Vector .. Artful

**Iterators**

1. [begin()](https://www.geeksforgeeks.org/vectorbegin-vectorend-c-stl/) – Returns an iterator pointing to the first element in the vector
2. [end()](https://www.geeksforgeeks.org/vectorbegin-vectorend-c-stl/) – Returns an iterator pointing to the theoretical element that follows the last element in the vector
3. [rbegin()](https://www.geeksforgeeks.org/vector-rbegin-and-rend-function-in-c-stl/) – Returns a reverse iterator pointing to the last element in the vector (reverse beginning). It moves from last to first element
4. [rend()](https://www.geeksforgeeks.org/vector-rbegin-and-rend-function-in-c-stl/) – Returns a reverse iterator pointing to the theoretical element preceding the first element in the vector (considered as reverse end)
5. [cbegin()](https://www.geeksforgeeks.org/vector-cbegin-vector-cend-c-stl/) – Returns a constant iterator pointing to the first element in the vector.
6. [cend()](https://www.geeksforgeeks.org/vector-cbegin-vector-cend-c-stl/) – Returns a constant iterator pointing to the theoretical element that follows the last element in the vector.
7. [crbegin()](https://www.geeksforgeeks.org/vectorcrend-vectorcrbegin-examples/) – Returns a constant reverse iterator pointing to the last element in the vector (reverse beginning). It moves from last to first element
8. [crend()](https://www.geeksforgeeks.org/vectorcrend-vectorcrbegin-examples/) – Returns a constant reverse iterator pointing to the theoretical element preceding the first element in the vector (considered as reverse end)

**Capacity**

1. [size()](https://www.geeksforgeeks.org/vectorempty-vectorsize-c-stl/) – Returns the number of elements in the vector.
2. [max\_size()](https://www.geeksforgeeks.org/vector-max_size-function-in-c-stl/) – Returns the maximum number of elements that the vector can hold.
3. [capacity()](https://www.geeksforgeeks.org/vector-capacity-function-in-c-stl/) – Returns the size of the storage space currently allocated to the vector expressed as number of elements.
4. [resize()](https://www.geeksforgeeks.org/vector-resize-c-stl/) – Resizes the container so that it contains ‘g’ elements.
5. [empty()](https://www.geeksforgeeks.org/vectorempty-vectorsize-c-stl/) – Returns whether the container is empty.
6. [shrink\_to\_fit()](https://www.geeksforgeeks.org/vector-shrink_to_fit-function-in-c-stl/) – Reduces the capacity of the container to fit its size and destroys all elements beyond the capacity.
7. [reserve()](https://www.geeksforgeeks.org/using-stdvectorreserve-whenever-possible/)– Requests that the vector capacity be at least enough to contain n elements.

**Element access:**

1. [reference operator [g]](https://www.geeksforgeeks.org/vectoroperator-vectoroperator-c-stl/) – Returns a reference to the element at position ‘g’ in the vector
2. [at(g)](https://www.geeksforgeeks.org/vectorat-vectorswap-c-stl/) – Returns a reference to the element at position ‘g’ in the vector
3. [front()](https://www.geeksforgeeks.org/vectorfront-vectorback-c-stl/) – Returns a reference to the first element in the vector
4. [back()](https://www.geeksforgeeks.org/vectorfront-vectorback-c-stl/) – Returns a reference to the last element in the vector
5. [data()](https://www.geeksforgeeks.org/vector-data-function-in-c-stl/) – Returns a direct pointer to the memory array used internally by the vector to store its owned elements.

