blog@jatinganhotra

render :object => Experience.new

- RSS

Search

- Blog
- Archives
- Resume

Forward Class Declaration in C++

Nov 25th, 2012 | Comments

“In computer programming, a forward declaration is a declaration of an identifier (denoting an entity such as a type, a variable, or a function) for which the programmer has not yet given a complete definition.”

In C++, you should forward declare classes instead of including headers. Don’t use an #include when a forward declaration would suffice. When you include a header file you introduce a dependency that will cause your code to be recompiled whenever the header file changes. If your header file includes other header files, any change to those files will cause any code that includes your header to be recompiled. Therefore, you should prefer to minimize includes, particularly includes of header files in other header files.

You can significantly reduce the number of header files you need to include in your own header files by using forward declarations. For example, if your header file uses the File class in ways that do not require access to the declaration of the File class, your header file can just forward declare class File; instead of having to #include “file.h”.

This will in turn speed a little bit the compilation.

When can I use the forward class declaration?
Put yourself in the compiler’s position: when you forward declare a type, all the compiler know is that this type exists; it knows nothing about its size, members

or methods. This is why it's called an *incomplete type*. Therefore, you cannot use the *incomplete* type at places where the compiler would need to know the layout of the type.

Assuming the following forward declaration `class Forward;`

You can use forward declarations in all the following cases, where

```
class Master {
private:
    // Declare a member to be a pointer or a reference to the incomplete type
    Forward *ptr1;
    Forward &ptr2;

public:

    // Declare functions or methods which accepts/return incomplete types:
    void ByValue(Forward by_value);
    Forward OrReturnValue();

    // Define functions or methods which accepts/return pointers/references
    // to the incomplete type (but without using its members):
    void OrByPointer(Forward* by_pointer);
    void OrByReference(const Forward& by_reference);
    Forward& ReturnByRef();
    Forward* ReturnByPointer();

};
```

Master.h hosted with ❤ by **GitHub** **view raw**

You can't use forward declarations in all the following cases, where

```
// Use it as a base class
class Master : Forward {} // compiler error!

```

```
// Use it to declare a member:
class Master {
    Forward cannot_define_obj_member; // compiler error!
};

// Define functions or methods using the incomplete type
void SomeFunc1(Forward x) {} // compiler error!
Forward    SomeFunc2()    {} // compiler error!

// Use its methods or fields, in fact trying to dereference a variable with incomplete type
class Master {
    Forward *forward_class_ptr;
    void SomeFunc()
    {
        forward_class_ptr->SomeMethod();      // compiler error!
        int i = forward_class_ptr->SomeField; // compiler error!
    }
};

// Whenever you use a class as a template parameter, the declaration of that class must be complete and not simply forward declared.
class Master {
      std::vector<Forward> some_stl_containers_;
};


// Delete the forward declared class in the module
class Forward;
class Master
{
    void SomeFunc(Forward* forward_class_ptr)
    {
        delete forward_class_ptr;
```

```
36      }
37  };
38  // From the C++ standard: 5.3.5/5:
39  // "If the object being deleted has incomplete class type at the point of deletion and the complete class has a // non-trivial destructor
40  // e.g gcc 4.2.1 errors out and reports a note: neither the destructor nor the class-specific operator delete will be called, even if the
```

Master.h hosted with ❤ by **GitHub** **view raw**

Posted by Jatin Ganhotra (@jatinganhotra) Nov 25th, 2012 C++, Coding Tips, Forward Class Declaration

« Why use GIT and hang CVS? Add Google Website Translator to your Octopress Blog »

Comments

1 • Reply • Share ›

Jatin Ganhotra Mod ➚ Lee Zhen Yong • 4 years ago

@Lee Zhen Yong Thanks. Hope you found it useful :)

1 • Reply • Share ›

Lee Zhen Yong ➚ Jatin Ganhotra • 4 years ago

I didn't realize you updated this article after I commented(they have this subcribe via email which is not enabled by default after I comment)! Your code additions are actually more detailed than that reply from stackoverflow. Neat! :D

1 • Reply • Share ›

Jatin Ganhotra Mod ➚ Lee Zhen Yong • 4 years ago

No, I didn't update it after your comment. It's in the same original state, when I made this post and hasn't gone any changes since then.

May be, you missed out on some of the code due to incomplete rendering of the gist embedded. I'll take a look at it.

Anyways, I'm glad that you liked it. If there's any other topic you'd like me to blog about, let me know. :)

• Reply • Share ›

Lee Zhen Yong ➚ Jatin Ganhotra • 4 years ago

O you're right. I was loading this on my ipad and realize the final code block didn't rendered out. Just needed to refresh though... Which I didn't do last time. Was like wondering, "hey why didn't Jatin finish his post?" Lol.

Ah and thanks for the blogging offer (:

• Reply • Share ›

Jatin Ganhotra Mod ➚ Lee Zhen Yong • 4 years ago

Peace :)

1 • Reply • Share ›

surbhijain93 • a year ago

@Jatin Ganhotra can you please explain why we can forward declare in the cases you mentioned and why can't we do the same in the other cases? Thanks

• Reply • Share ›

Jatin Ganhotra Mod ➚ surbhijain93 • a year ago

@surbhijain93 Put yourself in the compiler's position: when you forward declare a type, all the compiler know is that this type exists; it knows nothing about its size, members or methods. This is why it's called an incomplete type. Therefore, you cannot use the incomplete type at places where the compiler would need to know the layout of the type.

1 ^ v • Reply • Share ›

surbhijain93 ➚ Jatin Ganhotra • a year ago

Thanks

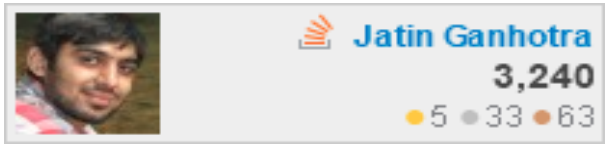
^ v • Reply • Share ›

About Me

Graduate student in Computer Science at University of Illinois at Urbana Champaign - Knows C/C++, Ruby, Ruby on Rails and a bit of Android.

Resume | @jatinganhotra on GitHub

Stack Overflow



Recent Posts

- Installing Octave on OS X 10.9 Mavericks
- Comparison is always false due to limited range of data type
- Keyboard Review - Microsoft Natural Ergonomic Keyboard 4000
- From Disqus to IntenseDebate & back again
- Concurrent and Sequential statements in Verilog
- C++ - Variable Declaration in 'if' expression

Latest Tweets

Tweets by @JatinGanhotra

Jatin Ganhotra
@JatinGanhotra

“Gathering information with IBM Watson Conversation” by @snrubnomis
medium.com/@snrubnomis/ga…

Gathering information with IBM Watson Conversation – Simon Burns – Medium
IBM Watson Conversation is a service that enables the building of chatbots, and oth…
medium.com

25 Apr

Jatin Ganhotra Retweeted

Embed View on Twitter

[Google+]