Name:-Ritesh Kumar Sahoo

GitHub Link:-[https://github.com/Riteshksahoo/Cognizant\_6364581\_Ritesh](#_top)CognizantDotNet

**-----------------------------------------------------------------------------------------**

**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**SingletonPatternExample---->   
  
Program.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\SingletonPatternExample\Program.cs)  
  
**INPUT**

**using System;**

**public class Logger**

**{**

**private static Logger instance = null;**

**private Logger()**

**{**

**Console.WriteLine("Hello,Logger instance has been created.");**

**}**

**public static Logger GetInstance()**

**{**

**if (instance == null)**

**{**

**instance = new Logger();**

**}**

**return instance;**

**}**

**public void Log(string message)**

**{**

**Console.WriteLine("Log:- " + message);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Logger logger1 = Logger.GetInstance();**

**logger1.Log("First log message.");**

**Logger logger2 = Logger.GetInstance();**

**logger2.Log("Second log message.");**

**if (logger1 == logger2)**

**{**

**Console.WriteLine("Only one instance exists.");**

**Console.WriteLine("Singleton works.");**

**}**

**else**

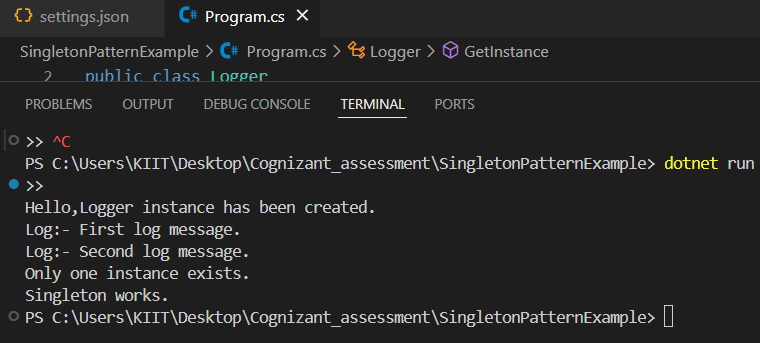
**{**

**Console.WriteLine("Different instances found. Singleton failed.");**

**Console.WriteLine("Sorry,Singleton failed.");**

**}**

**}**

**}  
  
OUTPUT**

**-----------------------------------------------------------------------------------------**

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

**FactoryMethodPatternExample---->   
  
IFile.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\IFile.cs)  
  
**INPUT  
  
public interface IFile**

**{**

**void Open();**

**void Save();**

**void Close();**

**}  
  
WordFile.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\WordFile.cs)  
  
**INPUT  
  
public class WordFile : IFile**

**{**

**public void Open() => Console.WriteLine("Wait,Opening Word file.");**

**public void Save() => Console.WriteLine("Saving Word file.");**

**public void Close() => Console.WriteLine("Closing Word file,Thank You.");**

**}**

**PdfFile.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\PdfFile.cs)  
  
**INPUT**

**public class PdfFile : IFile**

**{**

**public void Open() => Console.WriteLine("Wait,Opening PDF file.");**

**public void Save() => Console.WriteLine("Saving PDF file.");**

**public void Close() => Console.WriteLine("Closing PDF file,Thank You.");**

**}**

**ExcelFile.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\ExcelFile.cs)  
  
**INPUT  
  
public class ExcelFile : IFile**

**{**

**public void Open() => Console.WriteLine("Wait,Opening Excel file.");**

**public void Save() => Console.WriteLine("Saving Excel file.");**

**public void Close() => Console.WriteLine("Closing Excel file,Thank You.");**

**}**

**FileFactory.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\FileFactory.cs)  
  
**INPUT  
  
public abstract class FileFactory**

**{**

**public abstract IFile CreateFile();**

**}  
  
WordFileFactory.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\WordFileFactory.cs)  
  
**INPUT**

**public class WordFileFactory : FileFactory**

**{**

**public override IFile CreateFile()**

**{**

**return new WordFile();**

**}**

**}**

**PdfFileFactory.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\PdfFileFactory.cs)  
  
**INPUT**

**public class PdfFileFactory : FileFactory**

**{**

**public override IFile CreateFile()**

**{**

**return new PdfFile();**

**}**

**}  
  
 ExcelFileFactory.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\ExcelFileFactory.cs)  
  
**INPUT**

**public class ExcelFileFactory : FileFactory**

**{**

**public override IFile CreateFile()**

**{**

**return new ExcelFile();**

**}**

**}  
  
 Program.cs**(C:\Users\KIIT\Desktop\Cognizant\_assessment\FactoryMethodPatternExample\Program.cs)  
  
**INPUT  
  
using System;**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Console.WriteLine("Factory Method Pattern\n");**

**FileFactory wordFactory = new WordFileFactory();**

**IFile wordFile = wordFactory.CreateFile();**

**wordFile.Open();**

**wordFile.Save();**

**wordFile.Close();**

**Console.WriteLine("Loading....");**

**Console.WriteLine("");**

**FileFactory pdfFactory = new PdfFileFactory();**

**IFile pdfFile = pdfFactory.CreateFile();**

**pdfFile.Open();**

**pdfFile.Save();**

**pdfFile.Close();**

**Console.WriteLine("Loading....");**

**Console.WriteLine("");**

**FileFactory excelFactory = new ExcelFileFactory();**

**IFile excelFile = excelFactory.CreateFile();**

**excelFile.Open();**

**excelFile.Save();**

**excelFile.Close();**

**Console.WriteLine("Demo Completed Successfully.");**

**Console.WriteLine("Thank You");**

**}**

**}**

**OUTPUT**