[601] Instance Methods vs Class Methods

Instance methods are non-static, normal methods that require an instance of a class (i.e., an object) in order to work. They will often be called via the "variableName.MethodName()" format - ex. "deck.Shuffle();" or "cup.Roll();" - where the variable contains the instance of the class and MethodName() is the (typically public) method that then runs. In a sense, the variable is a "hidden" parameter of the method call.

Class methods are static, non-typical "helper" methods that do not need an instance of a class in order to do their job. Class methods are called via the "ClassName.MethodName()" format - ex. "Console.WriteLine();" or "Int32.TryParse();" Notice that the first part of those calls is CAPITALIZED - which is the clue that we are dealing with a Class method call.

In the real world, Class methods are relatively rare however we have been using them relatively frequently up to this point. DON'T GET CONFUSED!   We will be using fewer and fewer static Class methods from this point onwards. Object-oriented programming is all about using Instance methods on objects.