Steven Myers

30 October 2016

CSCI-A290 Windows Programming

1. **Variable**

**Links:** <https://msdn.microsoft.com/en-us/library/ms173104.aspx>, <https://www.tutorialspoint.com/csharp/csharp_variables.htm>, <http://csharp.net-tutorials.com/basics/variables/>

**Precis:**

”There are several basic value types provided in C#” including integral, floating point, decimal, Boolean, and nullable types. ­The correct syntax for declaring variables in C# is “<data\_type> <variable\_list>;”. You can then set a variable name equal to a value. You can also set the visibility of variables with “private” to clarify that the variable may only be visible within the scope of the current class.

**Summary:**

I think the most important thing about the articles regarding variables is that you should be careful about distinguishing between private and public variables. If you have information that shouldn’t be accessible to other parts of the program, you should use private variables when possible.

2. **Type**

**Links:** <https://msdn.microsoft.com/en-us/library/ya5y69ds.aspx>, <https://www.tutorialspoint.com/csharp/csharp_type_conversion.htm>,

<https://www.tutorialspoint.com/csharp/csharp_data_types.htm>

**Precis:**

“Variables in C# are categorized” into the following 3 type categories: value, reference, and pointer types. When you declare a variable a certain type, “the system allocates memory to store the value.” The following types are the ones available to C#: (bool, byte, sbyte, char, decimal, double, float, int, uint, long, ulong, object, short, ushort, string). You can convert data types to other data types. There are two forms of type conversion – “implicit type conversion” and “explicit type conversion.” This is also known as “type casting.” Most data types have built in conversion methods, like “ToBoolean, ToByte, toString.”

**Summary:**

I think understanding type conversions is very important, since there will almost always be a scenario where you need to at least convert a number to a string, or a string to a number. For example, if we are trying to calculate and display a date given a string of the current date, it’s more than likely we’re going to need to parse that string and convert it to an integer expression. It’s also incredibly important to understand what operations you can perform on certain data types. For example, you can’t do mathematical addition on two strings.

3. **Procedure**

**Links:** <http://www.c-sharpcorner.com/uploadfile/skumaar_mca/differences-between-procedures-and-functions/>,

**Precis:**

“Procedures may or may not return values, but functions should always return values.”

**Summary:**

4. **Method**

**Links:**

**Precis:**

**Summary:**

5. **Event**

**Links:**

**Precis:**

**Summary:**