
Subject: Re: Difficulty with Class declaration
Posted by [mr_ped](#) on Tue, 06 Apr 2010 15:45:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:

```
void UDMS::CallToPopupsMember() {  
    Popups pop;  
    pop.Member();  
};
```

This will allocate local fresh new "pop" every time you call the function, then it will call "Member()" upon this fresh new instance, then it will get destroyed.

If you want persistent instance of Popups, it should be either member variable of UDMS definition, thus visible for every member function of UDMS, or you can use "static Popups pop;" which will make it visible only inside CallToPopupsMember() function, but it will be created only after first time call, later the already instantiated object will be used every next time.

If the "pop" should live outside of UDMS, you may consider to have only Popups *poppointer; as UDMS member variable, set to NULL by default, and set to something else by the owner of "pop" when the "pop" exists.

Then the function would work with pointer

```
void UDMS::CallToPopupsMember() {  
    if ( poppointer == NULL ) return; //nobody gave me pop yet  
    poppointer->Member(); //call Member() upon the given pop  
};
```

```
void UDMS::SetPop( Popups *_newpop = NULL ) { poppointer = _newpop; }
```

And the owner of pop will call the UDMS::SetPop with either it's address or with NULL before it does destroy the pop.

(this is quite complicated design pattern with pointers usage, which can be most of the time avoided, but I'm putting it here to show you different approach)
