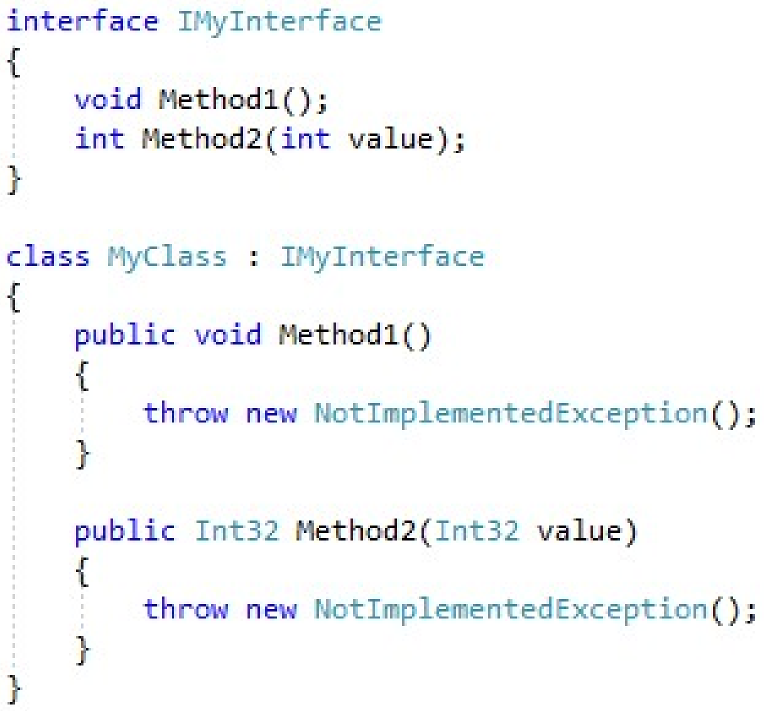
INTERFACES AND COLLECTIONS

INTERFACES

- ***Interfaces*** in the .NET base class libraries are prefixed with a capital letter ***I***.



