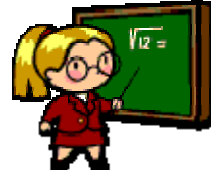


MATHEMATICAL OPERATORS

MATHEMATICAL OPERATORS

In order to compute inputs from users and generate results, we need to use various mathematical operators. In Visual Basic, other than the addition (+) and subtraction (-), the symbols for the operators are different from normal mathematical operators, as shown in the table below:



OPERATOR	MATHEMATICAL FUNCTION	EXAMPLE
^	Exponents	$2 \wedge 4 = 16$
*	Multiplication	$4 * 3 = 12$
/	Division	$12 / 4 = 3$
+	Addition	$7 + 5 = 12$
-	Subtraction	$7 - 5 = 2$
Mod	Modulus (returns the remainder from an integer division)	$15 \text{ Mod } 4 = 3$
\	Integer Division (discards the decimal places)	$19 \setminus 4 = 4$
+ or &	String concatenation	"TIK" & "201" = "TIK201"

ORDER OF OPERATIONS

Visual Basic evaluates a numeric expression using a specific order of operations, or operator precedence. The following table outlines the order of operations used in Visual Basic:

ORDER	OPERATOR	FUNCTION
1 st	^	Exponents
2 nd	+, -	Unary
3 rd	*, /	Multiplication and Division
4 th	\	Integer Division
5 th	Mod	Modulus
6 th	+, -	Addition and Subtraction

Operator precedence can, of course, be changed by including parentheses in a numeric expression. The operations within parentheses are evaluated first.

COMBINED ASSIGNMENT OPERATORS

Sometimes you might find it necessary to change the value of a variable by doubling it or subtracting a certain amount from it while keeping the result in the variable itself. **Combined assignment operators** (sometimes called **compound operators**) are designed to do just that. Without compound operators, you would have to type the variable name twice – once for the left portion of the equation and again in the right portion of the equation.

The following table outlines the compound operators available in Visual Basic:

COMPOUND OPERATOR	EXAMPLE	EQUIVALENT
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
\=	x \= 5	x = x \ 5
&=	name &= lastName	name = name & lastName