

Std:vector

⇒ It's a dynamic array, which doesn't enforce any uniqueness to its actual elements.

⇒ It's inside vector header file

#include <vector>

⇒ Declaration

template < class T, class Allocator = std::allocator<T> >

class vector;

* Member functions

Modifiers

(a) Push_back (Tkk value);

(b) Pop_back ();

↳ Remove last element of the vector.

(c) clear();

↳ Clear all element of the container.

Capacity

(a) bool empty() const;

(b) size_type size() const;

Element access

(a) operator [] (size-type pos);

(b) ↳ No bounds checking is performed.

(b) at (size-type pos);

↳ bounds checking is performed.

(d) erase (iterator pos)

Example: If you want to erase 2nd element then

`v.erase(v.begin()+1)`

③ front()

④ back()

⑤ T* data()

↳ Returns pointer to the underlying array
Serving as element storage.

* Optimization of std::vector

① v.reserve(cap)

⇒ By doing this vector will reserve cap size,
so that ~~the~~ vector will not resize until it
reaches cap.

② Use emplace_back instead of push_back

↓
{Construct the object
in place of vector
memory}

↓
{no extra copy
operation.}

↓
{Pass an object to
store}

↓
{1 extra copy
operation}