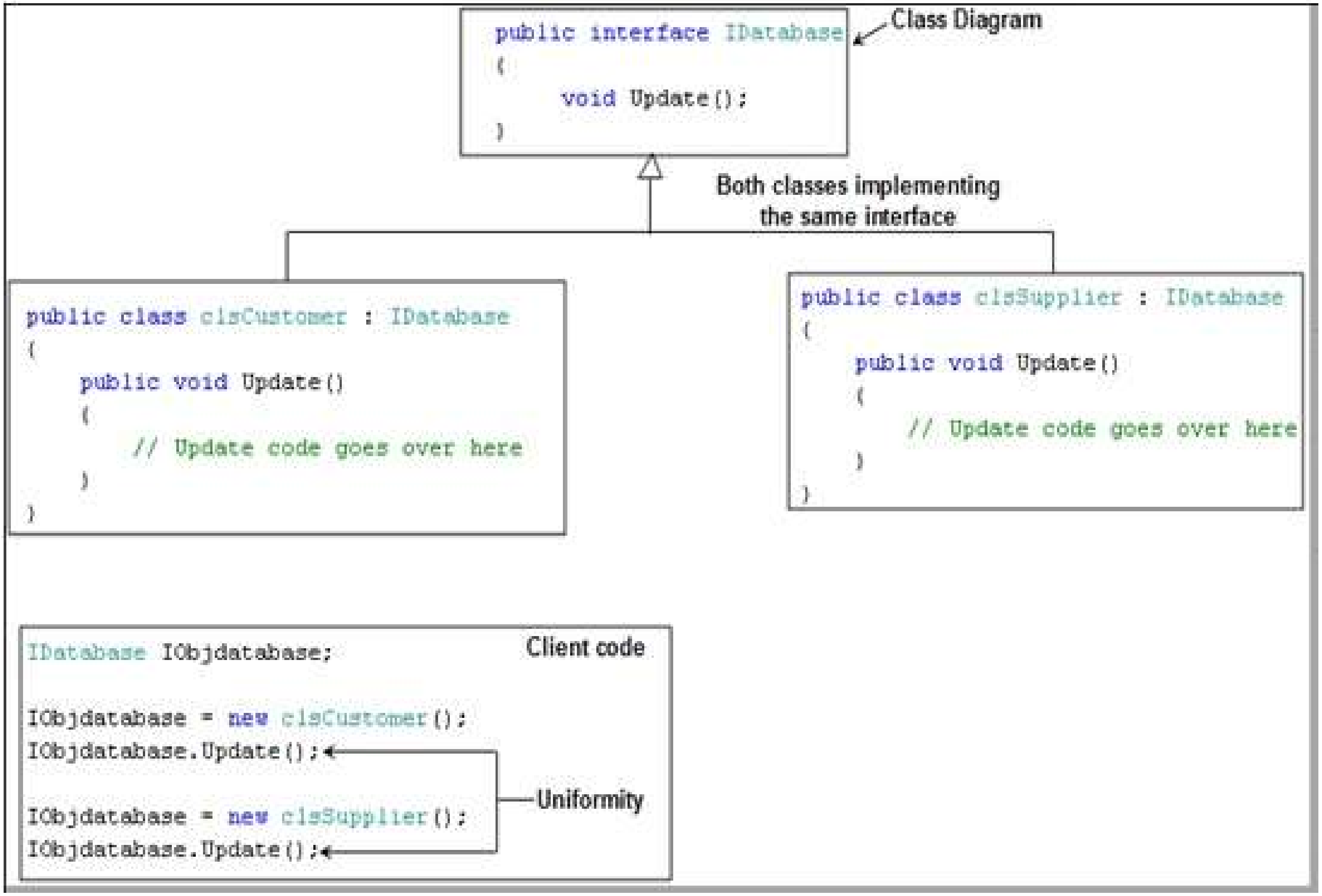
```
interface IMyInterface
{
    void Method1();
    int Method2(int value);
}

class MyClass : IMyInterface
{
    public void Method1()
    {
        throw new NotImplementedException();
    }

    public Int32 Method2(Int32 value)
    {
        throw new NotImplementedException();
    }
}
```

➤ *It is impossible to create an instance of an interface*

IPointy p = new IPointy(); // Compiler error!

INTERFACES

- Example:



```
public interface IDatabase
{
      void Update();
}
```

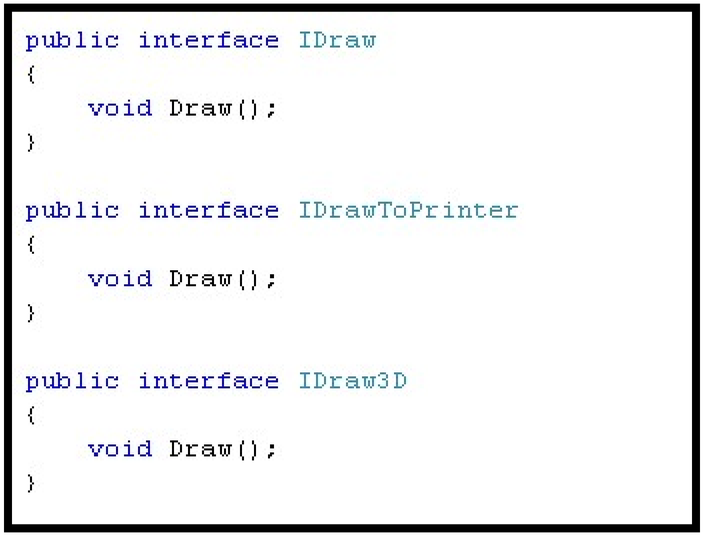
```
public class clsCustomer : IDatabase
{
    public void Update()
    {
        // Update code goes over here
    }
}
```

```
public class clsSupplier : IDatabase
{
    public void Update()
    {
        // Update code goes over here
    }
}
```

