

```
1 from exceptions import Empty
2
3 class LinkedStack:
4     class _Node:
5         __slots__ = '_element', '_next'
6
7         def __init__(self, element, next):
8             self._element = element
9             self._next = next
10
11     def __init__(self):
12         self._head = None
13         self._size = 0
14
15     def __len__(self):
16         return self._size
17     def is_empty(self):
18         return self._size == 0
19
20     def push(self, e):
21         self._head = self._Node(e, self._head)
22         self._size = self._size + 1
23
24     def pop(self):
25         if self.is_empty():
26             raise Empty('Stack is Empty')
27         value = self._head._element
28         self._head = self._head._next
29         self._size = self._size - 1
30         return value
31
32     def top(self):
33         if self.is_empty():
34             raise Empty('Stack is Empty')
35         return self._head._element
36
37     def display(self):
38         temp = self._head
39         while temp :
40             print(temp._element, end='-->')
41             temp = temp._next
42         print()
43
44
45 ls = LinkedStack()
46 ls.push(10)
47 ls.push(20)
```

```
48 ls.push(30)
49 ls.push(40)
50 ls.display()
51 print('Popped: ', ls.pop())
52 ls.display()
53 ls.push(70)
54 ls.display()
55 print('Top Element: ', ls.top())
56 print('Popped: ',ls.pop())
57 ls.display()
58
```