```cpp
#include "pch.h"
#include <iostream>
using namespace std;

struct Node
{
    int data;
    Node * next;
};

class Stack
{
private:
    Node *top;
    int count;
public:
    Stack()
    {
        top = NULL;
        count = 0;
    }
    bool isEmpty();// check if stack is empty
    void push(int);// add new element in last position of stack
    void pop();// remove last element
    void peek();// Show last element
    void display(); // show all data in stack
    void displayCount(); // show number of elements in stack
};

bool Stack::isEmpty()
{
    return top == NULL;
}
void Stack::push(int val)
{
    if (isEmpty())
    {
        top = new Node;
        top->data = val;
        top->next = NULL;
        count++;
    }
    else
    {
        Node * newNode = new Node;
        newNode->data = val;
        newNode->next = top;
        // copy address of newnode to top
        top = newNode;
        count++;
    }
}

```

```
54  void Stack::pop()
55  {
56      if (top == NULL)
57      {
58          cout << "\nStack underflow.\n";
59      }
60      else
61      {
62          cout << "Element Deleted: " << top->data << endl;
63          Node *temp = top;
64          top = top->next;
65          delete(temp);
66          count--;
67      }
68  }
69
70  void Stack::display()
71  {
72      if (!isEmpty())
73      {
74          Node * temp = top;
75          while (temp != NULL)
76          {
77              cout << temp->data << endl;
78              temp = temp->next;
79          }
80      }
81      else
82      {
83          cout << "\nStack is empty\n";
84      }
85  }
86
87  void Stack::displayCount()
88  {
89      cout << "\nElements in Stack: " << count << endl;
90  }
91
92
93  void Stack::peek()
94  {
95      if (!isEmpty())
96      {
97          cout << "\nFirst Element :" << top->data << endl;
98      }
99  }
100
101
102
103
104
105
106
```

```
107
108 void main()
109 {
110     Stack s;
111     s.push(10);
112     s.push(20);
113     s.push(30);
114     s.push(40);
115
116     s.displayCount();
117     s.display();
118     s.peek();
119
120     s.pop();
121     s.pop();
122     s.pop();
123     s.pop();
124     s.pop();
125
126     getchar();
127 }
128
```