

Boxing and UnBoxing

C# Programming by: Christopher Singleton

This is a simple illustration of how boxing and unboxing works and please also please note that I am accessing the method to display text using an object with a variable: **MyObjectName** to reference another object (**MyClass**). Note that the instantiation of the object using the key word “new” is of type **MyClass**, which is the same of the class name.

Class “Program”

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BoxingUnboxing
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create a new object as "MyClass" using the new keyword and variable name to access the Class.
            MyClass MyObjectName = new MyClass();

            // Call "MyMethod" by including the Variable name of the Object.
            MyObjectName.MyMethod();
        }
    }
}
```

Class "MyClass"

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BoxingUnboxing
{
    /***** BOXING AND UNBOXING *****/
    *Boxing means when you pass information from a value type to a reference type. *
    *UnBoxing is when you pass a information from a reference type to a value type. *
    *****/
    /*
    class MyClass
    {
        public void MyMethod()
        {
            // Boxing (Value Type to Reference Type). The value is passed from the variable, then to the object.
            int Value = 1;
            Object Obj = Value;

            // UnBoxing (Reference Type to Value Type). The value is passed from the object to i.
            int i = (int)Obj;

            // Write out our value.
            Console.WriteLine("The value that I have passed from Boxing to UnBoxing is: {0}", i);

            // Pause the Console.
            Console.ReadKey();
        }
    }
}
```