**INCREMENT AND DECREMENT OPERATOR**

**Prefix operator**: (Before the variable, as in ++n) The expression ++n increments n before its value is used.

If n is 5, then

x = ++n;

Sets x to 6

**Postfix operator**: (After the variable, as in n++) The expression n++ increments n after its value has been used.

If n is 5, then

x = n++;

Sets x to 5

In both cases, n eventually becomes 6.

In both case, the effect is to increment n.

**ASSIGNMENT OPERATOR**

Expressions such as

i = i + 2

in which the variable on the left hand side is repeated immediately on the right, can be written in the compressed form.

i += 2

The operator += is called an assignment operator.

In general, if expr1 and expr2 are expressions, then

expr1 op= expr2

is equivalent to

expr1 = (expr1) op (expr2)

for ex: x \*= y + 1

means x = (x) \* (y + 1)

rather than x = x \* y + 1

**CONDITIONAL EXPRESSIONS**

In the expression

expr1 ? expr2 : expr3

the expression expr1 is evaluated first. If it is non-zero (true), then the expression expr2 is evaluated, and that is the value of the conditional expression. Otherwise expr3 is evaluated, and that is the value. Only one expr2 and expr3 is evaluated.

**PRECEDENCE AND ASSOCIATIVITY OF OPERATORS**

|  |  |
| --- | --- |
| **OPERATORS** | **ASSOCIATIVITY** |
| () [] -> . | Left to Right |
| ! ~ ++ -- + - \* & (type) sizeof | Right to Left |
| \* / % | Left to Right |
| + - | Left to Right |
| << >> | Left to Right |
| < <= > >= | Left to Right |
| == != | Left to Right |
| & | Left to Right |
| ^ | Left to Right |
| | | Left to Right |
| && | Left to Right |
| || | Left to Right |
| ?: | Right to Left |
| = += -= \*= /= %= &= ^= |= <<= >>= | Right to Left |
| , | Left to Right |

Unary +, -, and \* have higher precedence than the binary forms.