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=228 9&&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!

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 00:39I 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 00:39It 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 00:39I 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 00:39Hey, you know if you really wanted to make U++ famous around the world.... wink!TheIDE 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

Subject: Re: No forward declaration.
Posted by akspring on Mon, 23 Sep 2013 07:53:19 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 22 September 2013 08:48Welcome to the forum akspring

Thank you dolik.rce, I feel very fortunate to be here Maybe you have asked a technical question online at microsoft or stackoverflow and the only responses you get don't answer any of your questions. But already I am getting help from a pro, such great customer service this place provides!

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

I am a C# programmer with a problem.

Thank you for not commenting on this

dolik.rce wrote on Sun, 22 September 2013 08:48
There is many points, for example it improves readability of the code and it solves some problems with dependencies between classes.

Yes, I agree. For the record, just so no one destroys my identity, I have no problems with header files and how other C++ programs are built. I just really like the idea of classes being part of a namespace, and residing in a single .cpp file. If you want to write members or functions of the class in a different file, you can just reference the namespace and prefix the class with the partial keyword. Add partial to both files and now the compiler knows what you are trying to do and how to group classes before building. It makes sense to me, and for me, makes grouping and compilation much easier. If I could take that from C# and extend it to C++, I think C++ would benefit from it very well.

Well, maybe not benefit so much, since it is not doing much more than working around .h and .cpp files, but it would make life easier for people like me!

dolik.rce wrote on Sun, 22 September 2013 08:48Besides it is how the language was designed, if you don't like it, you can stay with C#

N...Nooo!!
I am sorry! I didn't mean it like that! It was a misunderstanding!

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.
dolik.rce wrote on Sun, 22 September 2013 08:48This 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.

Thank you again dolik.rce
I haven't played with this yet but I suspect this is what I will be doing, if it solves my ParentClass ChildClass problems. Already I have a solution - what a great forum this is!

akspring wrote on Sun, 22 September 2013 00:393) I have read about friend classes. Is that a solution to use?
dolik.rce wrote on Sun, 22 September 2013 08:48
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.
I was looking at examples of these and I noticed they didn't seem to cause errors without using prototyping and when using class types as a return variable for class functions. No problem though, I will only use this if needed. So far I can still use my terrible C# habits with lazy I think.

akspring wrote on Sun, 22 September 2013 00:394) Use a different compiler that supports this with some kind of preprocessing?
dolik.rce wrote on Sun, 22 September 2013 08:48I don't know about any such compiler. They all (not surprisingly) follow the C++ standards.
Hm I do find this surprising actually. Looking at a list of compilers... There is GCC, MinGW.GCC, G++, TDM, LD or VS2003, 2005, 2008, 2010, 2012, Intel, Open Watcom?, Borland... And I think I am still missing quite a few. I don't blame them of course for following standards, but I would expect some optional features from some of them, like "-supportlazy" or "-supportak" or "-badprogrammer" or something. Whatever lets me keep my code in .cpp files and avoid using headers to reference my own .cpp files. I just want convenience, but with the C++!

But it looks like Lazy does this, I just have to play with it. I will update this when I figure it out.

dolik.rce wrote on Sun, 22 September 2013 08:48
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 00:39I 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 00:39It 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 00:39I 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 00:39Hey, you know if you really wanted to make U++ famous around the world.... wink!TheIDE 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

I looked at Alt+C, I read about it before but didn't quite understand it. Now I do! Great little macro. Unfortunately, I tried creating a .h file with it, but couldn't get it to work in my test program. I must be doing something wrong... But I think being lazy is the fix, so I will try that first

I will post my test .h/.cpp code if I can't get lzz working.
Thanks Honza, and what a great place this forum is! Do I owe anyone a check?

Subject: Re: No forward declaration.
Posted by dolik.rce on Mon, 23 Sep 2013 08:35:07 GMT
View Forum Message <> Reply to Message

akspring wrote on Mon, 23 September 2013 09:53akspring wrote on Sun, 22 September 2013 00:39Hello,

I am a C# programmer with a problem.

Thank you for not commenting on this I was very tempted to comment on this, but I decided not to scare a newcomer right away

akspring wrote on Mon, 23 September 2013 09:53I would expect some optional features from some of them, like "-supportlazy" or "-supportak" or "-badprogrammer" or something.This made me laugh a lot

akspring wrote on Mon, 23 September 2013 09:53I looked at Alt+C, I read about it before but didn't quite understand it. Now I do! Great little macro. Unfortunately, I tried creating a .h file with it, but couldn't get it to work in my test program. I must be doing something wrong...It doesn't create the entire .h file, only convert methods. You'd still have to write all the includes, class declaration itself etc.

akspring wrote on Mon, 23 September 2013 09:53Thanks Honza, and what a great place this forum is! Do I owe anyone a check? If you're really that enthusiastic, there is a paypall account

Honza

Subject: Re: No forward declaration.
Posted by helloworld on Tue, 03 Mar 2015 01:58:25 GMT
View Forum Message <> Reply to Message

I am having a similar problem as well. Using Windows, I cannot get lazy to function correctly. It appears to have a bug that prevents .h dependencies from registering and recursively seeks the same files without finding them. If I wanted to address the issues above regarding building code without headers or generating headers on the fly,how much would it cost for a more professional solution?