

Re: Explicitly calling the constructor

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2005-06/msg01266.html>

- *From:* "Antti Keskinen" <antti.keskinen@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 29 Jun 2005 10:06:31 +0300
-

Already solved.

The necessity was to place the scope operator in front of the constructor call, as in the following example:

```
pData->CTestClass::CTestClass();
```

"Antti Keskinen" <antti.keskinen@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:ebU%23fgHfFHA.3692@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

> Hello !

>

> I have a simple but interesting problem. I am using the 'malloc' routine
> to reserve memory for a class object, and then trying to call the object's
> constructor via the pointer returned. For some reason, this does not work
> at all, and I have no idea why.

>

> The reason why I need to do this is that I'm developing a COM component,
> and in there it's suggested to use CoTaskMemAlloc to reserve memory
> instead of the 'new keyword'. The object I'm trying to initialize is a
> complex class, thus I need to call it's constructor.

>

> Here is a code sample, to which Visual Studio .NET 2003 answers with error
> C2273:

>

> #include <stdlib.h>

> #include <malloc.h>

> #include <queue>

> class CTestClass

> {

> public:

> CTestClass()

> {

> m_nData = 0;

> }

> private:

> int m_nData;

> };

Re: Explicitly calling the constructor

```
>
> int main(void)
> {
> CTestClass* pData = (CTestClass*) malloc( sizeof(CTestClass) );
>
> if ( pData )
> {
> pData->CTestClass();
> free( pData );
> }
>
> return 0;
> }
>
```

- *Follow-Ups:*

- ◆ [Re: Explicitly calling the constructor](#)
◇ From: Tom Widmer [VC++ MVP]

- *References:*

- ◆ [Explicitly calling the constructor](#)
◇ From: Antti Keskinen

- Prev by Date: [Explicitly calling the constructor](#)
- Next by Date: [Re: Password Filter crashes LSASS.exe](#)
- Previous by thread: [Explicitly calling the constructor](#)
- Next by thread: [Re: Explicitly calling the constructor](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)