As you build apps, you’ll find you create code that is for general use. Build a component that’s a general library for resuse.

We’ve already seen this with layers.

Making general components is reuse

To build a reusable component, start with a new project. In C#, class library. Give it a general name.

String.Empty – gives you an initialised empty string.

String.IsNullOrWhiteSpace(theString)

Foreach ( char letter in theString)

Char.IsUpper()

Make sure reuseable components are well tested. Test projects don’t live with the component. Separate unit test project.

String.Trim() – takes away white space from beginning and end of string.

To use a component, reference it. References->add reference. Reference the component.

One off functionality should be put in a static class. static keyword. Static class can’t be instantiated.

An instance can hold state. If a class doesn’t have object level state, use static class.

Once a class is static, every member has to be made static.

Use static classes sparingly. Business level classes need instances to manage states.

Extension methods:

An extension method is a method that becomes a method of a class in which it is NOT defined. To accomplish this

Public static string InsertSpaces(**this** string source)

The use of the **this** keyword makes the InsertSpaces method a method of a string object.

An extension method allows the addition of methods to any existing type by using the this keyword in the parameter of the method without any changes at all to the original type.

It is then called like an instance method on an instance of the extended class.

An extension method **MUST** reside in a static class.

Extension methods become discoverable as a member of the class.

Static vs Extension:

Extension: Is the primary parameter an instance? Does the method operate on the instance? Appear in intellisense?