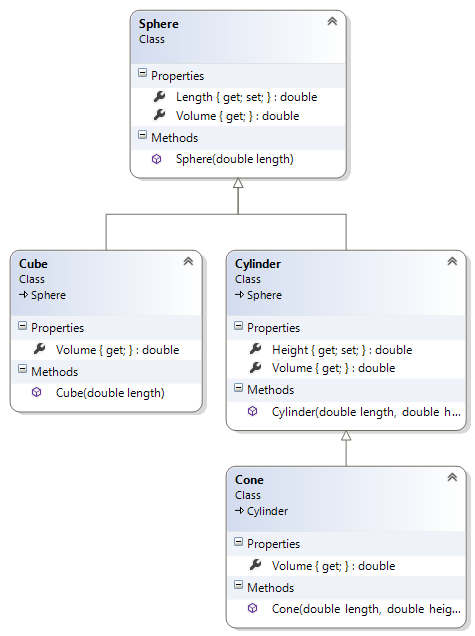
Inheritance allows you to use existing classes to build a new class that will better able to solve your problems.

In this exercise you will be implementing the following inheritance hierarchy. Remember the arrow point to the parent class.

# The Sphere class

This class serves as the base class of our hierarchy.

#### Description of the members:

##### Fields:

There are no fields

##### Properties:

Length: this double auto-implemented property has public read and private write access

Volume: this double property just has a get accessor. It is declared virtual and it returns the volume of this object based on its dimension.

##### Constructor:

There is one constructor that takes a single argument. This argument is assigned to the relevant property.

##### Methods:

There are no methods

# The Cyinder class

This class derives from the Sphere class. It also serves as the base class of the Cone class.

#### Description of the members:

##### Fields:

There are no fields

##### Properties:

Height: this double auto-implemented property has public read and private write access

Volume: this double property replaces the member of the same name in the base class. It returns the volume of this object based on its dimension.

##### Constructor:

There is one constructor that takes two arguments. It invokes the base constructor with the first one and the second one is used to set the property.

##### Methods:

There are no methods

# The Cone class

This class derives from the Cylinder class.

#### Description of the members:

##### Fields:

There are no fields

##### Properties:

No additional properties are defined

Volume: this double property replaces the member of the same name in the base class. It returns the volume of this object based on its dimension.

##### Constructor:

There is one constructor that takes two arguments. It invokes the base constructor with the two arguments.

##### Methods:

There are no methods

# The Cube class

This class derives from the Sphere class.

#### Description of the members:

##### Fields:

There are no fields

##### Properties:

No additional properties are defined

Volume: this double property replaces the member of the same name in the base class. It returns the volume of this object based on its dimension.

##### Constructor:

There is one constructor that takes one argument. It invokes the base constructor with this argument.

##### Methods:

There are no methods

### Test Harness

Insert the following code statements in your Program.cs file:

**List<Sphere> shapes = new List<Sphere>();**

**shapes.Add(new Sphere(2));**

**shapes.Add(new Cylinder(1.5, 2));**

**shapes.Add(new Cone(.75, 1.5));**

**shapes.Add(new Cube(1.2));**

**foreach (Sphere shape in shapes)**

**{**

**Console.WriteLine("{0:f2}", shape.Volume);**

**}**

### Additional tasks

Add a method public double GetMass(double density) to the Sphere class to return the product of the argument and the volume

Insert the proper code statements in your main to show the operation of this new method.

N.B. Although the method was implemented in a different class, you are still able to use the logic. This is the power of inheritance