

Forum:[Hack Forums](#)

Thread: [Different compilers - different syntax?](#)

ApexDexterity - March 10, 2016, 12:15 pm

Hi guys.

I'm trying to learn C++, and I'm just messing around in CodeBlocks. I made a simple program, that asks the user to type any number but 5, and it works.

I've now downloaded Visual Studio 2013 and I Copy/pasted the code into Visual Studio and it won't run. Is there different syntax for different IDE's & Compilers?

Here's the code:

Code `#include <iostream>`

`using namespace std;`

```
int main()
{
    cout << "Type any number but 5!" << endl;
    int number;

    while(number !=5){
        cin >> number;
        cout << "Another number" << endl;
    }
    return 0;
}
```

Black?jack - March 10, 2016, 12:17 pm

Consider giving us the error.

Most mainstream compilers comply with the same spec, but sometimes some compilers will let you do out-of-spec things that won't work on others. Doesn't look like you've done any such things here, though.

Remix - March 10, 2016, 12:22 pm

no it doesnt have another syntax
just post your error

ApexDexterity - March 10, 2016, 12:25 pm

Oh sorry. Here's the error from Visual Studio: error C4700: uninitialized local variable 'number' used
CodeBlocks shows no errors.

Okay from the error, I tried to set number = 0; and it works in VS.

I just don't get why I need to initialize the variable.

I don't know what the number will be, that's why I set it as "cin", so the user can decide the number.

Retribute - March 10, 2016, 12:34 pm

I think it's because you didn't initialize 'number'.

PHP Code

```
<?php  
int number = 0;
```

ApexDexterity - March 10, 2016, 12:40 pm

(Mar 10, 2016 01:34 PM)Retribute Wrote: [\[CITE\]](#) I think it's because you didn't initialize 'number'.
PHP Code

```
<?php  
int number = 0;
```

Ye apparently that's my problem. Thanks ?

So as I said, I'm new to programming, but does setting the variable to 0 mean it's initialized? because I've tried setting it to everything (like 2, 20, 5000) and the code still runs.. So why set it to 0?

Cheese - March 10, 2016, 12:43 pm

(Mar 10, 2016 01:40 PM)ApexDexterity Wrote: [\[CITE\]](#) Ye apparently that's my problem. Thanks ?

So as I said, I'm new to programming, but does setting the variable to 0 mean it's initialized? because I've tried setting it to everything (like 2, 20, 5000) and the code still runs.. So why set it to 0?

Look here, <http://www.cprogramming.com/tutorial/common.html> this will give you 8 common mistakes beginner programmers face, which includes uninitialized variables.

push rcx - March 10, 2016, 12:45 pm

(Mar 10, 2016 01:40 PM)ApexDexterity Wrote: [\[CITE\]](#) Ye apparently that's my problem. Thanks ?

So as I said, I'm new to programming, but does setting the variable to 0 mean it's initialized? because I've tried setting it to everything (like 2, 20, 5000) and the code still runs.. So why set it to 0?

Local variables are allocated on this list referred to as a stack, when a variable is allocated you don't know what is at that location. Initializing it ensures that the coder knows the definition.

Retribute - March 10, 2016, 1:01 pm

When a variable is declared, it will point to a piece of memory.

Accessing the value of the variable will give you the contents of that piece of memory.

However until the variable is initialised, that piece of memory could contain anything. This is why using it is unpredictable.

ApexDexterity - March 10, 2016, 1:10 pm

Okay, so it needs to be intialized, but shouldn't it matter wheter or not it's set to 0 or something like 5000?

Cheese - March 10, 2016, 1:18 pm

(Mar 10, 2016 02:10 PM)ApexDexterity Wrote: [\[CITE\]](#) Okay, so it needs to be intialized, but shouldn't it matter wheter or not it's set to 0 or something like 5000?

Yes it does matter, you are changing the value of that variable, if you were to want to add onto that variable later you would select what value to set it to, which could be 0,1,2,3,4...etc... I'm really bad at explaining things, but hopefully you understand that. Judging by your code, it doesn't matter what you set it as, because when the user enters there input it will overwrite the existing variable value.

ApexDexterity - March 10, 2016, 1:31 pm

(Mar 10, 2016 02:18 PM)Consult Wrote: [\[CITE\]](#) Yes it does matter, you are changing the value of that variable, if you were to want to add onto that variable later you would select what value to set it to, which could be 0,1,2,3,4...etc... I'm really bad at explaining things, but hopefully you understand that. Judging by your code, it doesn't matter what you set it as, because when the user enters there input it will overwrite the existing variable value.

I completely understood that. Thanks alot ?

0xDEAD10CC - March 10, 2016, 3:28 pm

(Mar 10, 2016 02:18 PM)Consult Wrote: [\[CITE\]](#) Yes it does matter, you are changing the value of that variable, if you were to want to add onto that variable later you would select what value to set it to, which could be 0,1,2,3,4...etc... I'm really bad at explaining things, but hopefully you understand that. Judging by your code, it doesn't matter what you set it as, because when the user enters there input it will overwrite the existing variable value.

If he set's it to 5, it would matter because the loop would never execute. A do-while loop makes much more sense than a while loop here for what he's trying to do however. He always wants it to at least loop through once, and there's no sense in the comparison before any meaningful number is even given to the variable as a value for when the code enters the loop conditional for the first time.

Cheese - March 10, 2016, 3:52 pm

(Mar 10, 2016 04:28 PM)0xDEAD10CC Wrote: [\[CITE\]](#) If he set's it to 5, it would matter because the loop would never execute. A do-while loop makes much more sense than a while loop here for what he's trying to do however. He always wants it to at least loop through once, and there's no sense in the comparison before any meaningful number is even given to the variable as a value for when the code enters the loop conditional for the first time. Yeah, but why would he put it to 5 if that's the number that executes the code in the loop? Just thought he would already would've known to not set the variable to 5.

I see what you are saying, thanks for the clarification.

sqsgkrldwg - March 11, 2016, 12:56 am

(Mar 10, 2016 01:22 PM)Teh Remix Wrote: [\[CITE\]](#) no it doesnt have another syntax just post your error

Some compilers are more strict than others and can lead to some syntax modifications.

For instance when you use a double template definition like : `class<type<type>>`, gcc will not accept this while visual studio will be ok with it ; gcc will force you to use `class<type<type>> >` (notice the space).

But this is not a big deal, it's not like it will be a completely different language so in practice we can say the syntax doesn't change effectively. This is more effecting the design of the code.

0xDEAD10CC - March 11, 2016, 10:35 am

(Mar 11, 2016 01:56 AM)Leffe Wrote: [\[CITE\]](#) Some compilers are more strict than others and can lead to some syntax modifications.

For instance when you use a double template definition like : `class<type<type>>`, gcc will not accept this while visual studio will be ok with it ; gcc will force you to use `class<type<type>> >` (notice the space).

But this is not a big deal, it's not like it will be a completely different language so in practice we can say the syntax doesn't change effectively. This is more effecting the design of the code.

That's not a compiler being strict or less strict, that's C++ standard feature support. The syntax doesn't change for a template, what you can do with it does.

