CPP08

STL → Standard Template Library

* Container & Algorithm :

ex :
```
std::list<int>    lst1;
std::vector<int>  v1;

lst1.push_back(1);
std::list<int>::const_iterator it;
std::list<int>::const_iterator ite = lst1.end();
for (it = lst1.begin(); it != ite; ++it) {
    std::cout << *it << std::endl;
```

Container = a holder object that stores a collection of other objects (its elements)

↳ implemented as class templates → ⊕ flexibility in supported elements.

↳ provides storage space for elems & provide mbr functions to access them, directly or through iterators (reference objects w/ similar properties to pointers)

- sequence containers: array, vector, deque (double ended queue, forward_list, list)
- container adaptors: stack (LIFO), queue (FIFO), priority_queue
- associative containers: set, multiset, map, multimap (multiple key map)