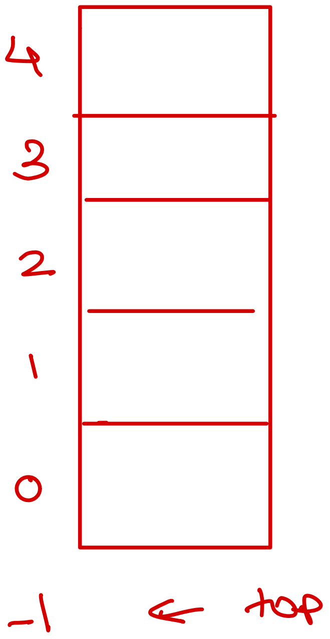
Sunbeam Infotech

Exploring new ideas, Reaching new heights!

Stack using array

arr[5]



stack.h

generic type → T
MAX → constant

```
template <typename T, int MAX>
class Stack {
        T arr[MAX];
        int top;
public:
        stack(){ }
        void push(T val){ }
        void pop(){ } ✓
        T peek(){ }
        bool empty(){ }
        bool full(){ }
};
```

= x init; ✓
        top = -1;

push:
    top++;
    arr[top]=val;

pop:
    top--;

peek:
    return arr[top];

full:
    top == MAX-1

empty:
    top == -1

STL - Standard Template Library

- STL is part of C++ standard.
- It has template implementations of common data structures.
- STL has three main components
  - Containers
  - Algorithms
  - Iterators
- Additionally STL also have
  - Function objects
  - Allocators
  - Utility

classes that implement data structures, e.g. stack<>

- private data members
- member fns.

global fns that operates on containers.

objects used to traverse through containers.

STL

- Containers hold data and operations to be performed on data.
- STL containers are of three types
  - Sequential: Linear collection
    - vector, list, deque
  - Associative: Key-value pair collection
    - set, map, multimap
  - Adapters: Limited container functionality
    - stack, queue

1) vector : dynamic array.
2) list : doubly linked list.
3) deque : double ended queue.

4) set : duplicate values are not allowed.
5) map : fast searching : key-value duplicate key not allowed.
6) multimap : key-value duplicate key allowed.

7) stack : LIFO
8) queue : FIFO

STL

list:

$$\begin{tikzcd} \text{head} \arrow[d] & & & & & \\ 60 \arrow[r] & 50 \arrow[r] & 10 \arrow[r] & 20 \arrow[r] & 30 \arrow[r] & 40 \end{tikzcd}$$

iterators are used to traverse through container.

C / C++ (user defined list)

```
node * trav;
trav = head;
  while (trav != NULL) {
      cout << trav->data;
      trav = trav->next;
  }
```

C++ STL (list) → like int*

```
list<int>::iterator trav;
 trav = l1.begin();
 while( trav != l1.end() ) {
      cout << *trav;
      trav++;
 }
```

STL

- Containers are traversed using iterators.
- Usually iterators are implemented as nested classes in containers.
- Iterators are smart pointers (with -> and * operators overloaded).
- There are six types of iterators
  - Input iterator (read ops, fwd)
  - Output iterator (write ops, fwd)
  - Bi-directional iterator (rw, bi-dirn)
  - Forward iterator (rw, fwd)
  - Reverse iterator (rw, rev)
  - Random access iterator (rw, any)

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>