

Forum:[Hack Forums](#)

Thread: [Void\(\) function](#)

Edri - February 18, 2019, 4:17 pm

I'm very confused with void() function difference between int() on CPP

I ran this code:

Code #include <iostream>

#include <conio.h>

using namespace std;

```
class Message {
private:
    int id;
public:
    void print() {
        std::cout << "Hello C++" << std::endl;
    }
};
```

```
int main() {
    Message m;
    m.print();
    _getch();
    return 0;
}
```

Does void function mean no return and int return integral value?

I'm so confused at this lol

slacker111 - February 18, 2019, 4:26 pm

Yes your correct, void as a function return type does not return a value when the function is executed.

Edri - February 18, 2019, 4:30 pm

(Feb 18, 2019 05:26 PM)slacker111 Wrote: [\[CITE\]](#) Yes your correct, void as a function return type does not return a value when the function is executed.

Thank you

Are there cases which void may return a value?

turbosoggy - February 18, 2019, 4:35 pm

(Feb 18, 2019 05:30 PM)Edri Wrote: [\[CITE\]](#) Thank you

Are there cases which void may return a value?

Void can return another void function or a void pointer can return a value

slacker111 - February 18, 2019, 4:40 pm

(Feb 18, 2019 05:30 PM)Edri Wrote: [\[CITE\]](#) Thank you

Are there cases which void may return a value?

No. often you write functions that do stuff and don't need to return anything; for this reason, someone decided that, to specify to the compiler that you don't want to return anything at all, you

have to use the void keyword as "return type".

6577657534735 - February 18, 2019, 7:29 pm

(Feb 18, 2019 05:35 PM)turbosoggy Wrote: [\[CITE\]](#) Void can return another void function or a void pointer can return a value

This is wrong, void can't return anything. And void pointers shouldn't be used as values.

halocedark - February 18, 2019, 9:47 pm

void functions dont return any data type, they just execute some code.

Mostly used for printing on the screen, setting up values for variables (when u work with classes)...

turbosoggy - February 19, 2019, 12:30 pm

(Feb 18, 2019 08:29 PM)Abraluna Wrote: [\[CITE\]](#) This is wrong, void can't return anything. And void pointers shouldn't be used as values.

Nah you're wrong. I recommend you read up on basic C before giving false advice, void isnt some special edge case that cant return, a void function can return something.

Code #include <stdio.h>

```
int main(void)
{
    hello();
    return 0;
}
```

```
void hello(void)
{
    printf("Hello");
    return world();
}
```

```
void world(void)
{
    printf(" World");
    return;
}
```

This is valid, also void pointers are used as basically generic pointer types so you can have a function that takes any type for example

Code send_data(void *data, int length)

0xDEAD10CC - February 19, 2019, 2:47 pm

(Feb 18, 2019 05:30 PM)Edri Wrote: [\[CITE\]](#) Thank you

Are there cases which void may return a value?

My answer to this would be yes, but not in the conventional sense that most would think. void return from a function just indicates "ignore the return register" in the same way other keywords with the language mean certain things to the compiler, but can be bypassed if you understand what they're doing, and you understand the language itself... In most cases they are compiler restrictions.

Just because a function return is void though doesn't mean you can't set something in the eax register and then retrieve it from the caller. void and void pointers are a different concept; different types for that matter all together.

(Feb 19, 2019 01:30 PM)turbosoggy Wrote: [\[CITE\]](#) ...

Maybe it's valid for clueless self-proclaimed programmers like you who don't even have a basic

understanding of what function prototypes are for. This fails the most basic pre-requisites to even compile successfully.

ISO C doesn't allow this either, even if you did manage to get the basics right. There's a reason why people avoid non-standard code and undefined behavior. It also shows that you don't truly understand the code you posted to begin with. You're not actually returning the data, and your code doesn't prove that either. You're just calling the 2nd function in the return statement...

6577657534735 - February 19, 2019, 7:23 pm

(Feb 19, 2019 01:30 PM)turbosoggy Wrote: [CITE](#) Nah you're wrong. I recommend you read up on basic C before giving false advice, void isnt some special edge case that cant return, a void function can return something.

Code #include <stdio.h>

```
int main(void)
{
    hello();
    return 0;
}
```

```
void hello(void)
{
    printf("Hello");
    return world();
}
```

```
void world(void)
{
    printf(" World");
    return;
}
```

This is valid, also void pointers are used as basically generic pointer types so you can have a function that takes any type for example

Code send_data(void *data, int length)

Void functions are allowed to have return statements as long as they return void. This is necessary and makes sense to allow things like C-autogen and C++ templates that work on void types and non-void types (for instance pass-through templated functions). Just because a void function can have a return statement doesn't mean it's returning anything. Your example takes the return of a void function, which is void, and returns it from itself, which is also a void function, so it's still not returning anything non-void. Try returning 1 from a void function or assigning the result from a void function to a variable. You can't. Your code is a syntactical edge case that results from the language's generalized and future-thinking grammar, it has no real semantic meaning.

Your function hello(void) is not returning a void pointer like you claimed above, it's returning the result of calling world, which is void, so the return statement is equivalent to a plain "return;"