**EXERCISE 1: Implementing the Singleton Class**

**CODE**

using System;

namespace SingletonPattern

{

public sealed class Logger

{

private static readonly Logger instance = new Logger();

private Logger(){}

public static Logger Instance

{

get { return instance; }

}

public void Log(string message)

{

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

}

}

class Program

{

static void Main(string[] args)

{

Logger logger1 = Logger.Instance;

Logger logger2 = Logger.Instance;

logger1.Log("This is a log message.");

if (object.ReferenceEquals(logger1, logger2))

{

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

}

else

{

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

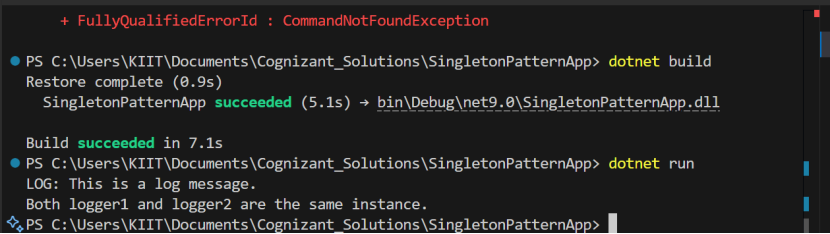
}

}

}

}

**OUTPUT(Exercise-1)**



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

**CODE**

using System;

namespace FactoryMethodPatternExample

{

public interface IDocument

{

void Open();

void Close();

}

public class WordDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening Word Document");

}

public void Close()

{

Console.WriteLine("Closing Word Document");

}

}

public class PdfDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening PDF Document");

}

public void Close()

{

Console.WriteLine("Closing PDF Document");

}

}

public class ExcelDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening Excel Document");

}

public void Close()

{

Console.WriteLine("Closing 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)

{

Console.WriteLine("Document Management System");

Console.WriteLine("--------------------------");

// Create factories

DocumentFactory wordFactory = new WordDocumentFactory();

DocumentFactory pdfFactory = new PdfDocumentFactory();

DocumentFactory excelFactory = new ExcelDocumentFactory();

// Create and use documents

Console.WriteLine("\nCreating Word Document:");

IDocument wordDoc = wordFactory.CreateDocument();

wordDoc.Open();

wordDoc.Close();

Console.WriteLine("\nCreating PDF Document:");

IDocument pdfDoc = pdfFactory.CreateDocument();

pdfDoc.Open();

pdfDoc.Close();

Console.WriteLine("\nCreating Excel Document:");

IDocument excelDoc = excelFactory.CreateDocument();

excelDoc.Open();

excelDoc.Close();

Console.WriteLine("\nPress any key to exit...");

Console.ReadKey();

}

}

}

**P.T.O -->**

**OUTPUT(Exercise-2)**

