## C++: Questions about using namespace std and cout [duplicate]

Asked 6 years, 10 months ago Active 6 years ago Viewed 22k times



This question already has answers here:

3

Why is "using namespace std;" considered bad practice? (36 answers)

Closed 6 years ago.



Why do I need to type in using namespace std; in order to be able to use cout and endl?

- Also what are these called; is cout a function?
- Is there cout in C? I heard it was implemented in C++ because it is better in many ways.







asked Mar 11 '13 at 19:28



- 5 <u>stackoverflow.com/questions/388242/...</u> Benjamin Lindley Mar 11 '13 at 19:31
- 1 cout is not a C library and there are many who would argue with the notion that it is "better in many ways." :-) acarlow Mar 11 '13 at 19:39

## 5 Answers



cout is a global object defined in the std namespace, and endl is a (stream manipulator) function also defined in the std namespace.



If you take no action to import their names into the global namespace, you won't be able to refer to them with the unqualified identifiers <code>cout</code> and <code>endl</code>. You have to use the fully qualified names:



std::cout << "Hello, World!" << std::endl;</pre>



Basically, what using namespace std does is to inject all the names of entities that exist in the std namespace into the global namespace:

```
using namespace std;
cout << "Hello, Wordl!" << endl;</pre>
```

However, keep in mind that have such a using directive in the global namespace is a **BAD** programming practice, which will almost certainly lead to evil **name clashes**.

If you really need to use it (e.g. if a function of yours is using many functions defined in the std

```
void my_function_using_a_lot_of_stuff_from_std()
{
    using namespace std;
    cout << "Hello, Wordl!" << endl;

    // Other instructions using entities from the std namespace...
}</pre>
```

Much better, as long as this is practical, is to use the following, less invasive *using declarations*, which will *selectively* import only the names you specify:

```
using std::cout;
using std::endl;
cout << "Hello, Wordl!" << endl;</pre>
```

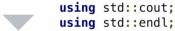
answered Mar 11 '13 at 19:33





**No!** You do not need using namespace std, and you shouldn't use it. Use fully qualified names std::cout and std::endl, or, in a small scope,

5



As for the other questions, std::cout is not a function. It is a kind of global output stream object bound to the standard output. And there isn't an std::cout in C.

edited May 23 '17 at 12:26



answered Mar 11 '13 at 19:29



If you don't have any other namespace, or the other namespaces have no way to interfere with the  $\,$  std namespace, it is ok to do it - Mihai Bujanca Mar 11 '13 at 19:33

@BujancaMihai no, it isn't. - juanchopanza Mar 11 '13 at 19:33

why not? Say you don't have any other namespace used. Why shouldn't you use using namespace std; — Mihai Bujanca Mar 11 '13 at 19:34

- @BujancaMihai because you may not know all the names that are under std::, and even if you do, you cannot know the names that will appear in future versions of the C++ standard library. juanchopanza Mar 11 '13 at 19:35
  - @BujancaMihai ideone.com/Xpgadi Mooing Duck Mar 11 '13 at 20:47



using namespace std;

brings the names in a collection of names (called a namespace) into the current scope. Most