**Modifiers:**

1. [assign()](https://www.geeksforgeeks.org/vector-assign-in-c-stl/)– It assigns new value to the vector elements by replacing old ones
2. [push\_back()](https://www.geeksforgeeks.org/vectorpush_back-vectorpop_back-c-stl/) – It push the elements into a vector from the back
3. [pop\_back()](https://www.geeksforgeeks.org/vectorpush_back-vectorpop_back-c-stl/) – It is used to pop or remove elements from a vector from the back.
4. insert() – It inserts new elements before the element at the specified position
5. [erase()](https://www.geeksforgeeks.org/vectorclear-vectorerase-c-stl/) – It is used to remove elements from a container from the specified position or range.
6. [swap()](https://www.geeksforgeeks.org/vectorat-vectorswap-c-stl/) – It is used to swap the contents of one vector with another vector of same type and size.
7. [clear()](https://www.geeksforgeeks.org/vectorclear-vectorerase-c-stl/) – It is used to remove all the elements of the vector container
8. [emplace()](https://www.geeksforgeeks.org/vector-emplace-function-in-c-stl/) – It extends the container by inserting new element at position
9. [emplace\_back()](https://www.geeksforgeeks.org/vectoremplace_back-c-stl/) – It is used to insert a new element into the vector container, the new element is added to the end of the vector.

vector<int> first,g1; // empty vector of ints

vector<int> second (4,100); // four ints with value 100

vector<int> third (second.begin(),second.end()); // iterating through second

vector<int> fourth (third); // a copy of third

int myints[] = {16,2,77,29}; //from array

vector<int> fifth (myints, myints + sizeof(myints) / sizeof(int) );

cout << "The contents of fifth are:";

for (vector<int>::iterator it = fifth.begin(); it != fifth.end(); ++it)

cout << ' ' << \*it;

gap

g1.push\_back(23);

g1.push\_back(3);

g1.push\_back(2);

g1.push\_back(6);

g1.push\_back(4);

cout << "Output of begin and end: ";

for (auto i = g1.begin(); i != g1.end(); ++i) /// same as cbegin

cout << \*i << " ";

cout << "\nOutput of rbegin and rend: ";

for (auto ir = g1.rbegin(); ir != g1.rend(); ++ir) /// reverse (same as crbegin)

cout << \*ir << " "; ///4 6 2 3 23

cout << "Size : " << g1.size();

cout << "\nCapacity : " << g1.capacity();

cout << "\nMax\_Size : " << g1.max\_size();

g1.resize(4);

cout << "\nSize : " << g1.size();

cout << "\nReference operator [g] : g1[2] = " << g1[2];

cout << "\nat : g1.at(4) = " << g1.at(3);

cout << "\nfront() : g1.front() = " << g1.front();

cout << "\nback() : g1.back() = " << g1.back();

int\* pos = g1.data();

cout << "\nThe first element is " << \*pos;

vector<int> v;

v.assign(5, 10);// fill the array with 10 five times

v.push\_back(15);// inserts 15 to the last position

int n = v.size();

cout << "\nThe last element is : " << v[n - 1];

v.pop\_back();

cout << "\nThe vector elements are: ";

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

v.insert(v.begin(), 5); // inserts 5 at the beginning

cout << "\nThe first element is1: " << v[0];

v.erase(v.begin());cout << "\nThe first element2: " << v[0];// removes the first element

v.emplace(v.begin(), 5);

cout << "\nThe first element is3: " << v[0];

v.emplace\_back(20); n = v.size(); // Inserts 20 at the end

cout << "\nThe last element is: " << v[n - 1];

vector<int> v1, v2;

v1.push\_back(1);

v1.push\_back(2);

v2.push\_back(3);

v2.push\_back(4);

cout << "\n\nVector 1: ";

for (int i = 0; i < v1.size(); i++)cout << v1[i] << " ";

cout << "\nVector 2: ";

for (int i = 0; i < v2.size(); i++) cout << v2[i] << " ";

v1.swap(v2); // Swaps v1 and v2

///After Swap

///Vector 1: 3 4

///Vector 2: 1 2

Output

The contents of fifth are: 16 2 77 29

Output of begin and end: 23 3 2 6 4

Output of rbegin and rend: 4 6 2 3 23 Size : 5

Capacity : 8

Max\_Size : 1073741823

Size : 4

Reference operator [g] : g1[2] = 2

at : g1.at(4) = 6

front() : g1.front() = 23

back() : g1.back() = 6

The first element is 23

The last element is : 15

The vector elements are: 10 10 10 10 10

The first element is1: 5

The first element is2: 10

The first element is3: 5

The last element is: 20

Vector 1: 1 2

Vector 2: 3 4