Increment and Decrement Operators

In this lesson, you will learn about ++, -- operators and what happens when you use them as the prefix or postfix operators.

You may come across two unusual-looking operators that may be used as a shorthand for incrementing and decrementing variables. The `++` and `--` operators add `1` and subtract `1`, respectively, from their operands. For example in the following code snippet, we increment the `int` variable `a` and we decrement the `int` variable `b`:

```
#include <stdio.h>

int main(int argc, char *argv[]) {

    int a = 0;
    int b = 0;

    printf("a=%d, b=%d\n", a, b);

    a++;
    b--;

    printf("a=%d, b=%d\n", a, b);

    return 0;
}
```

A note of caution, you can also use these two operators differently, by putting the operator before the operand, e.g. `++a` and `--b`.

> - When the operand is used **before** the operand it is called a **prefix operator**
> - When the operand is used **after** the operand it is called a **postfix operator**.

When using `++` and `--` as a prefix operator, the increment (or decrement) happens **before** its value is used. As for postfix operators, the increment (or decrement) occurs**after** its value has been used. Here is a concrete example:

```c
#include <stdio.h>

int main(int argc, char *argv[]) {

    int n, x;

    n = 3;
    x = 0;
    printf("n=%d, x=%d\n", n, x);
    x = n++;
    printf("n=%d, x=%d\n\n", n, x);

    n = 3;
    x = 0;
    printf("n=%d, x=%d\n", n, x);
    x = ++n;
    printf("n=%d, x=%d\n", n, x);

    return 0;
}
```

In lines 7 to 11, `x` is set to `3` (the value of `n`), and **then** `n` is incremented by `1`. In lines 13 to 17, `n` is incremented first and becomes `4`, and **then** `x` is set to the resulting value (also `4`).

If you think this is all a bit unnecessarily confusing, then you agree with me. I typically don't use these operators because of the risk of misusing them, and so when I want to increment or decrement the value by 1, I just write it out explicitly:

```c
#include <stdio.h>

int main(){
  int x = 1;
  x = x + 1;
  printf("%d",x);
}
```

We're at the end of this section. I hope it's been a fun way to enter the world of C programming. Be sure to check out the exercises in the next lesson.

The section ahead will deal with the flow and structure of our code's execution.