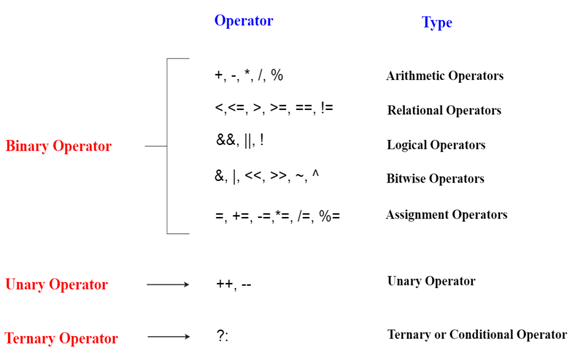
**C# operators**

An operator is simply a symbol that is used to perform operations. There can be many types of operations like arithmetic, logical, bitwise etc.

There are following types of operators to perform different types of operations in C# language.

* Arithmetic Operators
* Relational Operators
* Logical Operators
* Bitwise Operators
* Assignment Operators
* Unary Operators
* Ternary Operators
* Misc Operators



**Precedence of Operators in C#**

The precedence of operator specifies that which operator will be evaluated first and next. The associativity specifies the operators direction to be evaluated, it may be left to right or right to left.

Let's understand the precedence by the example given below:

1. int data= 10+ 5\*5

The "data" variable will contain 35 because \* (multiplicative operator) is evaluated before + (additive operator).

The precedence and associativity of C# operators is given below:

|  |  |  |
| --- | --- | --- |
| **Category (By Precedence)** | **Operator(s)** | **Associativity** |
| Unary | + - ! ~ ++ -- (type)\* & sizeof | Right to Left |
| Additive | + - | Left to Right |
| Multiplicative | % / \* | Left to Right |
| Relational | < > <= >= | Left to Right |
| Shift | << >> | Left to Right |
| Equality | == != | Right to Left |
| Logical AND | & | Left to Right |
| Logical OR | | | Left to Right |
| Logical XOR | ^ | Left to Right |
| Conditional OR | || | Left to Right |
| Conditional AND | && | Left to Right |
| Null Coalescing | ?? | Left to Right |
| Ternary | ?: | Right to Left |
| Assignment | = \*= /= %= += - = <<= >>= &= ^= |= => | Right to Left |