

C# does an *implicit converstion* when multiplying an int by a double. The answer will be converted to a double and the function will happen error free; however;

int result = 1.5+1;

will not compile because result is not an integer. To make this work, you would put (int) in parenthesis ahead of the 1.5 like:

int result = (int) 1.5 + 1;

The result will be 2 because C# truncates the 1.5. Only the next term after (int) is converted, ie:

(int)6.7+1 = 6+1= 7 is not equal to 6.7+(int)1 = 6.7+1 = 7.7

(int) (6.7+1) = (int) 7.7 = 7

All the variable type in C# follow from (http://msdn.microsoft.com/en-us/library/ya5y69ds(v=vs.110).aspx)

# Built-In Types Table (C# Reference)

* 07/11/2012
* 2 minutes to read

The following table shows the keywords for built-in C# types, which are aliases of predefined types in the [System](https://msdn.microsoft.com/en-us/library/yxcx7skw(v=vs.110)) namespace.

| **Table 1** | |
| --- | --- |
| **C# Type** | **.NET Framework Type** |
| [bool](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/c8f5xwh7%28v%3dvs.110%29) | System.Boolean |
| [byte](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/5bdb6693%28v%3dvs.110%29) | System.Byte |
| [sbyte](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/d86he86x%28v%3dvs.110%29) | System.SByte |
| [char](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/x9h8tsay%28v%3dvs.110%29) | System.Char |
| [decimal](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/364x0z75%28v%3dvs.110%29) | System.Decimal |
| [double](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/678hzkk9%28v%3dvs.110%29) | System.Double |
| [float](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/b1e65aza%28v%3dvs.110%29) | System.Single |
| [int](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/5kzh1b5w%28v%3dvs.110%29) | System.Int32 |
| [uint](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/x0sksh43%28v%3dvs.110%29) | System.UInt32 |
| [long](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/ctetwysk%28v%3dvs.110%29) | System.Int64 |
| [ulong](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/t98873t4%28v%3dvs.110%29) | System.UInt64 |
| [object](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/9kkx3h3c%28v%3dvs.110%29) | System.Object |
| [short](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/ybs77ex4%28v%3dvs.110%29) | System.Int16 |
| [ushort](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/cbf1574z%28v%3dvs.110%29) | System.UInt16 |
| [string](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/362314fe%28v%3dvs.110%29) | System.String |

## Remarks

All of the types in the table, except object and string, are referred to as simple types.

The C# type keywords and their aliases are interchangeable. For example, you can declare an integer variable by using either of the following declarations:

Copy

int x = 123;

System.Int32 x = 123;

To display the actual type for any C# type, use the system method GetType(). For example, the following statement displays the system alias that represents the type of myVariable:

Copy

Console.WriteLine(myVariable.GetType());

You can also use the [typeof](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/58918ffs%28v%3dvs.110%29) operator.

## See Also

#### Reference

[C# Keywords](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/x53a06bb%28v%3dvs.110%29)

[Value Types (C# Reference)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/s1ax56ch%28v%3dvs.110%29)

[dynamic (C# Reference)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/dd264741%28v%3dvs.110%29)

#### Concepts

[C# Programming Guide](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/67ef8sbd%28v%3dvs.110%29)

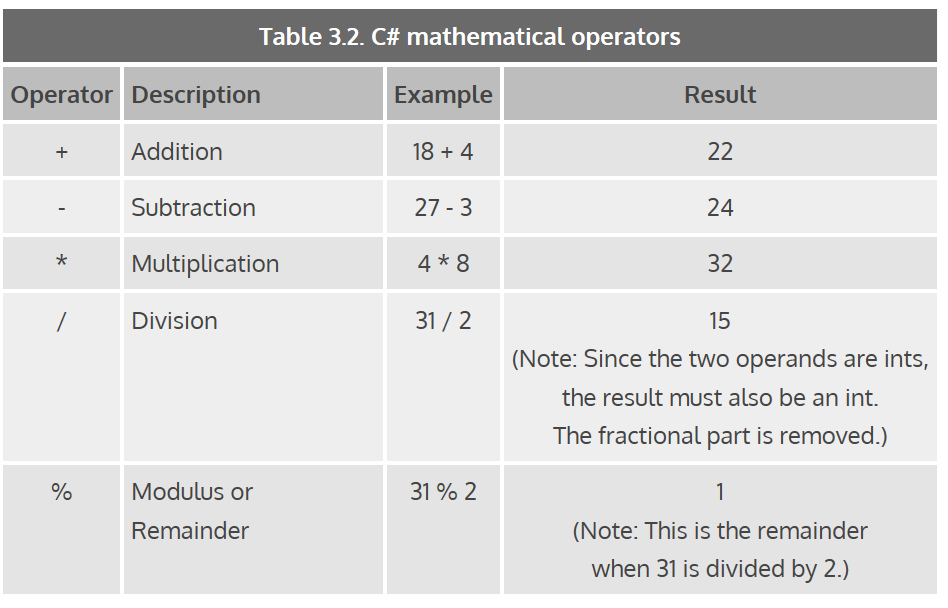
#### Other Resources

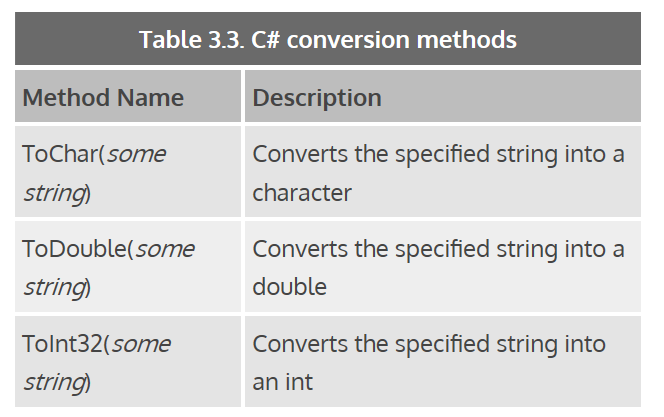
[C# Reference](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/618ayhy6%28v%3dvs.110%29)

[Default Values Table (C# Reference)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/83fhsxwc%28v%3dvs.110%29)

[Formatting Numeric Results Table (C# Reference)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/s8s7t687%28v%3dvs.110%29)

[Reference Tables for Types (C# Reference)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/1dhd7f2x%28v%3dvs.110%29)





Q: How can I fix it so that when I print out my cost, it shows two decimal places instead of one. That is 123.40, instead of 123.4.

**A:** There are many ways to format output in C#. One way is to use the placeholder method that is discussed in Lesson 03, Chapter 03 and include a formatting code along with the placeholder number. For example, if you want to format the value of the cost variable as currency, you can use the currency code "C", as in:   
  
System.Console.Out.WriteLine("The price is {0:C}", cost);   
  
There are many different formatting codes, which can be found at the MSDN.com site at: