What does ampersand "&" do in front of pointers?

Asked 8 years, 3 months ago Modified 4 years, 11 months ago Viewed 28k times

15

When functions are being called I often see the ampersand in front of the pointer in the function parameter.

E.g.

```
int *ptr;
randomFunction(&ptr);
```

7

I have done some research and found that this means that the function uses pointers to pointers. Is the `&` sign in front of a pointer used just to indicate this or does it do something else?

c++ c pointers reference operators

Share Edit Follow Flag

edited Apr 11, 2015 at 3:05
herohuyongtao
47.6k ● 25 ● 123 ● 162

asked Jan 22, 2014 at 3:15
user3213163
497 ● 3 ● 7 ● 10

4 returns the address of it. So `&ptr` will return an `int**` You would usually do this if you want the function to change what the pointer points at or do some sort of assignment to it. – Brandon Jan 22, 2014 at 3:16

Address of ptr. ie int** here – Digital_Reality Jan 22, 2014 at 3:17

@CantChooseUsernames: That's rather obsolete (C style). In C++, you'd use `int*&` for that. – MSalters Jan 22, 2014 at 8:48

@MSalters We have different opinions on obsolete I guess. Sometimes you want the user to be able to enter "nullptr". You cannot do this with a reference. Also it makes it "more" obvious what is being done to the parameter passed. Other than that, you are indeed very right. I myself tend to use `T* &ptr` more than `T**`. But you know, sometimes there's always that exception mentioned above. – Brandon Jan 23, 2014 at 0:40

@Brandon but you can change what pointer points to without getting pointer to the pointer and you can lead assignment too. – O.G. Sep 4, 2019 at 10:48

4 Answers

Sorted by: Highest score (default)

36

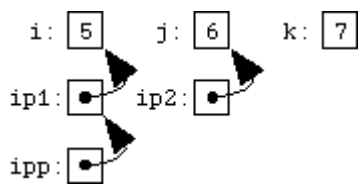
It's a pointer to the pointer.

`&` is the reference operator, and can be read as `address of`. In your example, it will get another pointer, that is the address of the pointer given as it's argument, i.e. a pointer to the pointer.

Look at the following example:

```
int **ipp;
int i = 5, j = 6, k = 7;
int *ip1 = &i, *ip2 = &j;
ipp = &ip1;
```

You will get:



In the above example, `ipp` is a pointer to pointer. `ipp` stores the address of `ip1` and `ip1` stores the address of `i`.

You can check out Pointers to Pointers for more info.

Share Edit Follow Flag

edited Jun 19, 2017 at 12:34
jdhao
18k ● 10 ● 114 ● 202

answered Jan 22, 2014 at 3:18
herohuyongtao
47.6k ● 25 ● 123 ● 162

1 Thanks! that website really helped – user3213163 Jan 22, 2014 at 5:25

3 Kind of off topic, but why would one want to use a pointer to a pointer? – user2344665 Feb 28, 2014 at 18:25

@JDMDev One example would be COM uses ptr-to-ptr to return an interface pointer using `CoCreateInstance()` and `IUnknown::QueryInterface()`. Check out here for more info. – herohuyongtao Feb 28, 2014 at 18:30

8

Take a step back. The fundamental rules of pointer operators are:

- The `*` operator turns a *value* of type `pointer to T` into a *variable* of type `T`.
- The `&` operator turns a *variable* of type `T` into a *value* of type `pointer to T`.

So when you have

```
int *ptr;
```

`ptr` is a variable of type `pointer to int`. Therefore `*ptr` is a variable of type `int` -- the `*` turns a *pointer* into a *variable*. You can say `*ptr = 123;`.

Since `ptr` is a variable of type `pointer to int`, `&ptr` is a *value* -- not a variable -- of type `pointer to pointer to int`:

```
int **pp = &ptr;
```

`&ptr` is a value of type `pointer to pointer to int`. `pp` is a variable of type `pointer to pointer to int`. `*pp` is a variable of type `pointer to int`, and in fact is the *same* variable as `ptr`. The `*` is the *inverse* of the `&`.

Make sense?

Share Edit Follow Flag

answered Jan 22, 2014 at 16:13
Eric Lippert
630k ● 172 ● 1210 ● 2051

1

It helps to think of "&" this way. int function_name (&(whatever)); You are passing the address of (whatever). Whatever can be a number of things: an elementary variable. a function. a structure. a union. an array. You should mentally translate "&" to "take the address of". So your example means : pass a COPY of the address of the address of the variable ptr of type int!

Share Edit Follow Flag

answered Jan 22, 2014 at 5:25
user3221497
11 ● 1

Do you mean '&' should be '*' – Zo Has Jan 22, 2014 at 5:49

1 @DamienJoe: No! `&` is the opposit of `*`. `&` takes the address of it's argument and `*` returns what`s stored under it's argument (interpreting the argument as address). – alk Jan 22, 2014 at 7:43

@alk Thanks for the info. Updated my concepts with that. – Zo Has Jan 22, 2014 at 8:51

0

```
Int *ptr;
```

&ptr returns the address of a pointer variable ptr. In short, double pointer or int** holds the address of ptr with &ptr.

Share Edit Follow Flag

answered Jan 22, 2014 at 5:07
Fahad Naeem
504 ● 6 ● 15