**The C++ Standard Template Library (STL)**

The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators. It is a generalized library and so, its components are parameterized. A working knowledge of [template classes](https://www.geeksforgeeks.org/templates-cpp/) is a prerequisite for working with STL.

**STL has four components**

* Algorithms
* Containers
* Functions
* Iterators

**Algorithms**

The header algorithm defines a collection of functions especially designed to be used on ranges of elements.They act on containers and provide means for various operations for the contents of the containers.

* Algorithm
  + [Sorting](http://quiz.geeksforgeeks.org/sort-algorithms-the-c-standard-template-library-stl/)
  + [Searching](http://quiz.geeksforgeeks.org/binary-search-algorithms-the-c-standard-template-library-stl/)
  + [Important STL Algorithms](https://www.geeksforgeeks.org/c-magicians-stl-algorithms/)
  + [Useful Array algorithms](https://www.geeksforgeeks.org/useful-array-algorithms-in-c-stl/)
  + [Partition Operations](https://www.geeksforgeeks.org/stdpartition-in-c-stl/)
* Numeric
  + [valarray class](https://www.geeksforgeeks.org/std-valarray-class-c/)

**Containers in C++ STL (Standard Template Library)**

A container is a holder object that stores a collection of other objects (its elements). They are implemented as class templates, which allows a great flexibility in the types supported as elements.

The container manages the storage space for its elements and provides member functions to access them, either directly or through iterators (reference objects with similar properties to pointers).

**Sequence containers**

Sequence containers implement data structures which can be accessed sequentially.

* [**array:**](https://www.geeksforgeeks.org/array-class-c/) Static contiguous array (class template)
* [**vector:**](https://www.geeksforgeeks.org/vector-in-cpp-stl/) Dynamic contiguous array (class template)
* [**deque:**](https://www.geeksforgeeks.org/deque-cpp-stl/) Double-ended queue (class template)
* [**forward\_list:**](https://www.geeksforgeeks.org/forward-list-c-set-1-introduction-important-functions/) Singly-linked list (class template)
* [**list :**](https://www.geeksforgeeks.org/list-cpp-stl/) Doubly-linked list (class template)

Associative containers implement sorted data structures that can be quickly searched (O(log n) complexity).

* [**Set:**](https://www.geeksforgeeks.org/set-in-cpp-stl/) Collection of unique keys, sorted by keys  
  (class template)
* [**Map:**](https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/) Collection of key-value pairs, sorted by keys, keys are unique (class template).
* [**multiset:**](https://www.geeksforgeeks.org/multiset-in-cpp-stl/) Collection of keys, sorted by keys (class template)
* [**multimap:**](https://www.geeksforgeeks.org/multimap-associative-containers-the-c-standard-template-library-stl/) Collection of key-value pairs, sorted by keys  
  (class template)

**Unordered associative containers**

Unordered associative containers implement unsorted (hashed) data structures that can be quickly searched (O(1) amortized, O(n) worst-case complexity).

* [**unordered\_set:**](https://www.geeksforgeeks.org/unorderd_set-stl-uses/) Collection of unique keys, hashed by keys. (class template)
* [**unordered\_map:**](https://www.geeksforgeeks.org/unordered_map-in-stl-and-its-applications/) Collection of key-value pairs, hashed by keys, keys are unique. (class template)
* [**unordered\_multiset:**](https://www.geeksforgeeks.org/unordered_multiset-and-its-uses/) Collection of keys, hashed by keys (class template)
* [**unordered\_multimap:**](https://www.geeksforgeeks.org/unordered_multimap-and-its-application/) Collection of key-value pairs, hashed by keys (class template)

**Container adaptors**

Container adaptors provide a different interface for sequential containers.

* [**stack:**](https://www.geeksforgeeks.org/stack-in-cpp-stl/) Adapts a container to provide stack (LIFO data structure) (class template).
* [**queue:**](https://www.geeksforgeeks.org/queue-cpp-stl/) Adapts a container to provide queue (FIFO data structure) (class template).
* [**priority\_queue:**](https://www.geeksforgeeks.org/priority-queue-in-cpp-stl/) Adapts a container to provide priority queue (class template).