
Subject: Re: No forward declaration.

Posted by [dolik.rce](#) on Sun, 22 Sep 2013 06:48:21 GMT

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

Welcome to the forum akspring

akspring wrote on Sun, 22 September 2013 00:39Hello,

I am a C# programmer with a problem.

//No .h! I only only using .cpp files in my packages. What is the point of .h if you are not porting your code// and use your own namespace?There is many points, for example it improves readability of the code and it solves some problems with dependencies between classes. Besides it is how the language was designed, if you don't like it, you can stay with C#

akspring wrote on Sun, 22 September 2013 00:391) Lzz, or Lazy -

<http://www.lazycplusplus.com/index.html>

Will offer to generate a .h and .cpp that I can use. I haven't played with it, but can I integrate it onto top of U++ for one click compiling? I will have to play with it later. Would hate to do much more than 1-2 clicks, and don't want to overwrite current source files, such as example above. However disposable output files from lazy would be great of course, as long as I don't see it. This could be integrated into TheIDE fairly easily. I just quickly looked at it, but if it works as advertised, you could just write .lzz files an set a custom build step or as a macro (see this page for details) to convert those into .h and .cpp automatically. No clicks required, it would happen when you run compilation.

akspring wrote on Sun, 22 September 2013 00:392) Rearrange code, so ParentClass is before ChildClass, as mentioned here: http://www.ultimatepp.org/forum/index.php?t=msg&goto=2289&&srch=law#msg_2289

This works if only one class is using the other, but not if they are both being used by each other! Not exactly solution for me. Does work for some of my classes, though I have to put them near the top of my file!Just as you say, it only works when there is one-sided dependency. Otherwise you need .h file or at least forward declaration of the classes.

akspring wrote on Sun, 22 September 2013 00:393) I have read about friend classes. Is that a solution to use?No, they'd still have to be declared. Friend functions/classes are only different from regular ones in access rights - they can use private variables/functions of the class they are friends with.

akspring wrote on Sun, 22 September 2013 00:394) Use a different compiler that supports this with some kind of preprocessing?I don't know about any such compiler. They all (not surprisingly) follow the C++ standards.

akspring wrote on Sun, 22 September 2013 00:395) This is the only solution that I have found

which seems to really work. I haven't tried it yet either:

```
class childClass //Thanks to guy who made this snippet
{
    parentClass* parent; //reference to class that created this child
    childClass(parentClass p)
    {
        parent = p;
    }
};
```

Somehow using a pointer resolves this, which I can partially understand.

The problem here is I am trying to convert C# code into C++ as an experiment. But while not so terrible compared to the other solutions I've found, it is not so feasible to add * to every declared member class of main class. I have many classes with similar dilemmas to this.

What does using * do exactly? It references a pointer to the class or where the class will be in memory, correct? Does appending * change the functionality of parent at all? Can I still use sub functions of parent member just fine? Or will I have to use *parent, or work with other limitations? Using pointer only postpones the trouble. I think the compiler would complain later in the code when you try to access the object pointed to, if it is not declared by then. Pointer in C++ is just an address in memory, so it can often lead to many problems, e.g. when it points to uninitialized data. In U++ we try to use them only when absolutely necessary. If your code in C# worked without pointers, it should be possible to use it without pointers in C++ as well.

akspring wrote on Sun, 22 September 2013 00:39 I would mention forward declaration, but that does not really give the functionality that I want in this case, and it only seems to work for certain cases that don't require return or use of B class within A class function. Or A class member inside a B class returning type A from function with params. Yes, forward declaration only works if you don't need to access members of the declared class.

akspring wrote on Sun, 22 September 2013 00:39 It is sad really, with all of the bloated stuff they are throwing into C++11/14, you would think this would make it considering all of the other crazy things. I just want some convenience! Even if underneath during compilation a separate .h and .cpp is generated and used. I guess nobody even asked for such thing. The C++ just works different then Java/C# and we like that

akspring wrote on Sun, 22 September 2013 00:39 I am wondering if there is a frontend or compiler that does such preprocessing for this scenario. If not, I am highly tempted to make it! Of course I can't program yet, but that is my problem, haha! The Lazy C++ you mentioned to be such a preprocessor...

akspring wrote on Sun, 22 September 2013 00:39 Hey, you know if you really wanted to make U++ famous around the world.... wink! The IDE provides some helping tools for .h maintenance, which are IMHO quite enough. Check Alt+C, Ctrl+V - it is like copying, but if you "copy" a declaration of a function(s), it is turned into definition(s) and vice versa. It saves a lot of time.

All that said, I'd still recommend you to create the header files if you really want to use C++, it might be the easiest way in the end

Best regards,
Honza
