C++ STL :

*C++ STL STACK:*

Function:

1. ‘empty’ : return true if the stack is empty, else return false -> O(1)
2. ‘push’ : push element into the stack -> O(1)
3. ‘pop’ : delete the reference of the top element [ actually just lower the index by 1] -> O(1)
4. ‘top’ : return the reference of the top element -> O(1)
5. ‘swap’ : swap the elements of two stack -> O(1)
6. ‘size’ : return the size of the stack -> O(1)

Usage:

**int main()**

**{**

**//freopen("output.txt", "w", stdout);**

**vector<string>vec( 5, "i am tanveer");**

**stack< int > s1;**

**stack< int, vector<int> > s2; //u can use this kind of initialization if the container (vector/deque/ list) has pop\_back, push\_back function**

**stack< string, vector<string> > s3(vec); //initiallize with the vector**

**stack< int > s4;**

**for( int i = 0; i < 5; i++ )**

**{**

**s1.push(i); //usage of push**

**s2.push(i\*i);**

**}**

**cout << "s1 top---- ";**

**cout << s1.top() << " "; // top element**

**printf("%d ", s1.top()); // using printf**

**puts("");**

**cout << "size of s1 before swapping with s4---- " << s1.size() << endl;**

**s1.swap(s4); // swap s1 and s2**

**cout << "size of s1 after swapping with s4---- " << s1.size() << endl;**

**cout << "s4 previously empty but after swapping with s1---- ";**

**while( s4.size() > 0 ) // chack s2 size**

**{**

**auto x = s4.top();**

**printf("%d ", x); // top element using printf**

**cout << typeid(x).name() << " -> ";**

**s4.pop();**

**}**

**cout << endl;**

**cout << "s3---- ";**

**while( !s3.empty() )**

**{**

**auto x = s3.top();**

**s3.pop();**

**cout << x << endl;**

**}**

**return 0;**

**}**

Output:

s1 top---- 4 4

size of s1 before swapping with s4---- 5

size of s1 after swapping with s4---- 0

s4 previously empty but after swapping with s1---- 4 i -> 3 i -> 2 i -> 1 i -> 0 i ->

s3---- i am tanveer

i am tanveer

i am tanveer

i am tanveer

i am tanveer