

A C++ Idiom

Knowing pointer arithmetic helps you understand one of the **most common idiomatic constructions** in C++, which you'll find on line 4 of the previous example:

```
while (beg != end) { result += *beg++; }
```

Here's how to decode the expression ***beg++**:

- The ***** operator and the **++** operator **compete** for the operand **beg**. Because unary operators in C++ are **right-associative**, the **++** takes precedence over the *****. The compiler interprets this as: ***(beg++)**.
- The **postfix ++** operator increments the value of **beg** so that it points to the next element, **but** returns the address that **beg** **was pointing to prior to the increment operation**. Since **beg** is a pointer, the increment operation uses pointer arithmetic; adding **1** to the address inside of **beg** creates a pointer to the next element in the array.
- If **beg** originally pointed to **a[0]**, the increment causes it to point to **a[1]**. The address that is used for dereferencing *****, is the address value it contained **before** the increment.

Thus, the expression ***beg++** has the following meaning in English:

*Dereference the pointer **beg** and return as an **lvalue** the object to which it currently points. As a side effect, increment the value of **beg** so that, if the original **lvalue** was an element in an array, the new value of **beg** points to the next element in that array.*



This course content is offered under a [CC Attribution Non-Commercial](#) license. Content in this course can be considered under this license unless otherwise noted.