

Forum:[Hack Forums](#)

Thread: [Console application closes right away](#)

Camf ? - March 13, 2018, 6:50 pm

Hey guys, I am going through the beggining stages of C++.

I ran into a few problems, and got most of them sorted but I'm still stuck with the error that once it runs all of the line of code the application closes.

I did some searching and found a fix using "cin.get()" but as I'm going through tutorials it's not working any more.

It works with,

```
Code int main ()
{

cout << "Hello World! \n new line";
cin.get();
return 0;
}
```

It stopped working with this,

```
Code int main ()
{
    int a;
    int b;
    int sum;

    cout << "Enter a number! \n";
    cin >> a;

    cout << "Enter Another number";
    cin >> b;

    sum = a + b;

    cout << "the sum is " << sum << endl;

    cin.get();
    return 0;
}
```

How can I fix this closing issue permanently?

I am also curious what exactly cin.get() does and what makes it a temporary fix?

Thanks everyone! I look forward to joining this section over the next few weeks ?

hello98321 - March 13, 2018, 9:23 pm

To fix the closing put system("pause"); at the end of your source code before return

0xDEAD10CC - March 13, 2018, 10:51 pm

(Mar 13, 2018 09:23 PM)hello98321 Wrote: [\[CITE\]](#) To fix the closing put system("pause"); at the end of your source code before return

In all honesty that is terrible advice. --

I'd suggest him learning how to actually program and understand that from the input provided by the user there's a left behind '\n' in the buffer that cin.get() grabs right before the return 0... This is because it isn't needed to be extracted when the integer is parsed out from the data in the stream. system("pause") is the very LAST thing he should do, if that.

Camf ? - March 14, 2018, 12:50 am

(Mar 13, 2018 10:51 PM)0xDEAD10CC Wrote: [\[CITE\]](#) In all honesty that is terrible advice. --

I'd suggest him learning how to actually program and understand that from the input provided by the user there's a left behind '\n' in the buffer that cin.get() grabs right before the return 0... This is because it isn't needed to be extracted when the integer is parsed out from the data in the stream. system("pause") is the very LAST thing he should do, if that.

ahhhh, okay that makes sense. Thank you very much ?

0xDEAD10CC - March 15, 2018, 4:47 pm

(Mar 14, 2018 12:50 AM)Camf ? Wrote: [\[CITE\]](#) ahhhh, okay that makes sense. Thank you very much ?

system() under any circumstance except NULL passed in is undefined behavior.

Gbull - March 16, 2018, 12:08 am

(Mar 13, 2018 09:23 PM)hello98321 Wrote: [\[CITE\]](#) To fix the closing put system("pause"); at the end of your source code before return

This is the horrible advice. Don't listen to this.

@OP:

The reason your program stops is because you don't tell it to keep running.

Think like a programmer.

Hoodie+ - March 16, 2018, 2:14 am

Didn't see any good answer at all. Since the Window closes you are most likely using an IDE which opens a new console window each time the program is run.

My guess is that you did not specify the subsystem of the application in your IDE. For example in Visual Studio, when you create a new Empty project the subsystem will not be set and if you make a hello world program the console will close as soon as it outputs the text. If you change the subsystem to console the console will stay after it outputs the text.

0xDEAD10CC - March 16, 2018, 10:22 am

(Mar 16, 2018 02:14 AM)Hoodie+ Wrote: [\[CITE\]](#) Didn't see any good answer at all. Since the Window closes you are most likely using an IDE which opens a new console window each time the program is run.

My guess is that you did not specify the subsystem of the application in your IDE. For example in Visual Studio, when you create a new Empty project the subsystem will not be set and if you make a hello world program the console will close as soon as it outputs the text. If you change the

subsystem to console the console will stay after it outputs the text.

"If you change the subsystem to console the console will stay after it outputs the text"

That's not true. A console application still doesn't have any message loop so there's no reason for it to stay open after.

Hoodie+ - March 16, 2018, 11:27 am

(Mar 16, 2018 10:22 AM)0xDEAD10CC Wrote: [\[CITE\]](#) "If you change the subsystem to console the console will stay after it outputs the text"

That's not true. A console application still doesn't have any message loop so there's no reason for it to stay open after.

Well, what is your explanation on my console staying open then?

Spoiler [\(Click to View\)](#)

Camf ? - March 16, 2018, 7:03 pm

(Mar 16, 2018 11:27 AM)Hoodie+ Wrote: [\[CITE\]](#) Well, what is your explanation on my console staying open then?

Spoiler [\(Click to View\)](#)

How do I go about this?

wit - March 16, 2018, 7:04 pm

system("pause") works perfectly fine if doing this for school related projects

Camf ? - March 16, 2018, 7:05 pm

(Mar 16, 2018 07:04 PM)Hyphonix Wrote: [\[CITE\]](#) system("pause") works perfectly fine if doing this for school related projects

I'm actually learning C++ to dive deeper into Block Chain Development.

What makes System("pause") a temp fix?

Gbull - March 16, 2018, 10:08 pm

(Mar 16, 2018 07:05 PM)Camf ? Wrote: [\[CITE\]](#) I'm actually learning C++ to dive deeper into Block Chain Development.

What makes System("pause") a temp fix?

The system() function is exclusive to windows, as it calls a windows commandline function. It will not be able to cross-compile.

Not to mention, it hogs a shitload of system resources and is extremely inefficient.

Using std::cin.get() is a much better method.

Zettabit - March 17, 2018, 12:27 am

(Mar 16, 2018 11:27 AM)Hoodie+ Wrote: [\[CITE\]](#) Well, what is your explanation on my console staying open then?

Spoiler [\(Click to View\)](#)

Are you running it inside your IDE?

