STL

deque

Saurabh Shukla (MySirG)

Agenda

1. deque
2. Creating deque object
3. Accessing deque elements
4. Implicit and explicit iterators
5. deque methods

deque

- The deque class is a sequential container
- deque is based on double ended queue

- The header required is <deque>
- deque provides random access iterator

How to create a deque object?

```
deque <int> d1;
deque <int> d2 = { 10, 35, 22, 18, 70 };
```

Accessing deque elements

You can access deque in the variety of ways.

1. at()
2. [ ]
3. implicit iterator
4. explicit iterator

Implicit Iterator

```
deque <int> d1 = {10, 20, 30, 40};

for ( int x : d1 )
      cout<<x<<" ";

          or

for (auto x : d1 )
      cout<<x<<" ";
```

Explicit iterator

You can get an iterator object from the following members

1	begin()	end()	iterator
2	cbegin()	cend()	const_iterator
3	rbegin()	rend()	reverse_iterator
4	crbegin()	crend()	const_reverse_iterator

Explicit Iterator

```
deque <int> d1 = { 50, 40, 10, 70, 60 };
deque <int> :: iterator it ;
for (it = d1.begin() ; it != d1.end() ; it++)
        cout << *it << " " ;

deque <int> :: const_iterator it ;
for (it = d1.cbegin() ; it != d1.cend() ; it++)
        cout << *it << " " ;
```

Methods of deque

assign()
empty()
front()
back()

push_front()
emplace_front()
push_back()
emplace_back()
emplace()
insert()

clear()
erase()
pop_front()
pop_back()

swap()
size()