**HOW TO USE STACK IN STANDARD TEMPLATE LIBRARY**

|  |
| --- |
| **Stacks** are a type of **container adaptor**, which is implemented to operate in a LIFO (Last in, First out). Elements are pushed/popped from the "back" of the specific container, which is known as the top of the stack. |

**Declare:** #include <stack>

|  |
| --- |
| #include <stack>  **Template:** stack<value\_type> stack\_name; |

**Member function:**

- size: return the current size of the stack. Complexity: O(1).

- empty: return true if stack is empty; otherwise, return false. Complexity: O(1).

- push: insert an element into the stack. Complexity: O(1).

- pop: remove the element on the top of the stack. Complexity: O(1).

- top: return the value of element at the top of the stack. Complexity: O(1).

**Demo program:**

|  |
| --- |
| #include <iostream>  using namespace std;  #include <stack>  int main() {  stack<int> s;  for (int i = 1; i <= 5; ++i) s.push(i); // s = {1, 2, 3, 4, 5}  s.push(10); // s = {1, 2, 3, 4, 5, 10}  cout << s.size() << endl; // Print out on the screen: 6  cout << s.top() << endl; // Print out on the screen: 10  s.pop(); // s = {1, 2, 3, 4, 5}  cout << s.empty() << endl; // Print out: 0 (since stack s is not empty).  return 0;  } |