

```

1  #include <iostream>
2  using namespace std;
3
4  // implementing the Stack ADT using array
5
6
7  class Stack{
8
9      private:
10         int *array;
11         int maxSize; // useful to decide if resizing (doubling the array size) is needed
12         int topOfStack; // same as endOfArray
13
14     public:
15         Stack(int size){
16             array = new int[size];
17             maxSize = size;
18             topOfStack = -1;
19         }
20
21         bool isEmpty(){
22
23             if (topOfStack == -1)
24                 return true;
25
26             return false;
27         }
28
29         void resize(int s){
30
31             int *tempArray = array;
32
33             array = new int[s];
34
35             for (int index = 0; index < min(s, topOfStack+1); index++){
36                 array[index] = tempArray[index];
37             }
38
39             maxSize = s;
40
41         }
42
43
44         void push(int data){ // same as insert 'at the end'
45
46             if (topOfStack == maxSize-1)
47                 resize(2*maxSize);
48
49             array[++topOfStack] = data;
50
51         }
52
53         /*
54         // Insert at Index is not allowed in Stack
55         // We have to always insert at the end (top) of the Stack
56
57
58         */
59
60         int peek(){
61
62             if (topOfStack >= 0)
63                 return array[topOfStack];
64             else

```

```

65         return -1000000; // an invalid value indicating
66                 // stack is empty
67     }
68 }
69
70
71 int pop(){
72
73     if (topOfStack >= 0){
74         return array[topOfStack--];
75         // the topOfStack is decreased by one
76     }
77     else
78         return -1000000; // an invalid value indicating
79                 // stack is empty
80 }
81
82
83
84
85
86 };
87
88 int main(){
89
90     Stack stack(1);
91
92     stack.push(10);
93     stack.push(23);
94     stack.push(100);
95     stack.push(45);
96     cout << "pop: " << stack.pop() << endl;
97     cout << "peek at top: " << stack.peek() << endl;
98
99     stack.push(85);
100    stack.push(12);
101
102    while (!stack.isEmpty())
103        cout << "pop: " << stack.pop() << endl;
104
105
106    return 0;
107 }

```

```

-
pop: 45
peek at top: 100
pop: 12
pop: 85
pop: 100
pop: 23
pop: 10

```