

Lesson 2:
Introduction to the C++ Language

SEARCH

RESOURCES

CONCEPTS

1. Intro

2. CODE: Write and Run Your First C...

3. Compiled Languages vs Scripted L...

4. C++ Output and Language Basics

5. CODE: Send Output to the Console

6. How to Store Data

7. Bjarne Introduces C++ Types

8. Primitive Variable Types

9. What is a Vector?

10. C++ Vectors

11. C++ Comments

12. Using Auto

13. CODE: Store a Grid in Your Progr...

14. Getting Ready for Printing

15. Working with Vectors

16. For Loops

17. Functions

18. CODE: Print the Board

19. If Statements and While Loops

20. Reading from a File

21. CODE: Read the Board from a File

22. Processing Strings

23. Adding Data to a Vector

24. CODE: Parse Lines from the File

25. CODE: Use the ParseLine Function

26. Formatting the Printed Board

27. CODE: Formatting the Printed Bo...

28. CODE: Store the Board using the ...

29. Great Work!

Adding Data to a Vector

SEND FEEDBACK

Vector push_back

Now that you are able to process a string, you may want to store the results of the processing in a convenient container for later use. In the next exercise, you will store the streamed `int` s from each line of the board in a `vector<int>` . To do this, you will add the `int` s to the back of the vector, using the `vector` method `push_back` :

In []:

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

int main() {
    // Initial Vector
    vector v {1, 2, 3};

    // Print the contents of the vector
    for (int i=0; i < v.size(); i++) {
        cout << v[i] << " ";
    }

    // Push 4 to the back of the vector
    v.push_back(4);

    // Print the contents again
    for (int i=0; i < v.size(); i++) {
        cout << v[i] << " ";
    }
}
```

Run Code

See Explanation

Loading terminal (id_givuguc), please wait...

The 4 has been added to the end of the vector!

On to an Exercise

In this section, you have learned about two useful tools:

• String streams, and

• vector::push_back.

The string streaming objects in C++ are very powerful, and there are many more ways that an `istringstream` can be used. We encourage you to have a look at all of the available functions in [the C++ reference \(http://www.cplusplus.com/reference/sstream/istringstream/\)](http://www.cplusplus.com/reference/sstream/istringstream/). However, at this point you are ready for the next exercise. In this exercise, you will parse string lines from the board and store the `int` s in a vector using the `istringstream` and `push_back` methods from above. Have a careful look at the examples again, and copy paste any code you think you might need before clicking the `Next` button below.

In []:

Menu

Shrink

NEXT