For Loop with an Index Variable

A simple `for` loop using an index variable has the following syntax. Click the button below for an explanation of the different parts.

In [ ]:
```
#include <iostream>
using std::cout;

int main() {
    for (int i=0; i < 5; i++) {
      cout << i << "\n";
    }
}
```

Run Code | See Explanation

Loading terminal (id_f8j8za6), please wait...

The Increment Operator

If you haven't seen the `++` operator before, this is the *post-increment operator*, and it is where the `++` in the name "C++" comes from. The operator increments the value of `i`.

There is also a *pre-increment operator* which is used before a variable, as well as *pre* and *post decrement* operators: `--`. The difference between *pre* and *post* lies in what value is returned by the operator when it is used.

You will only use the *post-increment operator* `i++` for now, but if you are curious, click below for an explanation of the code:

In [ ]:
```
#include <iostream>
using std::cout;

int main() {
    auto i = 1;

    // Post-increment assigns i to c and then increments i.
    auto c = i++;

    cout << "Post-increment example:" << "\n";
    cout << "The value of c is: " << c << "\n";
    cout << "The value of i is: " << i << "\n";
    cout << "\n";

    // Reset i to 1.
    i = 1;

    // Pre-increment increments i, then assigns to c.
    c = ++i;

    cout << "Pre-increment example:" << "\n";
    cout << "The value of c is: " << c << "\n";
    cout << "The value of i is: " << i << "\n";
    cout << "\n";

    // Decrement i:
    i--;
    cout << "Decrement example:" << "\n";
    cout << "The value of i is: " << i << "\n";
}
```

Run Code | See Explanation

Loading terminal (id_n5nvoh2), please wait...

Practice

Before you learn how to write a `for` loop using an iterator, practice writing a for loop that prints values from `-3` through `10` in the cell below. Don't forget to assign an initial value (like 0) to your index variable!

In [ ]:
```
#include <iostream>
using std::cout;

int main() {
    // Add your code here.

}
```

Run Code | Show Solution

Loading terminal (id_l3jg0gv), please wait...

For Loop with a Container

C++ offers several ways to iterate over containers. One way is to use an index-based loop as above. Another way is using a "range-based loop", which you will see frequently in the rest of this course. See the following code for an example of how this works:

In [ ]:
```
#include <iostream>
#include <vector>
using std::cout;
using std::vector;

int main() {
    // Add your code here.
    vector<int> a {1, 2, 3, 4, 5};
    for (int i: a) {
        cout << i << "\n";
    }
}
```

Run Code | See Explanation

Loading terminal (id_ji14y5z), please wait...

Challenge

In the next cell, try to write a double range-based for loop that prints all of the entries of the 2D vector `b`. If you get stuck, click on the solution button for an explanation.

In [ ]:
```
#include <iostream>
#include <vector>
using std::cout;
using std::vector;

int main() {
    // Add your code here.
    vector<vector<int>> b {{1, 2},
                           {3, 4},
                           {5, 6}};

    // Write your double loop here.
    for(auto v : b) {
        for(int i : v) {
            cout << i << " ";
        }
        cout << "\n";
    }
}
```

Run Code | Show Solution

Loading terminal (id_fyqnmpn), please wait...