**Lesson 2 Notes**

**Assembly –** a deployment unit. Represents one file on disk. If you have 4 projects in your solution, there will be 4 assemblys.

**Namespace**

You can have same types within different assemblies. Problem is when you try to use two different assemblies with the same type. To limit the possibility of this, you use namespaces. It’s sole purpose is for scoping.

A type is unique as far as it’s fully qualified name is unique. Fully qualified name is namespace + type name. The compiler always works with fully qualified namespaces.

Namespaces are always pascal cased. They follow identifier rules.

The defacto standard is: company name.product[.subnamespaces]

Company types are included in company namespace. Product types are in the product namespace.

Generally, the namespace name follows the assembly name.

The compiler always required the fully qualified type name. The compiler has a shortcut so you don’t have to use the FQTN every time.

Using name;

Now you can just use ‘product’ in your code.

Example. Int32 is not the FQTN. It is actually System.Int32. So you would put ‘using System’ a the top of the page. They are USING DECLARATIONS.

Using Keyword followed by Namespace name;

Namespace Nile.Host (contains two namespaces… Nile… and sub namespace Host).

To create a namespace, you create a namespace statement.

Namespace ‘name of namespace’

{

}

You can also nest namespaces; but, it isn’t advisable.

Projects talk to each other through assemblies