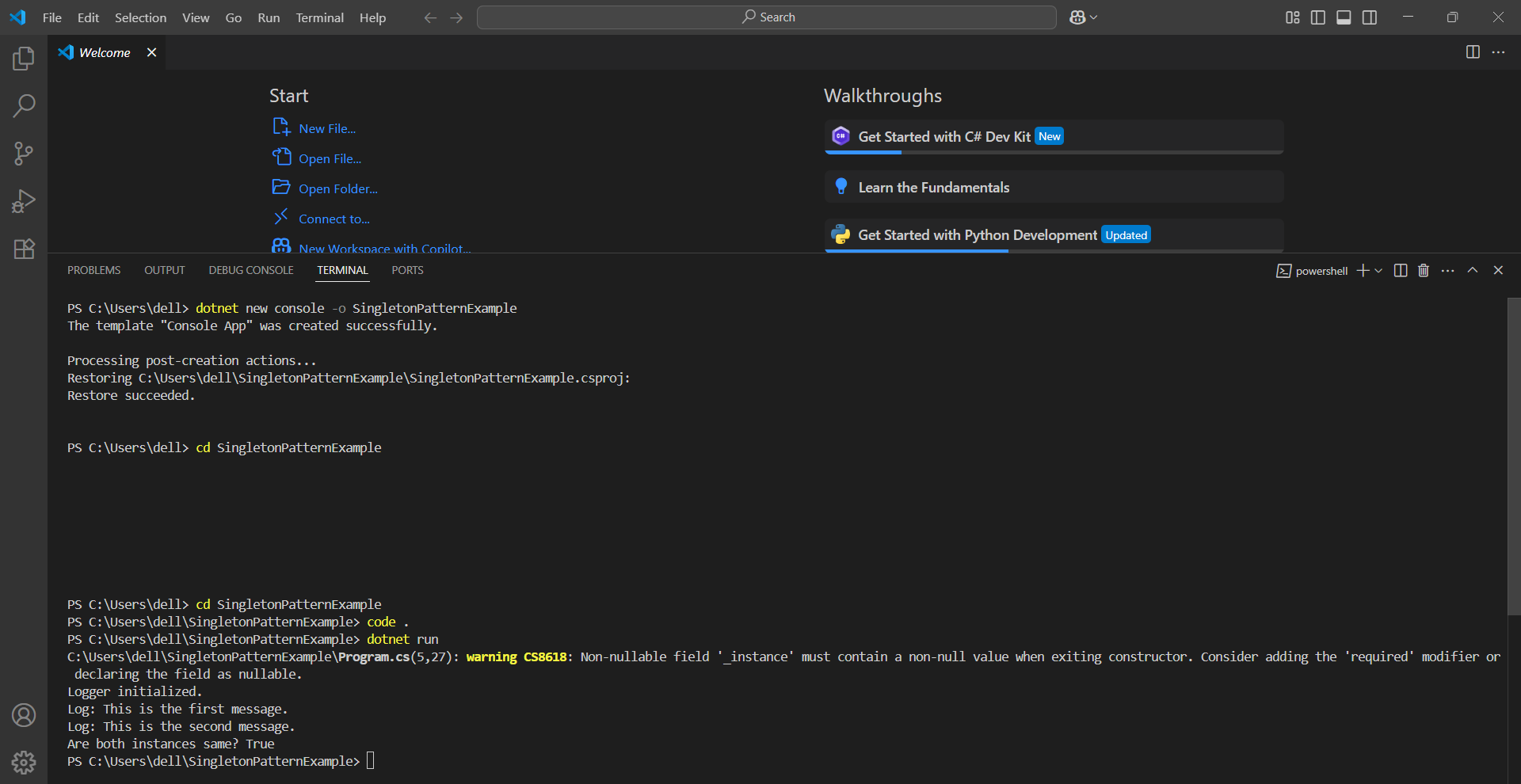
        logger2.Log("This is the second message.");

        Console.WriteLine($"Are both instances same? {ReferenceEquals(logger1, logger2)}");

    }

}

Output :



Exercise 2: Implementing the Factory Method Pattern

Program :

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();

}

}

class Program

{

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 :