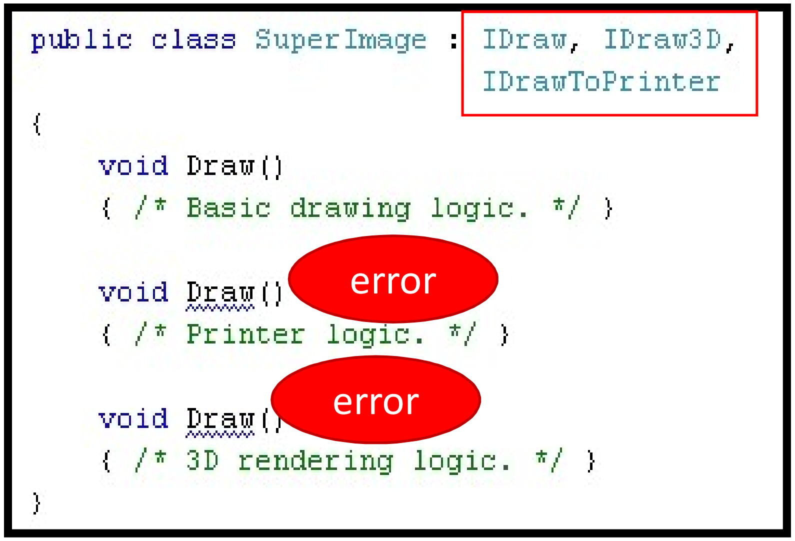
```
IDatabase IObjdatabase;

IObjdatabase = new clsCustomer();
IObjdatabase.Update();

IObjdatabase = new clsSupplier();
IObjdatabase.Update();
```

INTERFACES

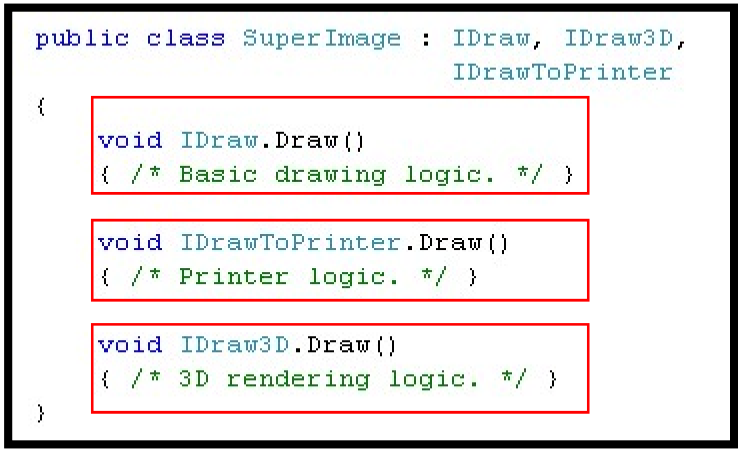
- Resolving Name Clashes:



```
public interface IDraw
{
    void Draw();
}

public interface IDrawToPrinter
{
    void Draw();
}

public interface IDraw3D
{
    void Draw();
}
```



