

Shortcuts to reduce character count in code and order of all operations

CS 10A – SHORTHAND & PRECEDENCE

Shorthand in C/C++

- In C++, we can reduce the number of characters we type in order to make code look cleaner and easy to read.
- Shorthand is about combining a math operation and variable storage into one line.
- Incrementation: `x++`, `x--`, `++x`, `--x`
- Incrementation is identical to `x = x+1` or `x = x-1`
- Math shorthand: combines any math operation (including bit manipulations) with a variable assignment back to the same variable. It's updating the variable value.

Nuances of Incrementation

Program

```
int x = 3, y = 7, a, b;
int main()
{
    cout << x++ << endl << ++y << endl;
    // x++ is a post-increment. Value is now 4.
    // ++y is a pre-increment. Value is still 8.

    a = x++; // assigns before incrementing
    b = ++y; // assigns after incrementing
    cout << a << endl << b << endl;
    cout << x << endl << y << endl;
    return 0;
}
```

Console

```
➤ ./a.exe
3
8
4
9
5
9
```

List of Other Shorthand Syntax

Description	Shorthand	Equivalent
Add then update	<code>x += 5</code>	<code>x = x+5</code>
Subtract then update	<code>x -= 2</code>	<code>x = x-2</code>
Multiply then update	<code>x *= 10</code>	<code>x = x*10</code>
Divide then update	<code>x /= 5</code>	<code>x = x/5</code>
Modulo then update	<code>x %= 3</code>	<code>x = x%3</code>
AND then update	<code>x &= 0b1101</code>	<code>x = x&0b1101</code>
OR then update	<code>x = 0b1001</code>	<code>x = x 0b1001</code>
XOR then update	<code>x ^= 0b0011</code>	<code>x = x^0b0011</code>

There are a few more shorthand notations but we'll touch upon those later.

Operation Precedence

- Regardless of how you order your symbols, there's always a specific order in how they execute.
- The order of which symbols are used first is called precedence. Basically an order of operations for C/C++.
- Remember that code generally executes left to right, top to bottom, but precedence can cause code to execute right to left instead.
- On the next table, precedence levels 3 and 16 are the only operators that execute right to left.

Table of Operator Precedence in C++

Precedence	Operators	Precedence	Operators
1		10	== !=
2	a++ a-- type() type{} a() a[]	11	&
3	++a --a ! ~ (type) sizeof()	12	^
4		13	
5	a*b a/b a%b	14	&&
6	a+b a-b	15	
7	<< >>	16	a?b:c = += -= *= /= %= <<= >>= &= ^= =
8		17	,
9	< <= > >=		

This table includes only what we'll cover in this class. For a full list, try this [link](#).