

Home

PUBLIC

Questions

Tags

Users

Companies

COLLECTIVES

Explore Collectives

TEAMS

Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.

>_?

Free

Create a free Team

Why Teams?

What is the use of "using namespace std"? [duplicate]

Asked 9 years ago

Modified 3 years, 7 months ago

Viewed 268k times

Ask Question

▲

61

▼

35

🕒

This question already has answers here:

[Why is "using namespace std;" considered bad practice?](#) (41 answers)

[Using std Namespace](#) (16 answers)

Closed 9 years ago.

What is the use of `using namespace std`?

I'd like to see explanation in Layman terms.

c++

namespaces


std

using

Share


Follow

edited Nov 2, 2016 at 13:45

 Saurav Sahu

12k ● 5 ● 55 ● 74

asked Sep 20, 2013 at 10:06

 Jarvis

619 ● 1 ● 6 ● 4

7

In layman terms, *don't do that*. [Here be dragons](#). – Frédéric Hamidi Sep 20, 2013 at 10:08

[youtube.com/watch?v=xfgcEjCd9as](#) – NoWar Sep 24, 2015 at 18:43

Add a comment

2 Answers

Sorted by:

Highest score (default)

⬆

▲

90

▼

🕒

- using**: You are going to use it.
- namespace**: To use what? A namespace.
- std**: The `std` namespace (where features of the C++ Standard Library, such as `string` or `vector`, are declared).

After you write this instruction, if the compiler sees `string` it will know that you may be referring to `std::string`, and if it sees `vector`, it will know that you may be referring to `std::vector`. (Provided that you have included in your compilation unit the header files where they are defined, of course.)


If you *don't* write it, when the compiler sees `string` or `vector` it will not know what you are refering to. You will need to explicitly tell it `std::string` or `std::vector`, and if you don't, you will get a compile error.

Share

Follow

edited Feb 13, 2019 at 11:02

answered Sep 20, 2013 at 10:43

 Daniel Daranas

22.2k ● 9 ● 61 ● 112

4

just perfect... – roottraveller Sep 11, 2017 at 11:05

Add a comment

▲

30

▼

🕒

When you make a call to `using namespace <some_namespace>`; all symbols in that namespace will become visible without adding the namespace prefix. A symbol may be for instance a function, class or a variable.

E.g. if you add `using namespace std`; you can write just `cout` instead of `std::cout` when calling the operator `cout` defined in the namespace `std`.

This is somewhat dangerous because namespaces are meant to be used to avoid name collisions and by writing `using namespace` you spare some code, but loose this advantage. A better alternative is to use just specific symbols thus making them visible without the namespace prefix. Eg:

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

int main() {
    cout << "Hello world!";
    return 0;
}
```

Share

Follow

edited Sep 20, 2013 at 10:24

answered Sep 20, 2013 at 10:17

 Ivaylo Strandjev

67.4k ● 17 ● 119 ● 171

Bare in mind there are subtle differences between a *using-directive* and a *using-declaration* in regards to name lookup. – Simple Sep 20, 2013 at 10:24

@Simple I wanted to keep it simple because Op requested explanation in layman terms. – Ivaylo Strandjev Sep 20, 2013 at 10:25

1

It's also worth mentioning that *using-directives* and *using-declarations* are also subject to scope. Using *using* isn't universally bad. – thokra Sep 20, 2013 at 10:51

Suppose one wishes to load everything from `std`. How to do? `using std::` only? – Sigur Apr 18, 2017 at 1:11

if you want to load everything from the namespace you should be write `using namespace std` as I have written in my answer. However you should be aware of the consequences and you should only be using that if you really need it – Ivaylo Strandjev Apr 18, 2017 at 7:13

Add a comment


The Overflow Blog

 For developers, flow state starts with your finger tips

 Cassidy becomes a CTO! (Ep. 492)

Featured on Meta

 Recent Color Contrast Changes and Accessibility Updates

 Reviewer overboard! Or a request to improve the onboarding guidance for new...

 Should I explain other people's code-only answers?

 Should we burninate the [placement] tag?

 Collectives Update: Recognized Members, Articles, and GitLab

Linked

3182

Why is "using namespace std;" considered bad practice?

119

Using std Namespace

1

why is it compulsory to use using namespace std in c++ language?

0

'list' was not declared in this scope

1

How does C++ know where to look for the namespace specified using "using namespace ..."?

-9

What's wrong with my code about find the max one and the min one by vector

-3

How do I declare an object using this provided header file? (missing template arguments error)

-2

How do I insert two values into a <list> or <vector> so that I can retrieve values later on?

Related

2908

What is the difference between #include <filename> and #include "filename"?

3763

What are the differences between a pointer variable and a reference variable?

3484

What does the explicit keyword mean?

2331

Should 'using' directives be inside or outside the namespace?

4230

The Definitive C++ Book Guide and List

7724

What does if __name__ == "__main__": do?

1681

What is the difference between const int*, const int * const, and int const *?

3182

Why is "using namespace std;" considered bad practice?

9847

What is the "-->" operator in C++?

26631

Why is processing a sorted array faster than processing an unsorted array?

Hot Network Questions

 Cut along the lines

 How can a species be fine with digesting poisons and toxins?

 How might derelict space ships/stations be only partially salvageable?

 Why jobs lost when after reconnect ssh?

 Does the human voice qualify as noise?

more hot questions

 How are we doing?

Take our short survey

the answer you're looking for? Browse other questions tagged

c++

namespaces

std

using

[sk your own question.](#)

 STACK OVERFLOW

Questions

Help

PRODUCTS

Teams

Advertising

Collectives

Talent

COMPANY

About

Press

Work Here

Legal

Privacy Policy

Terms of Service

Contact Us

Cookie Settings

Cookie Policy

STACK EXCHANGE NETWORK

Technology

Culture & recreation

Life & arts

Science

Professional

Business

API

Data

Blog

Facebook

Twitter

LinkedIn

Instagram

Site design / logo © 2022 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#), rev 2022.9.30.28471