Operators in VBA and operator precedence

Objectives : At the end of this lesson you shall be able to
• **explain the various operators and their precedence in VBA.**

Operators in VBA

An **Operator** can be defined using a simple expression - 4 + 5 is equal to 9. Here, 4 and 5 are called **operands** and + is called **operator**. VBA supports following types of operators "

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Concatenation Operators

The Arithmetic Operators

Following arithmetic operators are supported by VBA.

Assume variable A holds 5 and variable B holds 10, the results of the various operators as shown in Table 1.

Table 1

Operator	Description	Example
+	Adds the two operands	A + B will give 15
-	Subtracts the second operand from the first	A - B will give -5
*	Multiplies both the operands	A * B will give 50
/	Divides the numerator by the denominator	B / A will give 2
%	Modulus operator and the remainder after an integer division	B % A will give 0
^	Exponentiation operator	B ^ A will give 100000

The Comparison Operators

There are following comparison operators supported by VBA.

Table 1 Assume variable A holds 10 and variable B holds 20, the results of various comparison operators as shown

Table 2

Operator	Description	Example
=	Checks if the value of the two operands are equal or not. If yes, then the condition is true	(A = B) is False
<>	Checks if the value of the two operands are equal or not. If the values are not equal, then the condition is true	(A <> B) is True
>	Checks if the value of the left operand is greater than the value of the right operand. If yes, then the condition is true	(A > B) is False
<	Checks if the value of the left operand is less than the value of the right operand. If yes, then the condition is true	(A < B) is True
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand. If yes, then the condition is true	(A >= B) is False
<=	Checks if the value of the left operand is less than or equal to the value of the right operand. If yes, then the condition is true	(A <= B) is True

The Logical Operators

Following logical operators are supported by VBA.

Assume variable A holds 10 and variable B holds 0, the results on the various logical operators shown in Table 3

Table 3

Operator	Description	Example
AND	Called Logical AND operator. If both the conditions are True, then the Expression is true	a<>0 AND b<>0 is False
OR	Called Logical OR Operator. If any of the two conditions are True, then the condition is true	a<>0 OR b<>0 is true
NOT	Called Logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make false	NOT(a<>0 OR b<>0) is false
XOR	Called Logical Exclusion. It is the combination of NOT and OR Operator. If one, and only one, of the expressions evaluates to be True, the result is True.	(a<>0 XOR b<>0) is true

The Concatenation Operators

Following Concatenation operators are supported by VBA.

Assume variable A holds 5 and variable B holds 10, the result of various concatenation operators shown in Table 4

Table 4

Operator	Description	Example
+	Adds two Values as Variable. Values are Numeric	A + B will give 15
&	Concatenates two Values	A & B will give 510

Assume variable A = “Microsoft” and variable B = “VBScript”, the result of the various concatenation shown in Table 5

Table 5

Operator	Description	Example
+	Concatenates two Values	A + B will give MicrosoftVBScript
&	Concatenates two Values	A & B will give MicrosoftVBScript

Note: Concatenation Operators can be used for both numbers and strings. The output depends on the context, if the variables hold numeric value or string value.

Precedence

When several operations occur in an expression, each part is evaluated and resolved in a predetermined order called operator precedence.

When operators have the same precedence they are evaluated from left-to-right.Parentheses can be used to override the order and to evaluate certain parts of the expression. Operations inside parentheses are always performed before those outside.

When a series of operators appear in the same expression there is a strict order in which they will be evaluated.

The rules of precedence tell the compiler which operators to evaluate first.

Parentheses can obviously be used to change the order of precedence.

Operators are evaluated in the following order: Mathematical, Concatenation, Relational, Logical.

The table 6 shows the precedence order of operators.

Table 6

Order	Operator	Symbol
1	Exponentiation	^
2	Negation	-
3	Multiplication	*
3	Division	/
4	Division with Integer result	\
5	Modulo	Mod
6	Addition	+
6	Subtraction	-
7	String Concatenation	&
8	Equal or Assignment	=
8	Not Equal To	<>
8	Less Than	<
8	Greater Than	>
8	Less Than or Equal To	<=
8	Greater Than or Equal To	>=
9	Not	NOT
10	And	AND
11	Or	OR
12	Exclusive OR	XOR
13	Equivalence	EQV
14	Implication	IMP

The table 7 shows the expression, steps to evaluate and the result.

Expression	First Step	Second Step	Third Step	Result
3^(15/5)*2-5	3^3*2-5	27*2-5	54-5	49
3^((15/5)*2-5)	3^(3*2-5)	3^(6-5)	3^1	3
3^(15/(5*2-5))	3^(15/(10-5))	3^(15/5)	3^3	27