
Subject: Re: C++ templated class referencing each other

Posted by [Lance](#) on Sun, 11 Oct 2020 02:25:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

This hack seems to compile in CLANG. I am sure there are smarter and more standard ways to serve your purpose.

Object.h

```
#ifndef _TestTemplate_Object_h_
#define _TestTemplate_Object_h_

#include "CompManager.h"
class Object{
public:
    Object() : componentManager(*this){
    }

    void SayHello(){ componentManager.CreateComponent<int>(false,100); }
    Object& GetObject(){ return *this;}
    /*
        Many public things
    */
private:
    ComponentManager componentManager;
};
#include "CompManagerEx.h"
#endif
```

CompManager.h

```
class Object;

class ComponentManager{
public:
    ComponentManager(Object& _object) : object(_object){}

    template <class T> T& CreateComponent(bool active, int position);

    /*
        Many templated function that force me to write declaration in this .h file
    */
private:
    Object& object;
};
```

CompManagerEx.h

```
template <class T> T&
ComponentManager::CreateComponent(bool active, int position)
{
    static T t;
    /*
        Working code
    */
    Object& obj=object.GetObject(); //This line is problematic
    LOG("Hello inside ComponentManager::CreateComponent(bool active, int position)");
    /*
        Working code
    */
    return t;
}
```

test.cpp

```
#include <Core/Core.h>

#include "Object.h"

using namespace Upp;

CONSOLE_APP_MAIN
{
    Object obj;
    obj.SayHello();
}
```
