

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!

C++ Output and Language Basics

SEND FEEDBACK

First Code Example

The next cell contains the first example of code that might be included in a typical C++ program. Hover your cursor over each line of the code and then click play to hear an explanation, or have a look at the **Review** section below.

In {}:

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

int main() {
    cout << "Hello!" << "\n";
}
```

Run Code

Loading terminal (id\_bco5kpa), please wait...

Review

#include <iostream>

- The `#include` is a preprocessor command which is executed before the code is compiled. It searches for the `iostream` header file and pastes its contents into the program. `iostream` contains the declarations for the input/output stream objects.

using std::cout;

- Namespaces are a way in C++ to group identifiers (names) together. They provide context for identifiers to avoid naming collisions. The `std` namespace is the namespace used for the standard library.
- The `using` command adds `std::cout` to the global scope of the program. This way you can use `cout` in your code instead of having to write `std::cout`.
- `cout` is an output stream you will use to send output to the notebook or to a terminal, if you are using one.
- Note that the second two lines in the example end with a semicolon `;`. Coding statements end with a semicolon in C++. The `#include` statement is a preprocessor command, so it doesn't need one.

cout << "Hello!" << "\n";

- In this line, the code is using `cout` to send output to the notebook. The `<<` operator is the stream insertion operator, and it writes what's on the right side of the operator to the left side. So in this case, `"Message here"` is written to the output stream `cout`.

Menu

Shrink

NEXT