Advanced C++ – STL basics

container
eg: vector<int> c;
list<int> c;
map<string int> c;
	array<T,N>	vector<T>	list<T>	deque<T>	set<T> (tree)
					set<T, *comparator*>
					unordered_set<T>
					multiset<T>
						map<K,T> (tree)
						
						unordered_map<K,T>
						multimap<K,T>

Implementation	array	array	double list	array***	RB tree (ordered) Hashtable (unordered)	RB tree (ordered) Hashtable (unordered)
Override operator:	bool operator==(const T& rhs) const;				const bool operator < (const T& rhs) const	
Iterators **c : constant, r : reverse**						
iter = con.[c][r]begin() **iter = con.end()**	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓	✓ ✓
Capacity						
int n = con.size()	✓	✓	✓	✓	✓	✓
int n = con.max_size()	✓	✓	✓	✓	✓	✓
con.resize(10);	-	✓	✓	✓	-	-
bool b = con.empty()	✓	✓	✓	✓	✓	✓
con.shrink_to_fit()	-	✓	-	✓	-	-
Element access						
T elem = con[n] **T elem = con.at(n)**	✓ ✓	✓ ✓	- -	✓ ✓	- -	✓ (con[n] =… inserts pair) ✓
T elem = con.front() **T elem = con.back()**	✓ ✓	✓ ✓	✓ ✓	✓ ✓	- -	- -
T* p_elems = con.data()	✓	✓	-	-	-	-
Modifiers						
con.fill(value)	✓	-	-	-	-	-

con.push_front(value)
con.pop_front()
	-
	-
		-
		-
			✓
			✓
				✓
				✓
					-
					-
						-
						-

con.push_back(value)
con.pop_back()
	-
	-
		✓
		✓
			✓
			✓
				✓
				✓
					-
					-
						-
						-

con.insert(con.begin() + n, value)	-	✓	✓	✓	✓	✓
con.erase(con.begin() + n)	-	✓	✓	✓	✓	✓
con.clear()	-	✓	✓	✓	✓	✓
Operations						
con.remove(val) // removes all	-	-	✓	-	-	-
con.sort()	-	-	✓	-	-	-
con.reverse()	-	-	✓	-	-	-
con.merge(con2)	-	-	✓	-	-	-
iter = con.find(key) hint: con.erase(con.find(key))	-	-	-	-	✓	✓
con.count(key)	-	-	-	-	✓	✓

con.lower_bound(value)
con.upper_bound(value)
	-
	-
		-
		-
			-
			-
				-
				-
					✓
					✓
						✓
						✓

Notes:

In all containers, iterator definition includes automatically overloading of operator * so that it returns the element of the iterator.

In map, iterator points to pair(K, T). The pair provides access to key and value through first and second member functions:

```
map<string,int> data;
data.insert(pair<string,int>("Papaioannou",8));
data["Tsatiris"]=9;

map<string,int>::iterator result = data.find("Tsatiris");
if (result!=data.end())
        cout << result->first << " " << result->second << endl;
}
```