STL Stack in C++

`stack` is a **container adapter** in the C++ Standard Template Library (STL) that provides **LIFO (Last-In-First-Out)** functionality. You can only insert (`push`) and remove (`pop`) elements from the top of the stack.

---

1. Include Header

```cpp
#include <stack>
#include <iostream>
using namespace std;
```

---

2. Creating a Stack

```cpp
stack<int> s1;              // Empty stack of integers
stack<string> s2;           // Empty stack of strings
```

---

3. Common Stack Operations with Examples

3.1. Push Elements

```cpp
stack<int> s;

s.push(10);

s.push(20);

s.push(30);
```

Time Complexity: **O(1)** (Amortized)

3.2. Pop Element

```cpp
s.pop(); // Removes the top element (30)
```

Note: `pop()` does not return the popped value. Access it using `top()` before popping.

3.3. Top Element

```cpp
cout << s.top() << endl; // Prints the current top
```

3.4. Check Size

```
cout << s.size(); // Number of elements in stack
```

3.5. Check If Empty

```
if (s.empty())
    cout << "Stack is empty";
```

---

4. Print and Empty the Stack

```
while (!s.empty())
{
    cout << s.top() << " ";
    s.pop();
}
```

---

5. Stack Limitations

- No iterators (unlike vector)
- Cannot access elements other than top
- Cannot traverse using range-based for loop

6. Practice Problems

1. https://codeforces.com/group/c3FDl9EUi9/contest/263096/problem/D
2. Valid Parentheses in an Expression - GeeksforGeeks

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    string s; cin>>s;
    stack<char> st;
    for(char c: s)
    {
        if(c=='(' || c=='{' || c=='[')
        {
            st.push(c);
        }
        else
        {
            if(c==')' && st.top()=='(')
```

```
            {
                st.pop();
            }
            else if(c=='}' && st.top()=='{')
            {
                st.pop();
            }
            else if(c==']' && st.top()=='[')
            {
                st.pop();
            }
            else
            {
                st.push(c);
            }
        }
    }
    if(st.empty()) cout<<"Valid";
    else cout<<"Invalid";
    return 0;
}
```

3. Reverse a String using Stack - GeeksforGeeks
4. Delete middle element of a stack - GeeksforGeeks

```
class Solution {
  public:
    // Function to delete middle element of a stack.
    void deleteMid(stack<int>& s) {
        // code here..
        stack<int> s2;
        int c=-1;
        int n = s.size();
        while(!s.empty())
        {
            c++;
            if(c==(n/2))
            {
                s.pop();
                continue;
            }
            s2.push(s.top());
            s.pop();
```

```
        }
        while(!s2.empty())
        {
            s.push(s2.top());
            s2.pop();
        }
    }
};
```

5. Reverse individual words - GeeksforGeeks
6. Next Greater Element (NGE) for every element in given Array - GeeksforGeeks

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    vector<int> v(n);
    for (auto &x : v)
        cin >> x;

    vector<int> ans(n);
    stack<int> st;

    for(int i=n-1;i>=0;i--)
    {
        while(!st.empty() && st.top() <= v[i])
        {
            st.pop();
        }
        if(st.empty()) ans[i]=-1;
        else ans[i]=st.top();
        st.push(v[i]);
    }

    for(auto x: ans) cout<<x<< " ";
    return 0;
}
```

7. Sort a stack using a temporary stack - GeeksforGeeks

```
#include <bits/stdc++.h>
using namespace std;
```

```cpp
int main()
{
    int n;
    cin>>n;
    stack<int> input;
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        input.push(x);
    }

    stack<int> sorted;
    while(!input.empty())
    {
        int top=input.top();
        input.pop();
        while(!sorted.empty() && sorted.top()<top)
        {
            input.push(sorted.top());
            sorted.pop();
        }
        sorted.push(top);
    }

    while(!sorted.empty())
    {
        cout<<sorted.top()<<" ";
        sorted.pop();
    }
    return 0;
}
```