**Exercise 1: Implementing the Singleton Pattern**

*CODE:*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

public interface ILogger

{

void Log(string message);

}

public sealed class Logger : ILogger

{

private static Logger \_instance = null;

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

private Logger() { }

public static Logger Instance

{

get

{

lock (\_lock)

{

return \_instance ??= new Logger();

}

}

}

public void Log(string message)

{

Console.WriteLine($"[Log]: {message}");

}

}

public class Application

{

private readonly ILogger \_logger;

public Application(ILogger logger)

{

\_logger = logger;

}

public void Run()

{

\_logger.Log("Application started.");

}

}

class Program

{

static void Main()

{

ILogger logger = Logger.Instance;

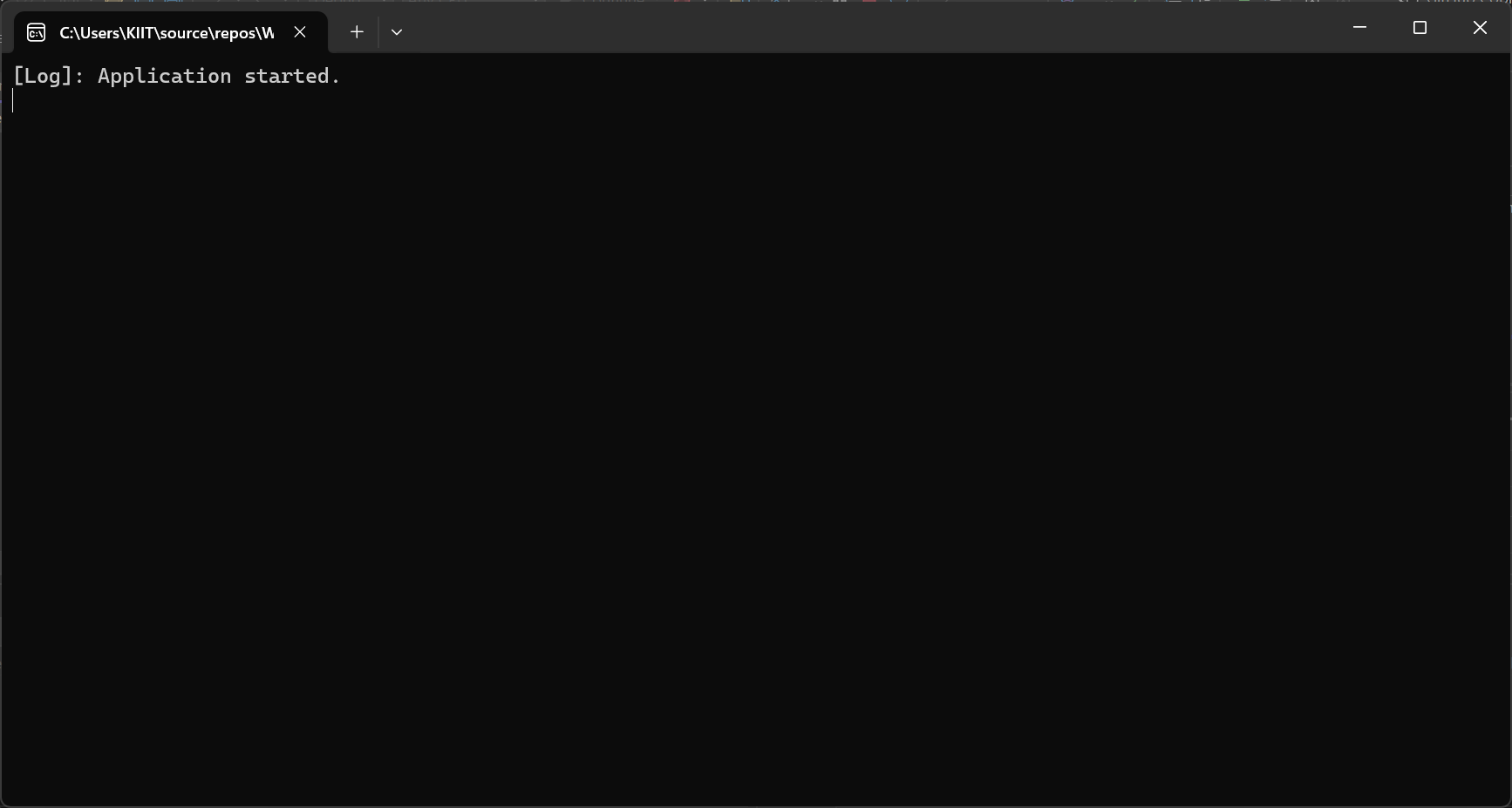
var app = new Application(logger);

app.Run();

Console.ReadLine();

}

}

*OUTPUT*:

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

*CODE:*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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 interface IDocumentFactory

