Some interesting facts about static member functions in





Hire with**Login**

മ C++ Classes and Objects 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 ype? Why is the size of an empty class not zero in C++? Static data members in C++

25/01/20, 1:17 am

C++

Friend class and function in C++

Local Classes in C++

Nested Classes in C++

Simulating final class 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

2 of 7

Use of explicit keyword in C++

When do we use Initializer List in C++?

C++ Internals | Default Constructors | Set 1

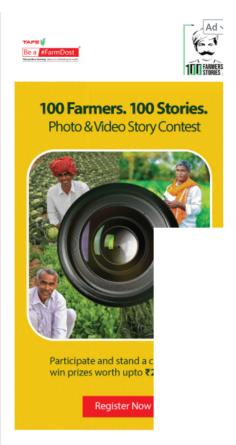
Trie Data Structure using smart pointer and OOP in C++

C++ program to compare two Strings using Operator Overloading

Derived Data Types in C++

Go vs C++

Pointers and References in C++



Nested Classes in C++

A nested class is a class which is declared in another enclosing class. A nested class is a member and as such has the same access rights as any other member. The members of an enclosing class have no special access to members of a nested class; the usual access rules shall be obeyed.

For example, program 1 compiles without any error and program 2 fails in compilation.

→ Program 1



```
#include<iostream>
   using namespace std;
    /* start of Enclosing class declaration */
   class Enclosing {
      private:
          int x;
      /* start of Nested class declaration */
      class Nested {
          int y;
         void NestedFun(Enclosing *e) {
           cout<<e->x; // works fine: nested class can access
                         // private members of Enclosing class
      }: // declaration Nested class ends here
   }; // declaration Enclosing class ends here
   int main()
   }
```

Program 2

References:

```
#include<iostream>

/ using namespace std;

/* start of Enclosing class declaration */
class Enclosing {

   int x;

   /* start of Nested class declaration */
   class Nested {
      int y;
   }; // declaration Nested class ends here

   void EnclosingFun(Nested *n) {
      cout<<n->y; // Compiler Error: y is private in Nested
   }
}; // declaration Enclosing class ends here

int main()
{
}
```

http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2005/n1905.pdf

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Sign Up on Into

Ad Earn & Redee Free Flights, Hote

InterMiles

Sign Up

Recommended Posts:

Virtual functions in derived classes

Catching base and derived classes as exceptions

Can namespaces be nested in C++?

Local Classes in C++

Pure Virtual Functions and Abstract Classes in C++

C++ Classes and Objects

Decision Making in C / C++ (if , if..else, Nested if, if-else-if)

Trivial classes in C++

Anonymous classes in C++

File Handling through C++ Classes

Nested list in C++ STL

C++ | Nested Ternary Operator

Nested switch statement in C++

C++ Stream Classes Structure

Enum Classes in C++ and Their Advantage over Enum DataType

Improved By: Akshit Agarwal 3

Article Tags: C++

Practice Tags: CPP



	2.4	
To-do Done	Based on 25 vote(s)	
Feedback/ Suggest Improvement Add Note	es 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		

Geeksfor Geeks
A computer science portal for geeks

5th Floor, A-118, Sector-136, Noida, Uttar Pradesh - 201305 feedback@geeksforgeeks.org

COMPANY	LEARN	PRACTICE	CONTRIBUTE
About Us	Algorithms	Courses	Write an Article
Careers	Data Structures	Company-wise	Write Interview Experience
Privacy Policy	Languages	Topic-wise	Internships
Contact Us	CS Subjects	How to begin?	Videos
	Video Tutorials		

--**>**

@geeksforgeeks, Some rights reserved