STL

array

Saurabh Shukla (MySirG)

Agenda

1. array
2. creating array object
3. accessing array elements
4. at() and []
5. implicit and explicit iterators
6. array methods

array

One of the basic classes implemented by the standard template library is the array class.

It is a sequential container.

array is based on array data structure

The header for the STL array library is array.

The array class is a part of the std namespace

How to create array object?

```
array <int, 3> a1 = {10, 20, 30};
array <int, 4> a2;
array <int, 3> a3 = a1;
array <int, 5> a4{11, 22, 33};
```

← Contains garbage values

↑ Size of array cannot be changed

↑ Remaining elements are 0.

Accessing array elements

You can access arrays in the variety of ways.

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

[] & at()

```
array <int, 5> a1 = {50, 70, 20, 30, 40};

for (int i=0; i<a1.size(); i++)
        cout<<a1[i]<<" ";

for (int i=0; i<a1.size(); i++)
        cout<<a1.at(i)<<" ";
```

[] no bound checking

at() bound checking

```
array <int, 5> a1 = {50, 70, 20, 30, 40};
```

```
cout << a1[5];
```
prints garbage

```
cout << a1.at(5);
```
throws an exception of out_of_range

Implicit Iterator

```
array <int, 5> a1 = {10, 50, 30, 80, 60};

for ( int x : a1)
    cout<< x <<" ";
```

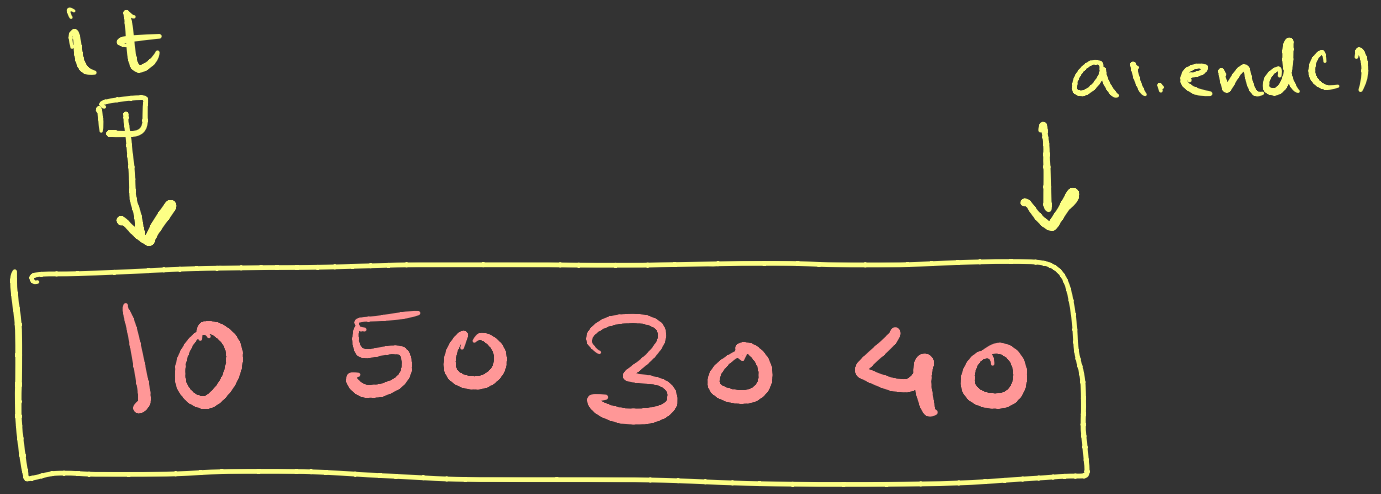
or

```
for (auto x : a1)
    cout << x <<" ";
```

Explicit iterator

You can get an iterator object from the following members of array.

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



it = a1.begin()

a1.end()

*(it + 2) => 30

it + 1

Random

it ++

*(it + i) → random access iterator

it -- → bidirectional iterator

it ++ → forward iterator

---

it ++ and it + 1

forward random access

Explicit Iterator

```
array <int, 5> a1 = { 50, 40, 10, 70, 60 };
 array <int, 5>:: iterator it ;
 for (it = a1.begin() ; it != a1.end() ; it++)
          cout << *it << "  " ;

 array <int, 5>:: const_iterator it ;
 for (it = a1.cbegin() ; it != a1.cend() ; it++)
          cout << *it << "  " ;
```

Explicit Iterator

```
array <int, 5> a1 = { 50, 40, 10, 70, 60 };
array <int, 5>:: reverse_iterator it ;
for (it = a1.rbegin() ; it != a1.rend() ; it++)
        cout << *it << "  " ;

array <int, 5>:: const_reverse_iterator it ;
for (it = a1.crbegin() ; it != a1.crend() ; it++)
        cout << *it << "  " ;
```

Methods of array class

at()
[]

back() returns last element
front() returns first element
empty() returns true or false
data() returns the addres of first element
size() returns the number of elements
swap() swap elements of two arrays.