# **Name- Sayandeep Dey (SupersetID:** **6363427)**

# **WEEK – 1 (Handson- Exercises)**

1. **Design principles & Patterns:**

**Exercise 1: Implementing the Singleton Pattern:**

**Code:**

**In Logger.cs:**

using System;

public class Logger

{

    private static Logger? instance = null;

    private static readonly object padlock = new object();

    private Logger()

    {

        Console.WriteLine("Logger Initialized");

    }

    public static Logger Instance

    {

        get

        {

            lock (padlock)

            {

                if (instance == null)

                {

                    instance = new Logger();

                }

                return instance;

            }

        }

    }

    public void Log(string message)

    {

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

    }

}

**In Program.cs:**

using System;

class Program

{

    static void Main(string[] args)

    {

        Logger logger1 = Logger.Instance;

        logger1.Log("Application started.");

        Logger logger2 = Logger.Instance;

        logger2.Log("Another log message.");

        if (logger1 == logger2)

        {

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

        }

        else

        {

            Console.WriteLine("Logger instances are different. Singleton failed!");

        }

    }

}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

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

**Code:**

**Document.cs – Interface:**

public interface Document

{

    void Open();

}

**WordDocument.cs:**

using System;

public class WordDocument : Document

{

    public void Open()

    {

        Console.WriteLine("Opening Word Document");

    }

}

**PdfDocument.cs:**

using System;

public class PdfDocument : Document

{

    public void Open()

    {

        Console.WriteLine("Opening PDF Document");

    }

}

**ExcelDocument.cs:**

using System;

public class ExcelDocument : Document

{

    public void Open()

    {

        Console.WriteLine("Opening Excel Document");

    }

}

**DocumentFactory.cs:**

public abstract class DocumentFactory

{

    public abstract Document CreateDocument();

}

**WordDocumentFactory.cs:**

public class WordDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new WordDocument();

    }

}

**PdfDocumentFactory.cs:**

public class PdfDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new PdfDocument();

    }

}

**ExcelDocumentFactory.cs:**

public class ExcelDocumentFactory : DocumentFactory

{

    public override Document CreateDocument()

    {

        return new ExcelDocument();

    }

}

**Program.cs:**using System;

class Program

{

    static void Main(string[] args)

    {

        DocumentFactory wordFactory = new WordDocumentFactory();

        Document word = wordFactory.CreateDocument();

        word.Open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();

        Document pdf = pdfFactory.CreateDocument();

        pdf.Open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();

        Document excel = excelFactory.CreateDocument();

        excel.Open();

    }

}

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**