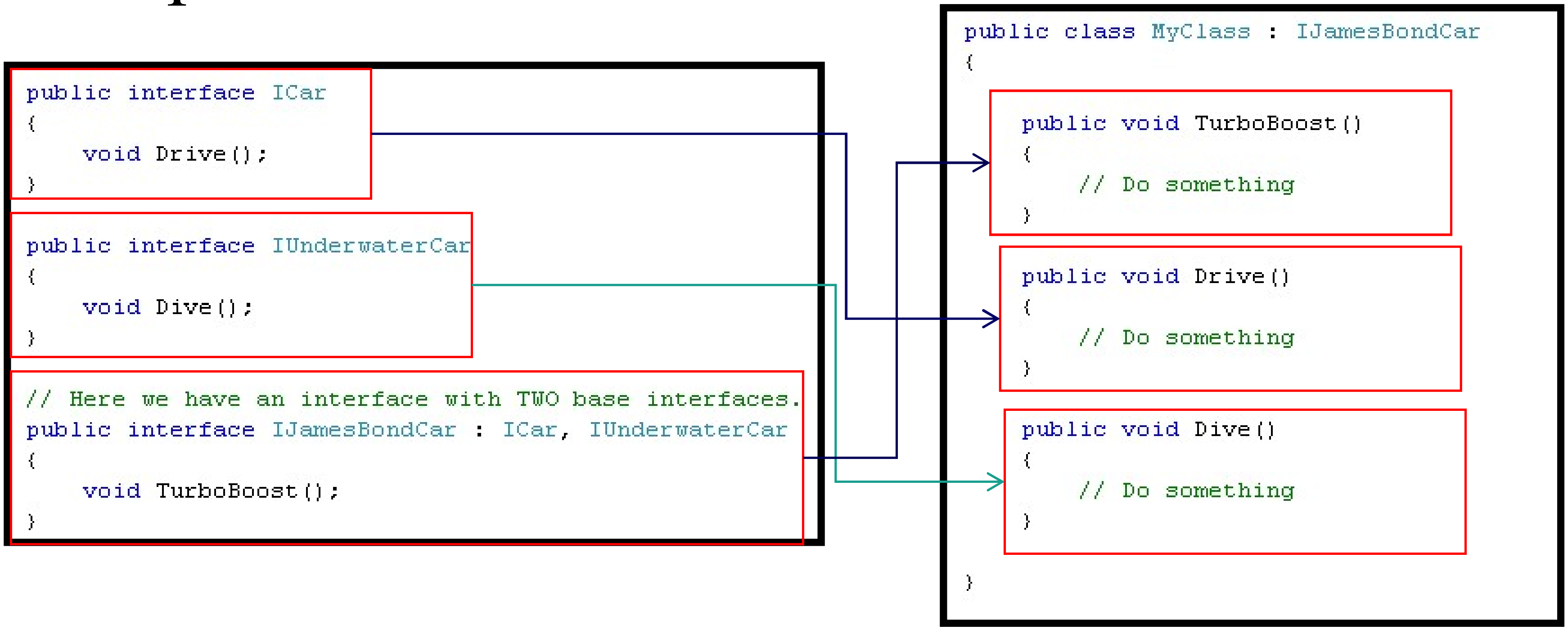
```
public class SuperImage : IDraw, IDraw3D,
                          IDrawToPrinter
{
    void Draw()
    { /* Basic drawing logic. */ }

    void Draw()
    { /* Printer logic. */ }

    void Draw()
    { /* 3D rendering logic. */ }
}
```



```
public class SuperImage : IDraw, IDraw3D,
                          IDrawToPrinter
{
    void IDraw.Draw()
    { /* Basic drawing logic. */ }

    void IDrawToPrinter.Draw()
    { /* Printer logic. */ }

    void IDraw3D.Draw()
    { /* 3D rendering logic. */ }
}
```

INTERFACES

- Multiple Base Interfaces:

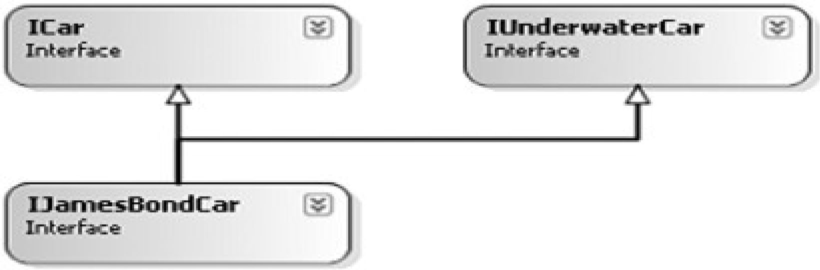


```
public interface ICar
{
    void Drive();
}
```

```
public interface IUnderwaterCar
{
    void Dive();
}
```

