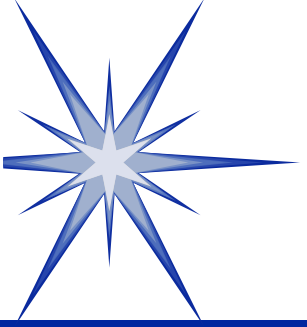


Numeric Data Types in C#.NET

by Derek Peacock

Identifiers in C#

- Must begin with a letter (or underscore)
- Can contain letters, digits, underscore
- Cannot be a C# Keyword (e.g. return)
- Should start with lower case letter
- Every subsequent word should begin with capital letter (camel case)
- Do not use abbreviations

Adding Methods

```
/// <summary>
/// A simple method to ask the user to enter an
/// amount in pounds (GBP) and convert it to
/// Euros (EUR) using the current exchange rate
/// </summary>
static void ConvertToEuros()
{
    const decimal ExchangeRate = 1.24710M;

    decimal pounds; // GBP
    decimal euros;  // EUR

    Console.Write("Enter amount in british pounds >");
    string value = Console.ReadLine();

    pounds = Convert.ToDecimal(value);
    euros = pounds * ExchangeRate;

    Console.WriteLine("Euros = {0}", euros);
    Console.WriteLine();
}
```

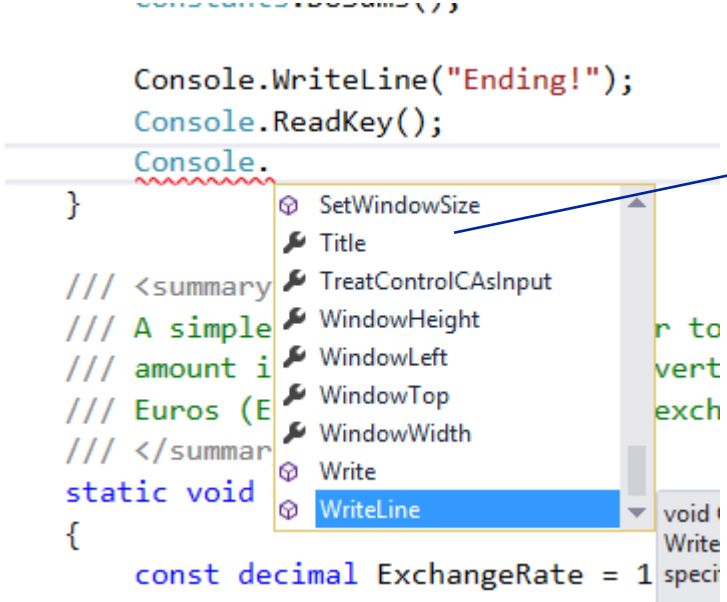
Comments

Name of Method

Start of method

End of Method

Auto Code Completion



List of available properties (spanner) and methods (cube) of the “Console” class

Declaring Constants

```
public const DataType CONSTNAME = expression;

public const int MAX_CLASS_SIZE = 20;
private const double PI = 3.14159;

public const int NO_DAYS_IN_WEEK = 7;
public const string COLLEGE = “Leicester”;
```

Declaring Variables

```
private TypeName variableName [= expression];

private int RoomLength;
private double WallArea;

private int StartYear = 2014;
private double TuitionFees = 9150.00;
```

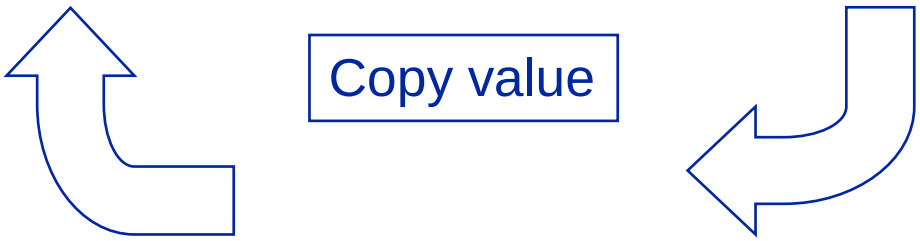
(Using un-initialised variables is a compile error!)

Primitive Data Types

bool	2 bytes		true or false
byte	1 byte		0..255
short	2 bytes		-32768..+32767
int	4 bytes		-2,147,483,648..+2,147,483,647
long	8 bytes		$-10^{20}..+10^{20}$
decimal	8 bytes		28 significant digits
double	8 bytes		14 significant digits
float	4 bytes		$-10^{45}..+10^{38}$

Assignment Statements

```
RoomLength = measuredLength;
```



Arithmetic Expressions

```
RoomArea = RoomLength * RoomWidth;

Result1 = 20 + 10 * 3;
Result2 = (20 + 10) * 3;

Total = 50;
Total = Total + 30;
```

Operator Precedence

()	Inside before outside	(2 + (6 – 4))
* /	Multiply and divide, Left to right	3 * 10 / 5 = 6
%	Remainder after division	7 % 2.5 = 2
+ -	Add and subtract	2 + 3 = 5

Quiz - Integer Expressions

Expression	**Value**
2 + 3 * 4 + 5	??
(2 + 3) * (4 + 5)	??
17 - 10 - 3	??
17 - (10 - 3)	??
(17 - 10) - 3	??
8 % 3 - 15 * (5 + 1)	??

Type Conversion

- Implicit conversion
  - double result = 10 / 3;
- Explicit conversion
  - int number = Convert.ToInt(“12”);
  - double value = Convert.ToDouble(“12.5”);
- Divide by Zero
  - Produces infinity or NaN (not a number!)

Formatting Numeric Values

```
euros.ToString("Euros = {C}");
euros.ToString("Euros = {0:C}");
euros.ToString("Euros = {0, 8:C}");
euros.ToString("Euros = {0, 6:C2}");

Console.WriteLine(
  "Euros = {0} Rate = {1}", euros, rate);
```

Format Characters

0, 1	A placeholder for a variable
C	A numeric value as currency £4321.50
D	Integer value 4321
F	Fixed point decimal 4321.50
P	Percentage notation 432,100%
N	Numeric 4,321
E	Scientific notation 4.321E+003

Maths Methods

- Math.Sin()
- Math.Tan()
- Math.Cos()
- Math.Sqrt()
- Math.Abs()
- Math.Round()

System Output

Console.WriteLine(“Length = {0}“, Length)

Outputs to the system console window (Debug window)

References

- “Microsoft Visual C# Step by Step”
  by John Sharp (2014)
  Microsoft Press £19 (£14 Kindle)
- Chapter 2