**Containers**

Containers are used to manage collections of objects of a certain kind. The containers may be implemented as arrays or as linked lists, or they may have a special key for every element. Every kind of container has its own advantages and disadvantages, so having different container types reflects different requirements for collections in programs.

## Sequence containers

Sequence containers are ordered collections in which every element has a certain position. This position depends on the time and place of the insertion, but it is independent of the value of the element.

Sequence containers are usually implemented as arrays or linked lists.

1. array (C++11)
2. vector
3. dequeue
4. list
5. forward\_list (C++11)

## Associative containers

Associative containers are sorted collections in which the position of an element depends on its value (or key, if it’s a key/value pair) due to a certain sorting criterion. If you put six elements into a collection, their value determines their order. The order of insertion doesn’t matter.

Associative containers are usually implemented as binary trees.

1. set
2. mutiset
3. map
4. multumap

## Unordered associative containers

Unordered (associative) containers are unordered collections in which the position of an element doesn’t matter. The only important question is whether a specific element is in such a collection. Neither the order of insertion nor the value of the inserted element has an influence on the position of the element, and the position might change over the lifetime of the container. Thus, if you put six elements into a collection, their order is undefined and might change over time.

Unordered containers are usually implemented as hash tables.

The major advantage of unordered containers is that finding an element with a specific value is even faster than for associative containers.

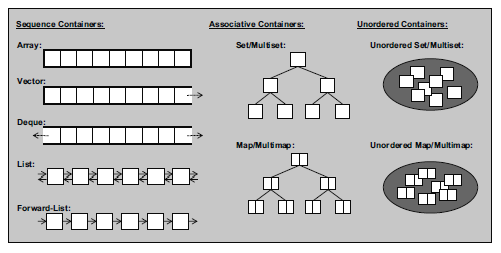
In fact, the use of unordered containers provides amortized constant complexity, provided that you have a good hash function. However, providing a good hash function is not easy, and you might need a lot of memory for the buckets.

1. unordered\_set
2. unordered\_multiset
3. unordered\_map
4. unordered\_multumap

## Container adaptors

Container adapters, which are predefined containers that provide a restricted interface to meet special needs. These container adapters are implemented by using the fundamental container classes.

1. stack
2. queue
3. priority\_queue



STL Container Types

# END