Multiple Inheritance in C# Using Interfaces - Real World Examples

# Example 1: Smart Device System

A SmartPhone can make calls and take photos. It implements both IPhone and ICamera interfaces.

Interfaces:

interface IPhone  
{  
 void MakeCall(string number);  
}  
  
interface ICamera  
{  
 void TakePhoto();  
}

Class Implementation:

class SmartPhone : IPhone, ICamera  
{  
 public void MakeCall(string number)  
 {  
 Console.WriteLine($"Calling {number}...");  
 }  
  
 public void TakePhoto()  
 {  
 Console.WriteLine("Photo taken with 12MP camera.");  
 }  
  
 public void BrowseInternet()  
 {  
 Console.WriteLine("Browsing internet...");  
 }  
}

Usage:

SmartPhone sp = new SmartPhone();  
sp.MakeCall("123-456-7890");  
sp.TakePhoto();  
sp.BrowseInternet();

Expected Output:

Calling 123-456-7890...  
Photo taken with 12MP camera.  
Browsing internet...

# Example 2: Multifunction Printer

A MultiFunctionPrinter can print, scan, and fax documents. It implements IPrinter, IScanner, and IFax interfaces.

Interfaces:

interface IPrinter  
{  
 void Print(string document);  
}  
  
interface IScanner  
{  
 void Scan();  
}  
  
interface IFax  
{  
 void Fax(string number);  
}

Class Implementation:

class MultiFunctionPrinter : IPrinter, IScanner, IFax  
{  
 public void Print(string document)  
 {  
 Console.WriteLine($"Printing: {document}");  
 }  
  
 public void Scan()  
 {  
 Console.WriteLine("Scanning document...");  
 }  
  
 public void Fax(string number)  
 {  
 Console.WriteLine($"Faxing to: {number}");  
 }  
}

Usage:

MultiFunctionPrinter mfp = new MultiFunctionPrinter();  
mfp.Print("Report.pdf");  
mfp.Scan();  
mfp.Fax("987-654-3210");

Expected Output:

Printing: Report.pdf  
Scanning document...  
Faxing to: 987-654-3210

# Example 3: Robot Assistant

A RobotAssistant can walk, talk, and clean. It implements IWalker, ITalker, and ICleaner interfaces.

Interfaces:

interface IWalker  
{  
 void Walk();  
}  
  
interface ITalker  
{  
 void Talk();  
}  
  
interface ICleaner  
{  
 void Clean();  
}

Class Implementation:

class RobotAssistant : IWalker, ITalker, ICleaner  
{  
 public void Walk()  
 {  
 Console.WriteLine("Robot is walking.");  
 }  
  
 public void Talk()  
 {  
 Console.WriteLine("Robot is talking.");  
 }  
  
 public void Clean()  
 {  
 Console.WriteLine("Robot is cleaning.");  
 }  
}

Usage:

RobotAssistant robot = new RobotAssistant();  
robot.Walk();  
robot.Talk();  
robot.Clean();

Expected Output:

Robot is walking.  
Robot is talking.  
Robot is cleaning.