

Re: Why it is not good code for constructor

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2007-12/msg00959.html>

- *From:* George <George@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 26 Dec 2007 21:39:02 -0800
-

Hi Ben,

Since the memory was allocated in C::C, it is most likely C::~~C that is responsible for freeing it,

How can C's destructor free the memory? There is no member variable in class C pointed to the memory. :-)

Please feel free to correct me if I am wrong.

regards,
George

"Ben Voigt [C++ MVP]" wrote:

"Alexander Grigoriev" <alegr@xxxxxxxxxxxxx> wrote in message news:OgcmX1%23RIHA.5976@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Are you sure argument of B() is evaluated no immediately before B() is called? If, suppose, the base classes are constructed in B, A order, then if A() throws, ~B() will be called.

Since the memory was allocated in C::C, it is most likely C::~~C that is responsible for freeing it, but C::~~C will not be called, and a function-try handler cannot access the pointer, while a catch handler in local scope will not be executed.

If B::~~B is freeing the pointer, then that would be correct usage of RAII and safe. But the example being discussed makes it clear this isn't the case, C is responsible for freeing the memory.

Re: Why it is not good code for constructor

"Ben Voigt [C++ MVP]" <rbv@xxxxxxxxxxxxxx> wrote in message
news:%23K3hjF9RIHA.5524@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

"George" <George@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote
in message
news:75F1E950-242F-46F3-9500-5BAA7FAC2AB6@xxxxxxxxxxxxxxxxxxxxx

Hello everyone,

Here is a sample from Dr. Dobb C++. In the
analysis, the code is bad
below.

But I do not think the code is bad,

1. if bad_alloc is thrown in new int[], we just
catch it and write some
log;
2. if there are any exception in B's
constructor, we will also be in
catch
block and we could also write some log.

Why it is bad code? Any comments?

(I do not agree that there is resource leak,
since if we met with
bad_alloc
in new int[], there is no memory allocated at
all, so no root of
memory/resource leak).

The code you provided isn't sufficient to guarantee that.

What if:

```
class C : public B, public A
{
...
};
```

now if new int[n] completes ok, but the constructor for A
throws, the
array is leaked with no chance to free it.

Re: Why it is not good code for constructor

<http://www.ddj.com/cpp/184401297>

```
[Code]
C::C(int)
try
: B(new int[n]) // horrible!
{
...
}
catch(Error &e)
{

}
[/Code]
```

thanks in advance,
George