[0610] Inheritance and Inheritance Terminology

Inheritance is the concept of creating more complex objects or classes from simpler, related objects or classes. For example, you could use the concept of inheritance to create an "Eagle" from the simpler "Bird" class. If you did that, then the Eagle class would automatically have all of the member variables, properties and methods that the Bird class has.

There are lots of different terms related to the concept of Inheritance that you need to become familiar with:

- The simpler class ("Bird") is called the **parent class**. The more complex class ("Eagle") is called the **subclass**.
- Subclasses are said to "**inherit from**" their parent classes.
- If a parent class doesn't inherit from an even simpler parent class, it is called a **base class**.
- Parent classes are also called **super classes** and/o**r base classes** (regardless of whether they have another parent class).
- Subclasses are also called: **child classes, derived class, sub-types**.
- **Single Inheritance** is where a subclass inherits from only one superclass. **Multiple Inheritance** is where a subclass inherits from two or more superclasses. C# (and Java) only support single inheritance. (C++ supports multiple-inheritance.)
- **Implementation Inheritance** is a more specific term for when subclasses are able to reuse code contained in their superclasses.
- **Interface Inheritance** is a term for when subclasses only inherit the type characteristics (i.e, the public APIs) of their superclasses.