

Re: help: class compile error

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-07/1319.html

From: Howard (alicebt_at_hotmail.com)

Date: 07/09/04

Date: Fri, 09 Jul 2004 21:21:26 GMT

"xuatla" <xuatla@gmail.com> wrote in message
news:40EF066E.90801@gmail.com...

```
> One more question about new and delete:
>
> I use new to create a CTest. And then delete it. But it seems that the
> deletion is not successful. When I print it, it's still there.
>
> Do I need to define new and delete for class by myself?
>
> // main function
> int main(void)
> {
> int N = 10;
> CTest t1(N);
> int i;
>
> for ( i = 0; i < N; i++ ) t1[i] = i;
>
> CTest *t2 = new CTest;
>
> *t2 = t1;
>
> for ( i = 0; i < N; i++ ) cout << (t1+*t2)[i] << endl;
>
> delete t2;
>
> for ( i = 0; i < N; i++ ) cout << (*t2)[i] << endl;
> // still print out t2.
>
> return 1;
> }
>
> Thanks a lot for your kind help!
>
> X
```

Once you call delete on the pointer t2, it is illegal to use that pointer (unless you assign it to another valid object first). What you are doing *may* work in some cases (because it is *possible* that the memory at that location has not changed...yet). But you can *never* rely on doing that successfully. You are invoking "undefined behavior", which means that anything can happen. It may work, it may crash, and it may even cause *serious* problems! Simply put, "don't do that!" :-)

By the way, calling delete does not cause the computer to go to that memory location and erase anything. What it does (well, one of the things it does, somehow, behind the scenes), is mark the memory as "invalid" that was previously used by the object that was destroyed. At any time in the future, the computer is free to use that memory for any other purpose it needs it for. In this simple case, it just happened that that memory was still the same as when you called delete, so nothing bad happened when you used it (illegally). But try it again sometime and you may have to reboot your computer...or worse!

-Howard