C++ STL MASTER CHEAT SHEET

Complete One-Page Reference for All Major STL Containers and Algorithms

VECTOR

Header:
Declaration: vector v; vector v(n); vector v(n, value); vector v = {1,2,3};

Common Operations:
- push_back(x), pop_back(), insert(it, x), erase(it), clear(), size(), empty(), resize(n)
- front(), back(), at(i), operator[](i)
- begin(), end(), rbegin(), rend()
- swap(v2), assign(n, value)

Loops:
for(int i=0;i
Algorithms:
sort(v.begin(), v.end()), reverse(v.begin(), v.end()), max_element(), min_element(), accumulate(), count(), find(), erase-remove idiom

Example:
vector v={1,2,3}; v.push_back(4); cout<

LIST

Header:
Declaration: list l; list l(n, val);

Operations:
push_back(), push_front(), pop_back(), pop_front(), insert(), erase(), clear(), size(), empty()
front(), back(), reverse(), sort(), remove(value), unique()

Iterators:
begin(), end(), rbegin(), rend()

Example:
list l={1,2,3}; l.push_front(0); for(int x:l) cout<

DEQUE

Header:
Declaration: deque d;

Operations:
push_back(), push_front(), pop_back(), pop_front(), insert(), erase(), clear(), size(), empty()
front(), back(), at(), []

Example:
deque d={1,2,3}; d.push_front(0);

STACK

Header:
Declaration: stack s;

Operations:
push(x), pop(), top(), size(), empty()

Example:
stack s; s.push(10); cout<

QUEUE

Header:
Declaration: queue q;

Operations:
push(x), pop(), front(), back(), size(), empty()

Example:
queue q; q.push(1); q.push(2); cout<

PRIORITY_QUEUE

Header:
Declaration: priority_queue pq; // max-heap
priority_queue, greater> pq; // min-heap

Operations:
push(x), pop(), top(), size(), empty()

Example:
priority_queue pq; pq.push(5); pq.push(1); cout<

PAIR

Header:
Declaration: pair p(1, "abc");

Access:
p.first, p.second

Example:
pair p1={1,2}; cout<

SET

Header:
Declaration: set s;

Operations:
insert(x), erase(x), count(x), find(x), size(), empty(), clear()

Iterators:
begin(), end()

Example:
set s={1,2,3}; s.insert(4); for(int x:s) cout<

UNORDERED_SET

Header:
Declaration: unordered_set s;

Operations:
insert(), erase(), find(), count(), clear(), size()

Example:
unordered_set us; us.insert(10); if(us.count(10)) cout<<"Found";

MAP

Header:
Declaration: map m;

Operations:
insert({key,val}), erase(key), find(key), count(key), clear(), size(), empty()
Access: m[key], at(key)

Iterators:
begin(), end()

Example:
map m; m[1]="A"; for(auto &p;:m) cout<

UNORDERED_MAP

Header:
Declaration: unordered_map m;

Operations:
insert({k,v}), erase(k), find(k), count(k), clear(), size(), empty()

Example:
unordered_map um; um[1]=10; cout<

ALGORITHMS

Header: ,

Common Algorithms:
- sort(v.begin(), v.end())
- reverse(v.begin(), v.end())
- max_element(v.begin(), v.end())
- min_element(v.begin(), v.end())
- accumulate(v.begin(), v.end(), 0)
- count(v.begin(), v.end(), x)
- find(v.begin(), v.end(), x)
- erase-remove idiom → v.erase(remove(v.begin(), v.end(), x), v.end())
- next_permutation(), prev_permutation()
- binary_search(v.begin(), v.end(), x)
- all_of(), any_of(), none_of()

Example:
```
int sum = accumulate(v.begin(), v.end(), 0);
int mx = *max_element(v.begin(), v.end());
if(binary_search(v.begin(), v.end(), 5)) cout<<"Found";
```