

GeeksforGeeks

A computer science portal for geeks



Courses

Login

**Write an
Article**



Access

Modifiers in
C++

Inheritance
in C++

Polymorphism
in C++

Encapsulation
in C++

Abstraction
in C++

Structure vs
class in C++

Can a C++
class have
an object of
self type?

Why is the
size of an
empty class
not zero in
C++?

Static data
members in
C++



Some
interesting
facts about
static
member
functions in
C++

Friend class
and function
in C++

Local
Classes in
C++

Nested
Classes in
C++

Constructors
in C++

Copy
Constructor
in C++

Destructors
in C++

Does C++
compiler
create
default
constructor
when we
write our
own?

When should
we write our



own copy
constructor?

When is copy
constructor
called?

Initialization
of data
members

Use of
explicit
keyword in
C++

When do we
use_INITIALIZER
List in C++?

C++ Internals
| Default
Constructors
| Set 1

Private
Destructor

Playing with
Destructors
in C++

int (1 sign bit
+ 31 data
bits)
keyword in C

Program
error signals

Why array



index starts
from zero ?

Dynamic
Memory
Allocation in
C using
malloc(),
calloc(),
free() and
realloc()

TCP Server-
Client
implementation
in C



Simulating final class in C++

Ever wondered how can you design a class in C++ which can't be inherited. Java and C# programming languages have this feature built-in. You can use `final` keyword in java, `sealed` in C# to make a class non-extendable.

Below is a mechanism using which we can achieve the same behavior in C++. It makes use of private constructor, virtual inheritance and friend class.

In the following code, we make the *Final* class non-inheritable. When a class *Derived* tries to inherit from it, we get compilation error.

An extra class *MakeFinal* (whose default constructor is private) is used for our purpose. Constructor of *Final* can call private constructor of *MakeFinal* as *Final* is a friend of *MakeFinal*.

Note that *MakeFinal* is also a virtual base class. The reason for this is to call the con-

structor of *MakeFinal* through the constructor of *Derived*, not *Final* (The constructor of a virtual base class is not called by the class that inherits from it, instead the constructor is called by the constructor of the concrete class).



```

/* A program with compilation error to demonstrate that Final cl
   be inherited */
#include<iostream>
using namespace std;

class Final; // The class to be made final

class MakeFinal // used to make the Final class final
{
private:
    MakeFinal() { cout << "MakFinal constructor" << endl; }
friend class Final;
};

class Final : virtual MakeFinal
{
public:
    Final() { cout << "Final constructor" << endl; }
};

class Derived : public Final // Compiler error
{
public:
    Derived() { cout << "Derived constructor" << endl; }
};

int main(int argc, char *argv[])
{
    Derived d;
    return 0;
}

```

Output: *Compiler Error*





```

In constructor 'Derived::Derived()':
error: 'MakeFinal::MakeFinal()' is private

```

In the above example, *Derived's* constructor directly invokes *MakeFinal's* constructor and the constructor of *MakeFinal* is private, therefore we get the compilation error.

You can create the object of *Final* class as it is friend class of *MakeFinal* and has access to its constructor. For example, the following program works fine.



```
/* A program without any compilation error to demonstrate that i
the Final class can be created */
#include<iostream>
using namespace std;

class Final;

class MakeFinal
{
private:
    MakeFinal() { cout << "MakeFinal constructor" << endl; }
    friend class Final;
};

class Final : virtual MakeFinal
{
public:
    Final() { cout << "Final constructor" << endl; }
};

int main(int argc, char *argv[])
{
    Final f;
    return 0;
}
```

Output: *Compiles and runs fine*

```
MakeFinal constructor
Final constructor
```

This article is compiled by Gopal Gorthi and reviewed by GeeksforGeeks team. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



Recommended Posts:

C++ final specifier

std::any Class in C++



Structure vs class in C++

Array class in C++

std::string class in C++

std:: valarray class in C++

Why is the size of an empty class not zero in C++?

Can a C++ class have an object of self type?

C++ String Class and its Applications | Set 2

Difference between namespace and class

What all is inherited from parent class in C++?

C++ string class and its applications

C++ | Class and Object | Question 5

C++ | Class and Object | Question 4

Friend class and function in C++

> CI DONE BETTER

Article Tags : C C++

Practice Tags : C CPP



1

☐ To-do ☐ Done

4.1

Based on **43** vote(s)

Feedback

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.



Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!



710-B, Advant Navis Business Park,
Sector-142, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE

Write an Article
Write Interview
Experience
Internships
Videos

@geeksforgeeks, Some rights reserved

