**Exercise 1: Implementing the Singleton Pattern**

**Code for the Singleton Class :**

namespace SingletonPatternExample;

public class Logger

{

private static Logger instance;

private static readonly object lockObj = new object();

private Logger() { }

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}");

}

}

**Code for the testing of Singleton Implementation :**

using System;

namespace SingletonPatternExample

{

class Program

{

static void Main(string[] args)

{

Logger logger1 = Logger.GetInstance();

Logger logger2 = Logger.GetInstance();

logger1.Log("1st message.");

logger2.Log("2nd message.");

if (ReferenceEquals(logger1, logger2))

{

Console.WriteLine("Both logger instances are same. Singleton works!");

}

else

{

Console.WriteLine("Different logger instances exist. Singleton failed!");

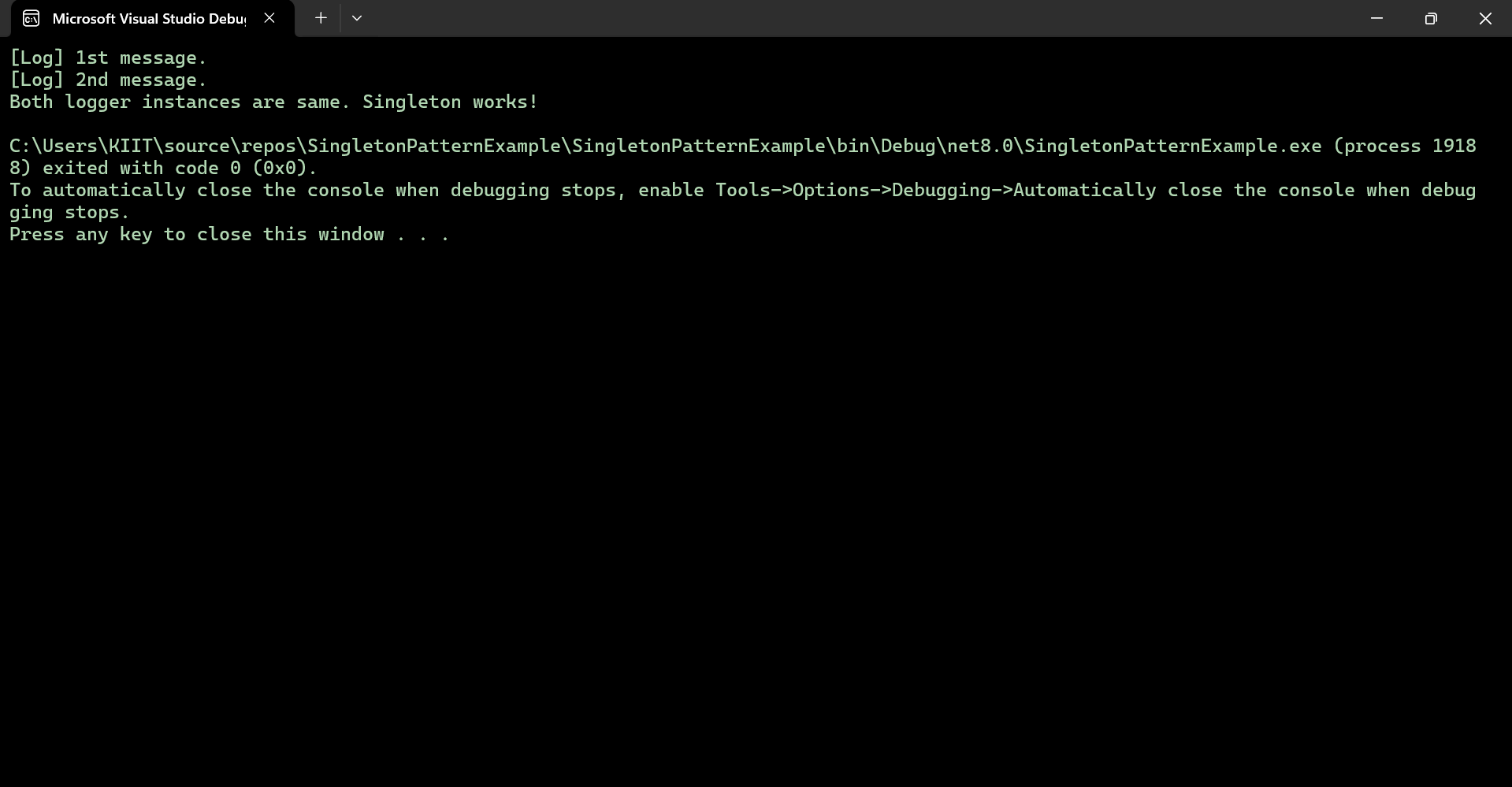
}

}

}

}

**OUTPUT :**

****

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

**Document Interface:**

namespace FactoryMethodPatternExample

{

public interface Document

{

void Open();

}

}

**Word Document Class:**

using System;

namespace FactoryMethodPatternExample

{

public class WordDocument : Document

{

public void Open()

{

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

}

}

}

**PdfDocument Class:**

using System;

namespace FactoryMethodPatternExample

{

public class PdfDocument : Document

{

public void Open()

