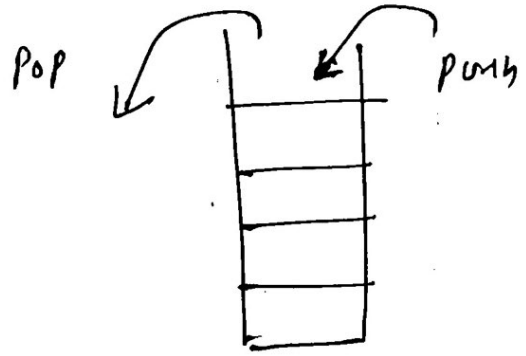
Stack

Stack is a linear type of LIFO (Last in first out) where insertion and deletion is possible from end called top of Stack.



~~Queue~~ operation on Stack

push :- To insert a New element at top of the Stack, Called Push.

pop :- To Remove top element from stack, Called Pop.

peek :- to display top element without deleting Called Peek

Stack Terminology

Stack overflow :- when stack is full and user want to ~~zero~~ insert... New element in stack, called stack overflow.

Stack Underflow :- When stack is empty and user want to remove element in stack

---

Stack Implementation

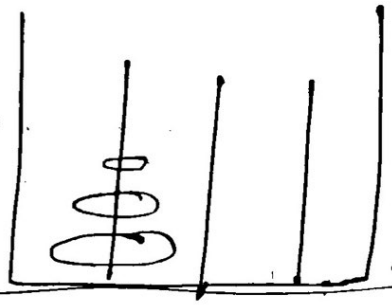
1) Static Implementation
   by using array

2) Dynamic Implementation
   by Using linked list

Stack Example

1) Reverse a string
2) Tower of HANOI
3) Polish Notation

Tower of HANOI



Dynamic Implementation of Linked List

Singly Linked List	Stack
addFirst()	push()
removeFirst()	pop()
getFirst()	peek()
display()	display()
count()	size()
~~head~~	~~top~~

(1) start → null — top → start

(2) start → 10 | ∝

(3) start → 20 | ∝ → 10 | ∝

(4) start → 30 | ∝ → 20 | ∝ → 10 | ∝

```
class Node
{
    int data;
    Node Next;
    Node(int x)
    {
        data = x;
        Next = null;
    }
}

class DStack
{
    Node top;

    void push(int x)
    {
        node Ptv = new node(x);
        if (top == null)
            top = Ptv;
        else
        {
            Ptv.next = top;
            top = Ptv;
        }
    }
}
```

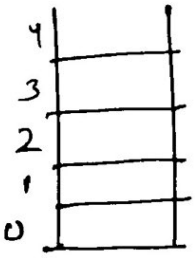
```
void pop()
{
    if (top == null)
        SOPL("stack underflow");
    else
    {
        [illegible] = [illegible]
        {
            [illegible] = top;
            [illegible]
        }
    }
}

void peek()
{
}
```

```
void display()
{

}

int size()
{

}

};
```

Static Implementation of Stack

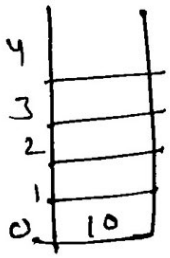
Stack size :- 5

top :- -1



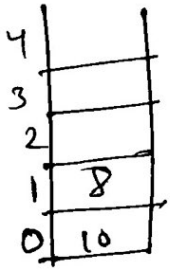
After insert first element (10)

top :- 0



After insert 2nd element (8)

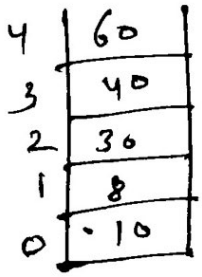
top = 1



:

After insert 5th element

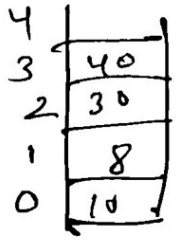
top = 4



Note:- If you want to add another element than stack overflow is occured

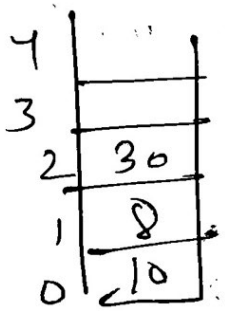
After Remove one element

top = 3



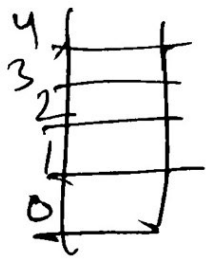
After Remove another element

top = 2



After remove all elements

top = -1



Note:- If you want to remove another elements than Queue Underflow is occured

```
class AStack
{
    int st[];
    int size, top;
    AStack (int N)
    {
     size = N;
     top = -1;
    st = new int.[N];
    }
void    push (int x)
{
 if(top = = -size - 1)
  SOPL ("Stack overflow CaN Not
                    insert");
else
 {
    top++;
   st[top] = x;
 }
}
```

```
void pop()
{
 if (top == -1)
  sopl("Stack Underflow CAN NOT
                    Delete");

 else
 {
  int t = st[top];
   top--;
  sopl("Deleted value =" + t);
 }
}
void disp()
 { if (top == -1)
    sopl("Stack is empty");

   else
   {
     for (int i = top; i >=0; i--)
                sopl(st[i]);
   }
}
```

```
int size()
{
    return top+1;
}

void peek()
{
  if (top==null)
   sopl("Stack is empty");
  else
   sopl(so[top]);
  }
}

class T
{
  p s v m(String K[])
  {
```

```
MStack obj = new MStack(10);
obj.push(10);
obj.push(30);
.
.
.
obj.pop();
.
obj.display();
.
.
}
}
```