



Stack in C++ STL

Stacks are a type of container adaptors with LIFO(Last In First Out) type of working, where a new element is added at one end and (top) an element is removed from that end only.

The functions associated with stack are:

empty() – Returns whether the stack is empty – Time Complexity : O(1)
size() – Returns the size of the stack – Time Complexity : O(1)
top() – Returns a reference to the top most element of the stack – Time Complexity : O(1)
push(g) – Adds the element 'g' at the top of the stack – Time Complexity : O(1)
pop() – Deletes the top most element of the stack – Time Complexity : O(1)

```
// CPP program to demonstrate working of STL stack
#include <iostream>
#include <stack>
using namespace std;

void showstack(stack <int> s)
{
    while (!s.empty())
    {
        cout << '\t' << s.top();
        s.pop();
    }
    cout << '\n';
}

int main ()
{
    stack <int> s;
    s.push(10);
    s.push(30);
    s.push(20);
    s.push(5);
    s.push(1);

    cout << "The stack is : ";
    showstack(s);

    cout << "\ns.size() : " << s.size();
    cout << "\ns.top() : " << s.top();

    cout << "\ns.pop() : ";
    s.pop();
    showstack(s);

    return 0;
}
```

Output:

```
The stack is :      1      5      20      30      10

s.size() : 5
s.top() : 1
s.pop() :      5      20      30      10
```

List of functions of Stack:

- **stack::top()** in C++ STL

- [stack::empty\(\) and stack::size\(\) in C++ STL](#)
- [stack::push\(\) and stack::pop\(\) in C++ STL](#)
- [stack::swap\(\) in C++ STL](#)
- [stack::emplace\(\) in C++ STL](#)
- [Recent Articles on C++ Stack](#)

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



Recommended Posts:

- [stack empty\(\) and stack size\(\) in C++ STL](#)
- [Sort a stack using a temporary stack](#)
- [stack top\(\) in C++ STL](#)
- [stack emplace\(\) in C++ STL](#)
- [stack swap\(\) in C++ STL](#)
- [stack push\(\) and pop\(\) in C++ STL](#)
- [Stack Unwinding in C++](#)
- [Sort a stack using recursion](#)
- [Sudo Placement\[1.3\] | Stack Design](#)
- [Delete middle element of a stack](#)
- [Check if the elements of stack are pairwise sorted](#)
- [Level order traversal in spiral form | Using one stack and one queue](#)
- [Design a stack that supports getMin\(\) in O\(1\) time and O\(1\) extra space](#)
- [Design and Implement Special Stack Data Structure | Added Space Optimized Version](#)
- [Heap overflow and Stack overflow](#)

3 EASY STEPS TO GET MAC SPEED BACK

1. [Click Start Now](#)
2. Install MacKeeper Easily
3. Enjoy the Must-Have Mac Optimizer

[START NOW](#)

MACKEEPER®

Article Tags : [C++](#) [cpp-containers-library](#) [cpp-stack](#) [cpp-stack-functions](#) [STL](#)

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



30

1.6

To-do Done

Based on 90 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](#)

GeeksforGeeks

A computer science portal for geeks

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