**Name: Deepanshu Rathore**

**Superset ID: 6358199**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Week 1  
Exercise 1: Implementing the Singleton Pattern  
  
Code:  
Console.WriteLine("Singleton Pattern Example - Logger");

Console.WriteLine("\nCreating first logger instance:");

Logger logger1 = Logger.GetInstance();

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

Console.WriteLine("\nCreating second logger instance:");

Logger logger2 = Logger.GetInstance();

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

Console.WriteLine("\nChecking if both loggers are the same instance:");

if (ReferenceEquals(logger1, logger2))

{

Console.WriteLine("Both logger variables refer to the same instance.");

}

else

{

Console.WriteLine("Error: Different instances were created!");

}

Console.WriteLine("\nTest completed. Press any key to exit.");

Console.ReadKey();

public sealed class Logger

{

private static Logger? \_instance;

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

private Logger()

{

Console.WriteLine("Logger instance created.");

}

public static Logger GetInstance()

{

if (\_instance == null)

{

lock (\_lock)

{

if (\_instance == null)

{

\_instance = new Logger();

}

}

}

return \_instance;

}

public void Log(string message)

{

Console.WriteLine($"[{DateTime.Now}] LOG: {message}");

}

}

Output:  
A screenshot of a computer

AI-generated content may be incorrect.  
  
Exercise 2: Implementing the Factory Method Pattern  
  
Code:  
using System;

// Document interface

public interface IDocument

{

void Open();

void Save();

void Close();

}

// Concrete document classes

public class WordDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening Word document");

}

public void Save()

{

Console.WriteLine("Saving Word document");

}

public void Close()

{

Console.WriteLine("Closing Word document");

}

}

public class PdfDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening PDF document");

}

public void Save()

{

Console.WriteLine("Saving PDF document");

}

public void Close()

{

Console.WriteLine("Closing PDF document");

}

}

public class ExcelDocument : IDocument

{

public void Open()

{

Console.WriteLine("Opening Excel document");

}

public void Save()

{

Console.WriteLine("Saving Excel document");

}

public void Close()

{

Console.WriteLine("Closing Excel document");

}

}

// Abstract factory class

public abstract class DocumentFactory

{

public abstract IDocument CreateDocument();

public void OperateDocument()

{

IDocument document = CreateDocument();

document.Open();

document.Save();

document.Close();

}

}

// Concrete factory classes

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("Factory Method Pattern Example - Document Management System");

Console.WriteLine("\nCreating and operating on a Word document:");

DocumentFactory wordFactory = new WordDocumentFactory();

wordFactory.OperateDocument();

Console.WriteLine("\nCreating and operating on a PDF document:");

DocumentFactory pdfFactory = new PdfDocumentFactory();

pdfFactory.OperateDocument();

Console.WriteLine("\nCreating and operating on an Excel document:");

DocumentFactory excelFactory = new ExcelDocumentFactory();

excelFactory.OperateDocument();

Console.WriteLine("\nTest completed. Press any key to exit.");

Console.ReadKey();

}

}  
  
Output:

A computer screen shot of a black screen

AI-generated content may be incorrect.