

## Practical 8

---

**Aim:** Write a Program in Python to implement a Stack Data Structure using Class and Objects, with push, pop, and traversal method.

s/w: PyCharm Community edition 2021.3

---

### Theory and approach:

**Stack:** A linear data structure with LIFO characteristics. Insertion and deletion are performed by push and pop.

---

11	← top
21	
31	
41	
51	

---

### Program:

```
# 20CE140 - Dhwani Suthar
# Write a Program in Python to implement a Stack Data Structure using Class
and Objects, with push, pop, and traversal
# method.
```

```
class stack:
    def __init__(self):
        self.item=[]

    #push
    def push(self,data):
        self.item.append(data)

    #pop function
    def pop(self):
        return self.item.pop()

    #traversal
    def traversal(self):
        for i in self.item:
            print(i,end=" ")
```

```
s1=stack()
s1.push(11)
s1.push(21)
s1.push(31)
print("Traversed: ")
s1.traversal()
print('\n')
s1.push(41)
print('Popped: ',s1.pop())
print('Popped: ',s1.pop())
s1.push(51)
print("Printing the element --> ")
s1.traversal()
```

```
# 20CE140 - Dhwani Suthar
# Write a Program in Python to implement a Stack Data Structure using Class and Objects, with push, pop, and traversal
# method.
class stack:
    def __init__(self):
        self.item=[]

    #push
    def push(self,data):
        self.item.append(data)

    #pop_function
    def pop(self):
        return self.item.pop()

    #traversal
    def traversal(self):
        for i in self.item:
            print(i,end=" ")

s1=stack()
s1.push(11)
s1.push(21)
s1.push(31)
print("Traversed: ")
s1.traversal()
print('\n')
s1.push(41)
print('Popped: ',s1.pop())
print('Popped: ',s1.pop())
s1.push(51)
print("Printing the element --> ")
s1.traversal()
```

---

**Output:**

Traversed:

11 21 31

Popped: 41

Popped: 31

Printing the element -->

11 21 51

Process finished with exit code 0

---

**Conclusion:**

1. push(): Elements are pushed at the end.
2. pop(): Elements are popped from the end and then deleted.
3. peek(): Last element is returned without deletion.

---

GitHub: