**Lesson04 Interface Segregation Principle**

**Notes:-**

**1-basic idea is when you have interfaces with too much stuff break it into separate parts based on the class needs**

**using System;**

**namespace InterfaceSegerationPro{**

**public class Document{}**

**public interface IMachine{**

**void Print(Document d);**

**void Fax(Document d);**

**void Scan(Document d);}**

**// ok if you need a multifunction machine**

**public class MultiFunctionPrinter : IMachine{**

**public void Print(Document d){}**

**public void Fax(Document d){}**

**public void Scan(Document d){}}**

**//if we have old printer that make only print , we see that the Fax and scan not used**

**public class OldFashionedPrinter : IMachine{**

**public void Print(Document d){// yep}**

**public void Fax(Document d){throw new System.NotImplementedException();}**

**public void Scan(Document d){throw new System.NotImplementedException();}}**

**//the solution is to split into new interfaces as below**

**public interface IPrinter{void Print(Document d);}**

**public interface IScanner{void Scan(Document d);}**

**public class Printer : IPrinter{public void Print(Document d){}}**

**public class Photocopier : IPrinter, IScanner{**

**public void Print(Document d){throw new System.NotImplementedException();}**

**public void Scan(Document d){throw new System.NotImplementedException();}}**

**public interface IMultiFunctionDevice : IPrinter, IScanner //{}**

**//we can use composite interface the inherit from the multiple interfaces**

**public struct MultiFunctionMachine : IMultiFunctionDevice{**

**// compose this out of several modules**

**private IPrinter printer;**

**private IScanner scanner;**

**public MultiFunctionMachine(IPrinter printer, IScanner scanner){**

**if (printer == null){throw new ArgumentNullException(paramName: nameof(printer));}**

**if (scanner == null){throw new ArgumentNullException(paramName: nameof(scanner));}**

**this.printer = printer;**

**this.scanner = scanner;}**

**public void Print(Document d){printer.Print(d);}**

**public void Scan(Document d){scanner.Scan(d);}}}**