[0699] Chapter 6: Summary

- Inheritance lets you create a tree-like hierarchy of related classes. The lower classes are said to "inherit from" the upper classes.
- A subclass automatically gets ("inherits") all of the non-private member variables and methods that all of its superclasses have.
- C# does not support multiple inheritance. Subclasses can only have one superclass.
- If a method can be overridden, add the "virtual" keyword to its method signature.
- If a method in a subclass is overriding a method in a superclass, add the "override" keyword to its method signature.
- A "sealed" class cannot be used for inheritance. You can not inherit from a sealed class.
- An "abstract" class is essentially an "incomplete class" because it has one or more incomplete "abstract" methods.
- Abstract classes cannot be instantiated. Abstract classes must have a subclass inherit from them and implement their abstract methods.
- Polymorphism lets you call the same method on two different related objects. The system determines at runtime which method to actually call based on the type of the object calling the method.
- System.Object is the superclass of everything in C#. It's methods includes: ToString(), Equals(), and MemberwiseClone() which means that everything in C# has those methods.