

Stack Implementation in C++

A Stack is a data structure which works on the principle LIFO (Last In, First Out). The basic operations in a Stack are :

1. Push : In which we push the data into the stack.
2. Pop : In which we remove an element from the Stack.

All insertions and removals are done only from one side of the Stack , which is called the 'top' of the Stack.

A stack is generally used in function calls where the local variables are pushed onto the Stack and when the function returns , it pops the variables from the Stack.

This is a simple implementation of Stack in C++.

```
1  #include <iostream>
2
3  using namespace std;
4
5  class Stack
6  {
7  private:
8      int *p;
9      int top,length;
10
11 public:
12     Stack(int = 0);
13     ~Stack();
14
15     void push(int);
16     int pop();
17     void display();
18 };
19
20 Stack::Stack(int size)
21 {
22     top=-1;
23     length=size;
24     if(size == 0)
25         p = 0;
26     else
27         p=new int[length];
28 }
29
30 Stack::~~Stack()
31 {
32     if(p!=0)
33         delete [] p;
34 }
35
36 void Stack::push(int elem)
37 {
38     if(p == 0)                //If the stack size is
```

```
39     {
40         cout<<"Stack of zero size"<<endl;
41         cout<<"Enter a size for stack : ";
42         cin >> length;
43         p=new int[length];
44     }
45     if(top==(length-1))    //If the top reaches to t
46     {
47         cout<<"\nCannot push "<<elem<<" , Stack full"<
48         return;
49     }
50     else
51     {
52         top++;
53         p[top]=elem;
54     }
55 }
56 int Stack::pop()
57 {
58     if(p==0 || top==-1)
59     {
60         cout<<"Stack empty!";
61         return -1;
62     }
63     int ret=p[top];
64     top--;
65     length--;
66
67     return ret;
68 }
69
70 void Stack::display()
71 {
72     for(int i = 0; i <= top; i++)
73         cout<<p[i]<<" ";
74     cout<<endl;
75 }
76
77 int main()
78 {
79     Stack s1;    //We are creating a stack o
80     s1.push(1);
81     s1.display();
82     s1.push(2);
83     s1.push(3);
84     s1.push(4);
85     s1.push(5);
86     s1.display();
87     s1.pop();
88     s1.display();
89     s1.pop();
90     s1.display();
91     s1.pop();
92     s1.display();
93     s1.pop();
94     s1.display();
95 }
```

Labels: [Data Structures](#)

3

[View comments](#)**Vikas Taneja** [June 25, 2012 at 1:19 PM](#)

Good and simple implementation! :)

You can make use of templates to make the stack class as a generic one. Small improvement though. :)

[Reply](#)**zem LoveJC** [May 28, 2013 at 10:35 PM](#)

i would like to know whether the method display() does conform to the basic format or basic structure of stacks, does it print in reverse of the order of input?

[Reply](#)**Varun Gupta** [May 28, 2013 at 10:43 PM](#)

It prints in the bottom-top order. Its not in the conventional manner.

[Reply](#)