Hoodie+ - March 17, 2018, 12:41 am

(Mar 17, 2018 12:27 AM)Zettabit Wrote: [\[CITE\]](#) Are you running it inside your IDE?

Of course I am.

Who would use an IDE and not run his app in it.

On the other end if you want the end user to use your application, any normal end user will open up cmd or powershell, cd to the program's directory and executing the program that way and they don't have to worry about the console closing.

Keat - March 17, 2018, 8:46 am

I use getchar() in C. It will keep the prompt open.

Nomad.Null - March 17, 2018, 11:23 am

If you want your program to wait for user input the cin.get() function is sufficient to hold the program open until you strike a key. cin.get() can be described as a "ConsoleInput.get()" function, meaning it is wait for the user to feed it expected input. It is generally just for getting a single character, although you can use it for getting multiple value types. What you are trying to do is manipulate character arrays better known as strings. You can do this two ways, by explicitly defining a character array pointer or by using the <string> library to manipulate string types which is often neater and more readable.

In this example, I avoid using namespaces because I think it helps you to understand where all these functions exist. :)

Code #include <iostream>

#include <string>

```
int main(void){
    string myString;
    while(myString != "quit"){
        std::cout << "Enter a string (quit to exit): " << std::endl;
        std::getline(cin, myString);
        if(myString == "quit"){
            break;
        }
        cout << "You typed: \n" << myString << endl;
    }
    return 0;
}
```

Note the getline() function utilizes cin to capture the user input and assign it to the myString variable each time the while loop iterates. It will not break out of the loop until the user enters "quit" at which point the flow continues to the return and the program ends gracefully. Notice that NONE of my code accounts for a maximum buffer length which is technically not very secure code and since I am using a while loop, if I were using malloc() to allocate memory, this could lead to some serious memory leak problems in the future. Another potential consequence is that someone might figure out a way to abuse this and gain control of my code to cause it to break or execute malicious code.

Always consider these things when writing your programs this way. Use secure functions that define a maximum buffer length and account for bad user input.

Try rewriting this program using cin.getline() which does allow you to use maximum buffer lengths.

Also consider the difference between strcpy() and strncpy(). Try rewriting my program to make it more secure and account for bad user input. ?

Camf ? - March 17, 2018, 11:57 am

(Mar 17, 2018 11:23 AM)True Demon Wrote: [\[CITE\]](#) If you want your program to wait for user input the cin.get() function is sufficient to hold the program open until you strike a key. cin.get() can be described as a "ConsoleInput.get()" function, meaning it is wait for the user to feed it expected input. It is generally just for getting a single character, although you can use it for getting multiple value types. What you are trying to do is manipulate character arrays better known as strings. You can do this two ways, by explicitly defining a character array pointer or by using the <string> library to manipulate string types which is often neater and more readable.

In this example, I avoid using namespaces because I think it helps you to understand where all these functions exist. :)

```
Code #include <iostream>
#include <string>

int main(void){
    string myString;
    while(myString != "quit"){
        std::cout << "Enter a string (quit to exit): " << std::endl;
        std::getline(cin, myString);
        if(myString == "quit"){
            break;
        }
        cout << "You typed: \n" << myString << endl;
    }
    return 0;
}
```

Note the getline() function utilizes cin to capture the user input and assign it to the myString variable each time the while loop iterates. It will not break out of the loop until the user enters "quit" at which point the flow continues to the return and the program ends gracefully. Notice that NONE of my code accounts for a maximum buffer length which is technically not very secure code and since I am using a while loop, if I were using malloc() to allocate memory, this could lead to some serious memory leak problems in the future. Another potential consequence is that someone might figure out a way to abuse this and gain control of my code to cause it to break or execute malicious code.

Always consider these things when writing your programs this way. Use secure functions that define a maximum buffer length and account for bad user input.

Try rewriting this program using cin.getline() which does allow you to use maximum buffer lengths. Also consider the difference between strcpy() and strncpy(). Try rewriting my program to make it more secure and account for bad user input. ?

Am I on the right path?

```
Code #include <iostream>
#include <string>
```

```
using namespace std;
```

```
class QuitProgram(){
public:
```

```

void setquitnow(string x){
    quitnow = x
}

string quitstring(){
    return quitnow;
}

private:
string quitnow;

}

int main(void){
    QuitProgram QuitO;
    QuitO.setquitnow("quit")
    while(QuitO.myString != "quit"){
        cout << "Enter a string (quit to exit): " << endl;
        getline(cin, quitstring);
        if(myString == "quit"){
            break;
        }
        cout << "You typed: \n" << myString << endl;
    }
    return 0;
}

```

int main() - March 18, 2018, 5:46 am

Jesus fuck since when has keeping the console window up become rocket science

just put `cin.get()` before you return the gucci code

Electron?¹ - March 18, 2018, 12:36 pm

(Mar 18, 2018 05:46 AM)int main() Wrote: [\[CITE\]](#) Jesus fuck since when has keeping the console window up become rocket science

just put `cin.get()` before you return the gucci code

That won't work everytime. In this example the coder presses enter. Thats why `cin.get()` and also `getchar()` will read `\n` and continue then.

To prevent that you have to clear the input buffer first like someone said before.

Btw. here is another solution:

Code `fseek(stdin,0,SEEK_END);`

0xDEAD10CC - March 19, 2018, 10:24 pm

(Mar 16, 2018 11:27 AM)Hoodie+ Wrote: [\[CITE\]](#) Well, what is your explanation on my console staying open then?

Spoiler [\(Click to View\)](#)

Simply this: <https://stackoverflow.com/a/1152873>

It has nothing to do with the fact that it's a console application, it's an IDE thing.