The Standard Template Library (STL)

- The STL is:
  - A generic library that provides solutions to managing collections of data with modern and efficient algorithms.
  - Allows programmers to benefit from innovations in data structures and algorithms without needing to learn how they work.
  - From the programmer's point of view, the STL provides a bunch of collection classes that meet different needs, together with several algorithms that operate on them.

STL

- STL's flexibility, has a price, chief of which is that it is not self-explanatory.
- STL is based on the cooperation of different well-structured components, key of which are containers, iterators, and algorithms.

STL Components

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

STL

- All components of the STL are templates, so they can be used for arbitrary element types.
- Provides a framework for supplying other collection classes or algorithms for which existing collection classes and algorithms work.
- STL gives C++ a new level of abstraction.
- Forget programming dynamic arrays, linked lists, and binary trees.
- Forget programming different search algorithms.
- To use the appropriate kind of collection, you simply define the appropriate container and call the member functions and algorithms to process the data.

Supplemental Text

- The C++ Standard Library: A tutorial and Reference, by Nicolai M. Josuttis, Addison-Wesley, 1999
- The Standard Template Library (STL) influenced the overall architecture of the C++ standard library
STL Components

- **Iterators** are used to step through the elements of collections of objects.
  - These collections may be containers or subsets of containers.
  - The major advantage of iterators is that they offer a small but common interface for any arbitrary container type.
  - For example, one operation of this interface lets the iterator step to the next element in the collection.
  - This is done independently of the internal structure of the collection, i.e., it works regardless of whether the collection is an array or a tree.
  - Every container class provides its own iterator type that "does the right thing" because it knows the internal structure of its container.

- **Algorithms** are used to process the elements of collections.
  - For example, they can search, sort, modify, or simply use the elements for different purposes.
  - Algorithms use iterators.
  - Thus, an algorithm has to be written only once to work with arbitrary containers because the iterator interface for iterators is common for all container types.
  - To give algorithms more flexibility you can supply certain auxiliary functions called by the algorithms.
  - You can use a general algorithm to suit your needs even if that need is very special or complex.
  - For example, you can provide your own search criterion or a special operation to combine elements.

The Concept of the STL

- The concept of the STL is based on a separation of data and operations.
- The data is managed by container classes, and the operations are defined by configurable algorithms.
- Iterators are the glue between these two components.
- Iterators let any algorithm interact with any container.

Containers

- Container classes, or containers for short, manage a collection of elements.
Containers

There are two general kinds of containers:

1. **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.
   - For example, if you put six elements into a collection by appending each element at the end of the actual collection, these elements are in the exact order in which you put them.
   - The STL contains three predefined sequence container classes: vector, deque, and list.

2. **Associative containers** are sorted collections in which the actual position of an element depends on its value due to a certain sorting criterion.
   - If you put six elements into a collection, their order depends only on their value.
   - The order of insertion doesn't matter.
   - The STL contains four predefined associative container classes: set, multiset, map, and multimap.

An associative container can be considered a special kind of sequence container because sorted collections are ordered according to a sorting criterion.