
Subject: No forward declaration.

Posted by [akspring](#) on Sat, 21 Sep 2013 22:39:47 GMT

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

Hello,

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?

```
#include <CtrlLib/Core> //<< Just example to make this snippet seem more legitimate.
```

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

class parentClass
{
    childClass child(this);
};
```

This sort of thing compiles in C# / Java. But not in C++.

This is what I have found so far:

1) 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.

2) 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!

- 3) I have read about friend classes. Is that a solution to use?
- 4) Use a different compiler that supports this with some kind of preprocessing?
- 5) 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?

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.

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 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!

Hey, you know if you really wanted to make U++ famous around the world.... wink!
