

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!

Vector Containers

1D Vectors

C++ also has several container types that can be used for storing data. We will start with `vector` s, as these will be used throughout this lesson, but we will also introduce other container types as needed.

Vectors are a sequence of elements of a single type, and have useful methods for getting the size, testing if the vector is empty, and adding elements to the vector.

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

int main() {
    // Three ways of declaring and initializing vectors.
    vector<int> v_1(0, 1, 2);
    vector<int> v_2 = {3, 4, 5};
    vector<int> v_3;
    v_3 = {6};
    cout << "Everything worked!" << "\n";
}
```

Run Code

See Explanation

Loading terminal (id_9fyahs), please wait...

2D Vectors

Unfortunately, there isn't a built-in way to print vectors in C++ using `cout` . You will learn how to access vector elements and you will write your own function to print vectors later. For now, you can see how vectors are created and stored. Below, you can see how to nest vectors to create 2D containers.

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

int main() {
    // Creating a 2D vector.
    vector<vector<int>> v {{1,2}, {3,4}};
    cout << "Great! A 2D vector has been created." << "\n";
}
```

Run Code

See Explanation

Loading terminal (id_zt557y), please wait...

Practice

Practice declaring a `vector<int>` in the cell below, and assign the value `{6, 7, 8}` .

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

int main() {
    // Declare and initialize a vector v here.
    vector<int> v{6,7,8};
    for(int i : v)
        cout << i << " ";
    cout << "\n";
}
```

Run Code

Show Solution

Loading terminal (id_o8p0gry), please wait...