



namespace in C++ | Set 2 (Extending namespace and Unnamed namespace)

We have introduced namespaces in below set 1.

[Namespace in C++ | Set 1 \(Introduction\)](#)

It is also possible to create more than one namespaces in the global space. This can be done in two ways.

- **namespaces having different names**

```

// A C++ program to show more than one namespaces
// with different names.
#include <iostream>
using namespace std;

// first name space
namespace first
{
    int func() { return 5; }
}

// second name space
namespace second
{
    int func() { return 10; }
}

int main()
{
    // member function of namespace
    // accessed using scope resolution operator
    cout << first::func() << "\n";
    cout << second::func() << "\n";
    return 0;
}

```

Output:

```

5
10

```

- **Extending namespaces (Using same name twice)**

It is also possible to create two namespace blocks having the same name. The second namespace block is nothing but actually the continuation of the first namespace. In simpler words, we can say that both the namespaces are not different but actually the same, which are being defined in parts.

```
// C++ program to demonstrate namespace extension
#include <iostream>
using namespace std;

// first name space
namespace first
{
    int val1 = 500;
}

// rest part of the first namespace
namespace first
{
    int val2 = 501;
}

int main()
{
    cout << first::val1 << "\n";
    cout << first::val2 << "\n";
    return 0;
}
```

Output:

```
500
501
```

Unnamed Namespaces

- They are directly usable in the same program and are used for declaring unique identifiers.
- In unnamed namespaces, name of the namespace is not mentioned in the declaration of namespace.
- The name of the namespace is uniquely generated by the compiler.
- The unnamed namespaces you have created will only be accessible within the file you created it in.
- Unnamed namespaces are the replacement for the static declaration of variables.

```
// C++ program to demonstrate working of unnamed
// namespaces
#include <iostream>
using namespace std;

// unnamed namespace declaration
namespace
{
    int rel = 300;
}

int main()
{
    cout << rel << "\n"; // prints 300
    return 0;
}
```

Output:

300

This article is contributed by **Abhinav Tiwari** .If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

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

Recommended Posts:

Cons of using the whole namespace in C++

Namespace in C++ | Set 1 (Introduction)

Why "using namespace std" is considered bad practice

Difference between namespace and class

Namespace in C++ | Set 3 (Accessing, creating header, nesting and aliasing)

fill in C++ STL

Conditional or Ternary Operator (?:) in C/C++

forward_list insert_after() function in C++ STL

Count of distinct remainders when N is divided by all the numbers from the range [1, N]

C++ | asm declaration

Pointers and References in C++

Strings in C++ and How to Create them?

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

Deque vs Vector in C++ STL

Article Tags : [C](#) [C++](#) [cpp-namespaces](#)

Practice Tags : [C](#) [CPP](#)



4

1.9

☐

To-do

☐

Done

Based on **13** vote(s)

Feedback/ Suggest Improvement

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

5th Floor, A-118,
Sector-136, 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

[Courses](#)
[Company-wise](#)
[Topic-wise](#)
[How to begin?](#)

CONTRIBUTE

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)



@geeksforgeeks, Some rights reserved