

# VB.NET - OPERATORS PRECEDENCE

[http://www.tutorialspoint.com/vb.net/vb.net\\_operators\\_precedence.htm](http://www.tutorialspoint.com/vb.net/vb.net_operators_precedence.htm)

Copyright © tutorialspoint.com

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator:

For example,  $x = 7 + 3 * 2$ ; here,  $x$  is assigned 13, not 20 because operator  $*$  has higher precedence than  $+$ , so it first get multiplied with  $3*2$  and then adds into 7.

Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Operator	Precedence
Await	Highest
Exponentiation $\wedge$	
Unary identity and negation $+$ , $-$	
Multiplication and floating-point division $*$ , $/$	
Integer division ( $\backslash$ )	
Modulus arithmetic <i>Mod</i>	
Addition and subtraction $+$ , $-$	
Arithmetic bit shift $<<$ , $>>$	
All comparison operators $=$ , $<>$ , $<$ , $<=$ , $>$ , $>=$ , <i>Is</i> , <i>IsNot</i> , <i>Like</i> , <i>TypeOf</i> ... <i>Is</i>	
Negation <i>Not</i>	
Conjunction <i>And</i> , <i>AndAlso</i>	Lowest
Inclusive disjunction <i>Or</i> , <i>OrElse</i>	
Exclusive disjunction <i>Xor</i>	

## Example:

The following example demonstrates operator precedence in a simple way:

```
Module assignment
    Sub Main()
        Dim a As Integer = 20
        Dim b As Integer = 10
        Dim c As Integer = 15
        Dim d As Integer = 5
        Dim e As Integer
        e = (a + b) * c / d      ' ( 30 * 15 ) / 5
        Console.WriteLine("Value of (a + b) * c / d is : {0}", e)
        e = ((a + b) * c) / d   ' (30 * 15 ) / 5
        Console.WriteLine("Value of ((a + b) * c) / d is : {0}", e)
        e = (a + b) * (c / d)   ' (30) * (15/5)
        Console.WriteLine("Value of (a + b) * (c / d) is : {0}", e)
        e = a + (b * c) / d     ' 20 + (150/5)
        Console.WriteLine("Value of a + (b * c) / d is : {0}", e)
        Console.ReadLine()
    End Sub
```

End Module

When the above code is compiled and executed, it produces the following result:

```
Value of (a + b) * c / d is : 90  
Value of ((a + b) * c) / d is : 90  
Value of (a + b) * (c / d) is : 90  
Value of a + (b * c) / d is : 50
```

Loading [Mathjax]/jax/output/HTML-CSS/jax.js