Examples :

**Evaluating complex expressions.**

1: #include <iostream>

Using namespace std;

2: int main()

3: {

4: int a=0, b=0, x=0, y=35;

5: cout << "a: " << a << " b: " << b;

6: cout << " x: " << x << " y: " << y << endl;

7: a = 9;

8: b = 7;

9: y = x = a+b;

10: cout << "a: " << a << " b: " << b;

11: cout << " x: " << x << " y: " << y << endl;

12: return 0;

13: }

Output: a: 0 b: 0 x: 0 y: 35

a: 9 b: 7 x: 16 y: 16

you can write

x = 35; // ok

but you can't legally write

35 = x; // error

if you have a variable, C, and you want to increment it, you would use this statement:

C++; // Start with C and increment it.

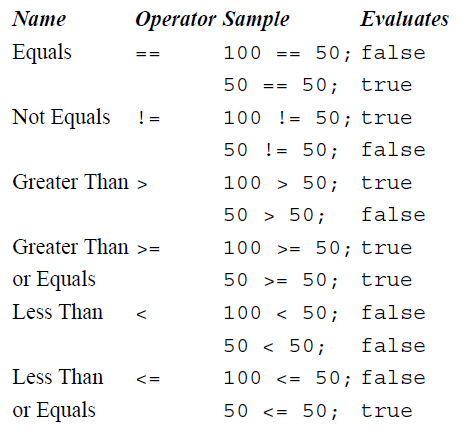
This statement is equivalent to the more verbose statement

C = C + 1;

which you learned is also equivalent to the moderately verbose statement

C += 1;

**The Relational Operators**



4: #include <iostream>

Using namespace std ;

5: int main()

6: {

7: int myAge = 39; // initialize two integers

8: int yourAge = 39;

9: cout << "I am: " << myAge << " years old.\n";

10: cout << "You are: " << yourAge << " years old\n";

11: myAge++; // postfix increment

12: ++yourAge; // prefix increment

13: cout << "One year passes...\n";

14: cout << "I am: " << myAge << " years old.\n";

15: cout << "You are: " << yourAge << " years old\n";

16: cout << "Another year passes\n";

17: cout << "I am: " << myAge++ << " years old.\n";

18: cout << "You are: " << ++yourAge << " years old\n";

19: cout << "Let's print it again.\n";

20: cout << "I am: " << myAge << " years old.\n";

21: cout << "You are: " << yourAge << " years old\n";

22: return 0;

23: }

Output: I am 39 years old

You are 39 years old

One year passes

I am 40 years old

You are 40 years old

Another year passes

I am 40 years old

You are 41 years old

Let's print it again

I am 41 years old

You are 41 years old