{

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

}

}

}

**ExcelDocument Class:**

using System;

namespace FactoryMethodPatternExample

{

public class ExcelDocument : Document

{

public void Open()

{

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

}

}

}

**Implementation interface of factory method:**

namespace FactoryMethodPatternExample

{

public abstract class DocumentFactory

{

public abstract Document CreateDocument();

}

}

**WordDocumentFactory class :**

namespace FactoryMethodPatternExample

{

public class WordDocumentFactory : DocumentFactory

{

public override Document CreateDocument()

{

return new WordDocument();

}

}

}

**PdfDocumentFactory class :**

namespace FactoryMethodPatternExample

{

public class PdfDocumentFactory : DocumentFactory

{

public override Document CreateDocument()

{

return new PdfDocument();

}

}

}

**ExcelDocumentFactory class :**

namespace FactoryMethodPatternExample

{

public class ExcelDocumentFactory : DocumentFactory

{

public override Document CreateDocument()

{

return new ExcelDocument();

}

}

}

**Testing of Factory Implementation :**

using System;

namespace FactoryMethodPatternExample

{

class Program

{

static void Main(string[] args)

{

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.CreateDocument();

wordDoc.Open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.CreateDocument();

pdfDoc.Open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.CreateDocument();

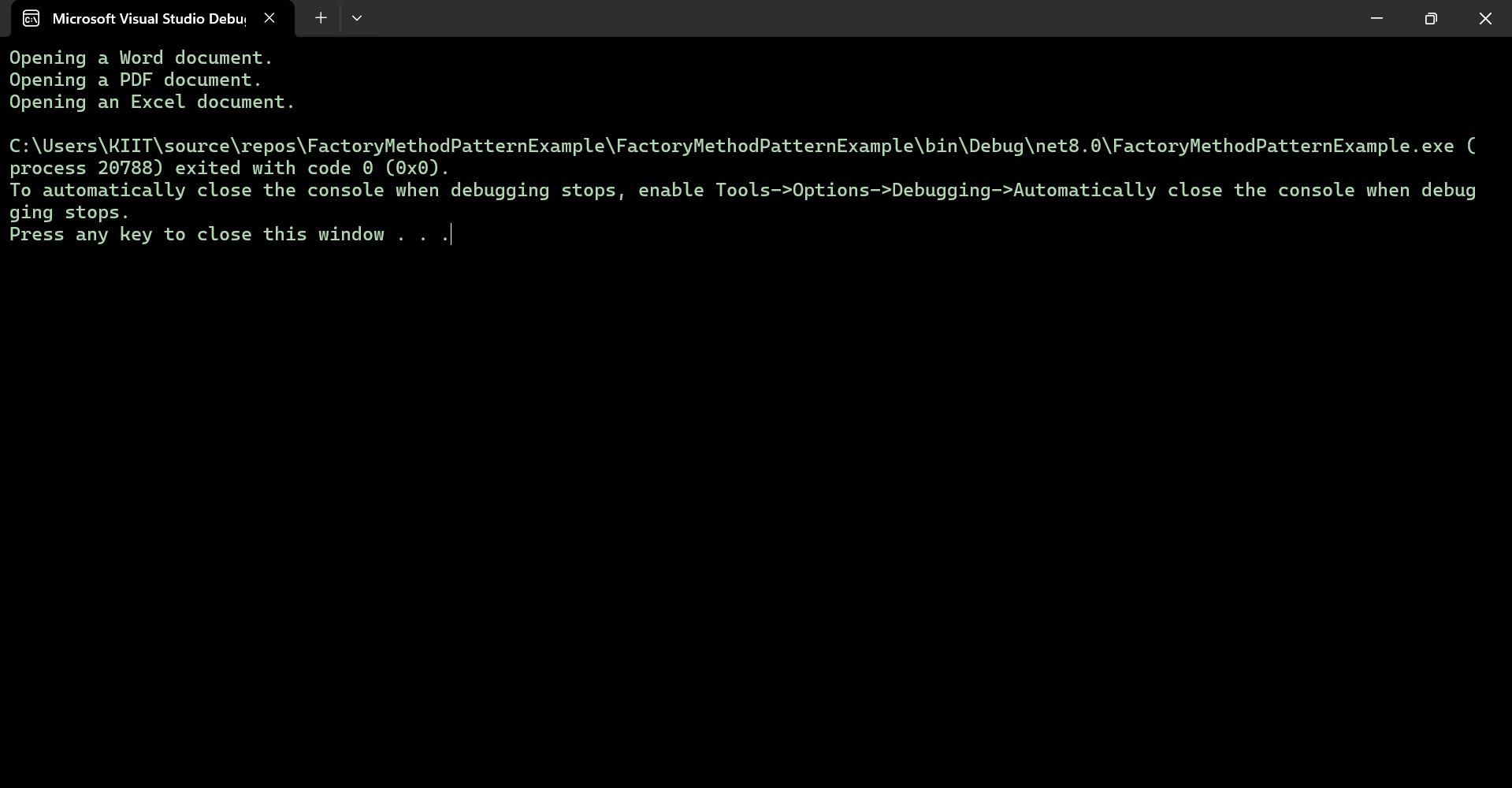
excelDoc.Open();

}

}

}

**OUTPUT :**

****