

Operators

Tuesday, September 9, 2025 9:23 AM

Operators

Operators are used to compute values. The different types of operators include assignment operators, arithmetic operators, and relational operators. Assignment operators can change the value of the variable or field.

Operator	Description	Example
=	Assigns the value after the equal sign to the variable before.	Int i = 1;
+=	Assigns the current variable value plus the value after the equal sign.	int l ; l += 1;
++	Increments the variable by 1.	int l; i++;
-=	Assigns the current variable value minus the expression after the equal sign.	int l; l -= 1;
--	Decrements the variable by 1.	int l; i--;

Arithmetic operators are used to make numeric calculations.

From <<https://learn.microsoft.com/en-us/training/modules/get-started-xpp-finance-operations/2-base-types>>

Operator	Description	Example
<<	The left shift operator multiplies the left value by 2 by the number of times equal to the value on the right.	int a = 1 << 2 (this equals 1*2*2.)
>>	The right shift operator divides the left value by 2 by the number of times equal to the value on the right.	int a = 1 >> 2 (this equals 1/2/2.)
*	This operator multiplies operands.	int a = 1*2
/	This operator divides the left value by the right value.	int a = 1/2
div	The integer division operator performs an integer division of the left value and the right value.	int a = 20 div 6 (This returns a division of 3 with a remainder of 2. A = 3)
mod	The integer remainder operator returns the remainder of an integer of the left value and the right value.	int a = 20 mod 6 (This returns a division of 3 with a remainder of 2. A = 2)
~	The not operator performs a binary not operation.	int a = ~1 (This returns -2.)
&	The binary AND operator performs a binary operation on the left and right value. This returns the bits in common.	int a = 1 & 3 (This returns 1.)
^	The binary XOR operator performs a binary XOR operation on the left and right value. This returns the bits set in one, but not the other.	int a = 1 ^ 3 (This returns 2.)
	The binary OR operator performs a binary operation on the left and right value. This returns the bits set in either.	int a = 1 3 (This returns 3.)
+	The plus operator adds operands.	int a = 1 + 4 (This returns 5.)
-	The minus operator subtracts the left or right values.	int a = 4 - 1 (This returns 3.)
?	The ternary operator takes three values. If value 1 is true, value 2 is	int a = (x > 3) ? 1:5 (If x is greater than

returned. Else, value 3 is returned.

3, then return 1, else return 5.)

Relational operators are used in X++ to compare the values between variable values of similar types. Some limitations with complex data types such as GUIDs apply.

Operator	Description
Like	This operator is exclusively used in SQL statements and returns true if the criteria are found within the second value. For example, select a table where a field such as '12*' returns all rows from a table where the field values start with 12.
==	Returns true if the values are equal.
!=	Returns true if the first value isn't equal to the second value.
>=	Returns true if the first value is greater than or equal to the second value.
<=	Returns true if the first value is less than or equal to the second value.
>	Returns true if the first value is greater than the second value.
<	Returns true if the first value is less than the second value.
&&	Returns true if the first and second value are true.
	Returns true if the first or second value is true.
!	Returns true if the value is false, and it returns false if the value is true.

A ternary operator allows you to evaluate if an expression is true or false to get a result. This is like an if statement, which is discussed in the Use conditional and iterative statements and Exception handling units. However, where an if statement provides conditional branching for the code that is being run, the ternary operator result can be assigned to a variable. The following example checks if the variable x is equal to 5. If it's equal to 5, then the ternary operator assigns the value Yes to the result variable. If x doesn't equal 5, the ternary operator assigns the value No to the result variable.

From <<https://learn.microsoft.com/en-us/training/modules/get-started-xpp-finance-operations/2-base-types>>

```
result = (x==5) ? "Yes" : "No";
```