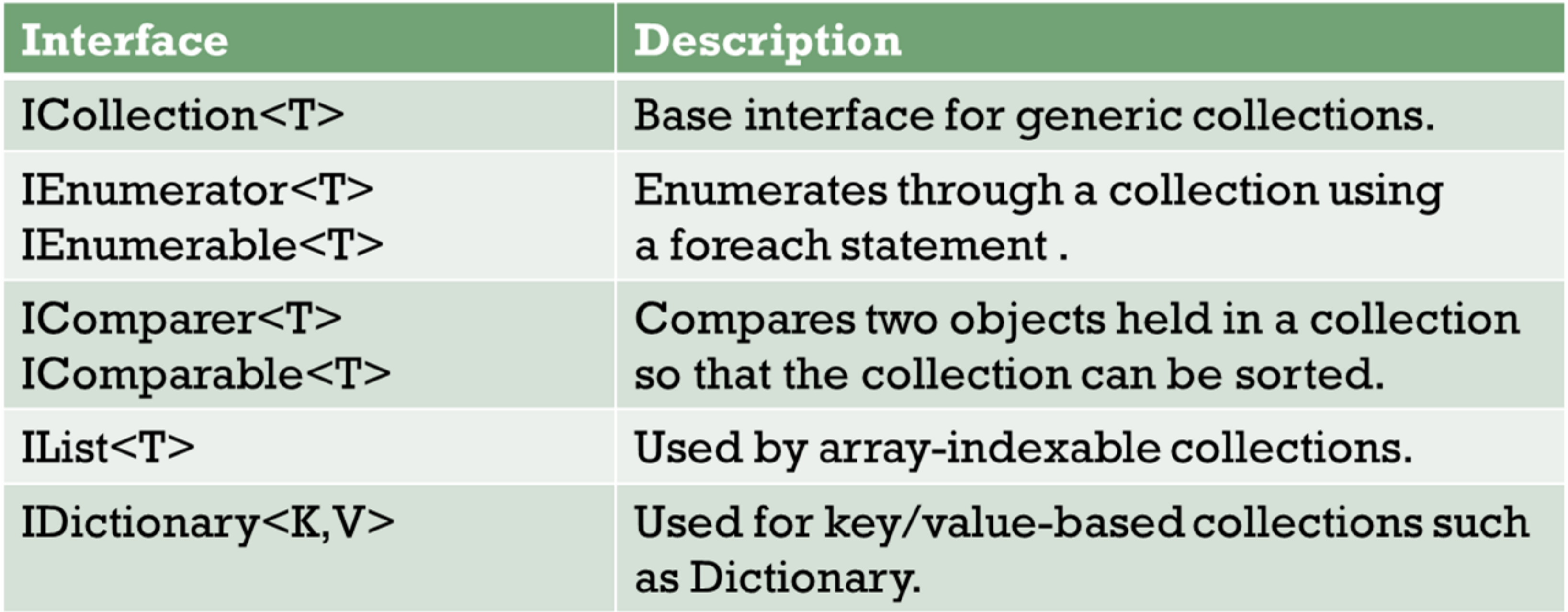
```
// Here we have an interface with TWO base interfaces.
public interface IJamesBondCar : ICar, IUnderwaterCar
{
    void TurboBoost();
}
```

```
public class MyClass : IJamesBondCar
{

    public void TurboBoost()
    {
        // Do something
    }

    public void Drive()
    {
        // Do something
    }

    public void Dive()
    {
        // Do something
    }

}
```



INTERFACES

- Collections Interfaces:



Interface	Description
ICollection<T>	Base interface for generic collections.
IEnumerator<T> IEnumerable<T>	Enumerates through a collection using a foreach statement .
IComparer<T> IComparable<T>	Compares two objects held in a collection so that the collection can be sorted.
IList<T>	Used by array-indexable collections.
IDictionary<K,V>	Used for key/value-based collections such as Dictionary.

Thank You !

IT FACULTY