

Re: better understanding references

Source: http://coding.derkeiler.com/Archive/C_CPP/alt.comp.lang.learn.c-cpp/2004-03/0318.html

From: David White (*no_at_email.provided*)

Date: 03/04/04

Date: Thu, 4 Mar 2004 15:50:39 +1100

"thides" <swatts@globalserve.net> wrote in message
news:67y1c.68104\$9j1.25598@nntp-post.primus.ca...
>
> "Gary" <glabowitz@comcast.net> wrote in message
> news:f6-dnecc3btOGdvdRVn-tw@comcast.com...
>> Stop trying to think about pointers, addresses and the rest. Think
> instead:
>>
>> When I write a function with a parameter list containing &x, I am
renaming
>> the value being passed to me as "x." The identifier x is a name for a
>> location in memory containing the value being passed.
>>
>> Caller... z = foo(y);
>>
>> Called... int foo(int &x) { ...
>>
>> The variable addressed by "y" is the same variable addressed by "x" even
>> though the identifiers are in different code blocks.
>>
>> If you understand this, you understand references in general.
>>
>> int a;
>> int &b = a;
>>
>> The two identifiers "a" and "b" each refer to the value in a single
>> variable. The variable addressed by "a" is the same variable addressed
by
>> "b."
>>
> I guess if a changes then so will b. And if b changes so will a.
>
> Like this:
>
> int a;
> int &b=a;
>

> *b=4; // therefore a=4*

Correct.

> *But that wont work because b is a reference to a value not a value itself...*

What do you mean "that won't work"?

b is another name for a, or b refers to a (hence it is a reference). If you assign b to 4, a and b will equal 4, because a and b refer to the same value. If you assign a to 4, a and b will again equal 4.

> *OK I think if I am correct on this point i might be able to move on.*

Well, I don't know if you are.

DW