{

IDocument CreateDocument();

}

public class WordDocumentFactory : IDocumentFactory

{

public IDocument CreateDocument() => new WordDocument();

}

public class PdfDocumentFactory : IDocumentFactory

{

public IDocument CreateDocument() => new PdfDocument();

}

public class ExcelDocumentFactory : IDocumentFactory

{

public IDocument CreateDocument() => new ExcelDocument();

}

public class DocumentService

{

private readonly IDocumentFactory \_factory;

public DocumentService(IDocumentFactory factory)

{

\_factory = factory;

}

public void OpenDocument()

{

var doc = \_factory.CreateDocument();

doc.Open();

}

}

class Program

{

static void Main()

{

IDocumentFactory factory = new PdfDocumentFactory();

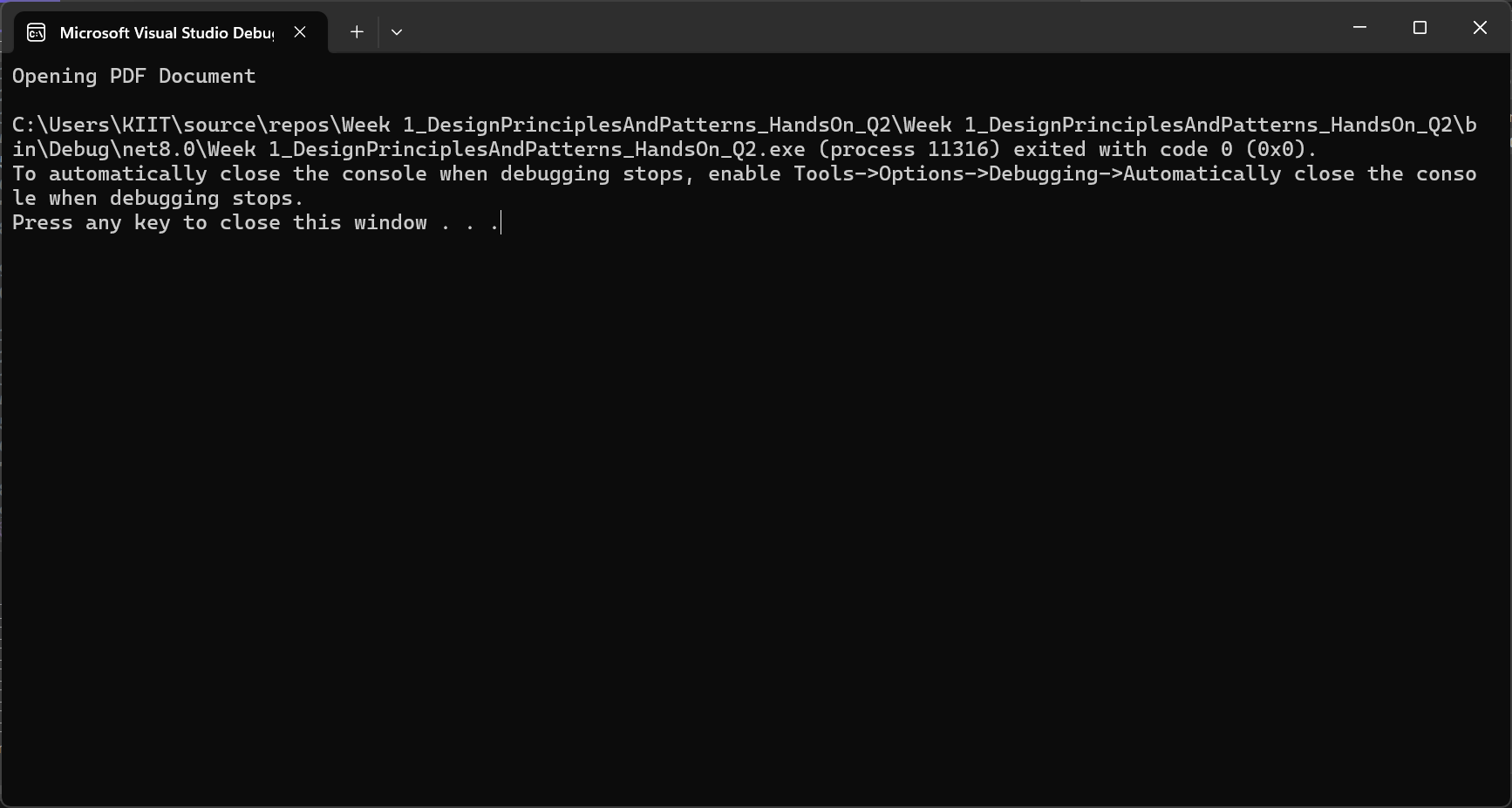
var service = new DocumentService(factory);

service.OpenDocument();

}

}

*OUTPUT:*

****