# STANDARD TEMPLATE LIBRARY

# Standard Template Library

|  |  |
|---|---|
| Concept | a library of C++ template classes for containers, iterators and algorithms |
| Containers | classes which manage collections of objects (OOP data structures) |
| Iterators | classes for iterating through containers |
| Algorithms | common algorithms for searching, sorting and modifying containers |

# Standard Containers

| | |
|---|---|
| Reference | http://www.cplusplus.com/reference/stl/ |
| Containers | these are some of the container classes included in the STL |

| | |
|---|---|
| array | static array |
| deque | double ended queue |
| forward_list | singly linked list |
| map | associative key, value pair |
| list | doubly linked list |
| queue | queue |
| stack | stack |
| vector | dynamic array |

# Vector

Concept a vector is an STL container class for a dynamic array

Example

```
#include <vector>                    // enable the use of vectors in this program

int main() {
    std::vector<int> v;              // create a vector of integers
    std::cout << v.size();           // print the number of elements in the vector v
    std::cout << v.capacity();       // print the current capacity of the vector v

    v.push_back(5);                  // add 5 to the end of the vector
    v.push_back(10);                 // add 10 to the end of the vector

    for(int i=0; i<v.size(); ++i) {
        cout << v[i] << " ";         // print each element in the vector
    }
}
```

## Ranged For Loop

Concept a loop for iterating through container classes

iterates through all values currently in the container

Example

```
#include <vector>                    // enable the use of vectors in this program

int main() {
      std::vector<int> v;            // create a vector of integers
      v.push_back(5);                // add 5 to the end of the vector
      v.push_back(10);               // add 10 to the end of the vector
      v.push_back(15);               // add 15 to the end of the vector

      for(int e: v) {                // ranged for loop to iterate through vector v
            cout << e << " ";        // print each element e in the vector
      }
}
```

## Containers and Functions

Example

```
void init(std::vector<int> &v) {             // function to initialize a vector of integers
      for(int i=0; i<10; ++i) {              // store 10 integers into the vector
            v.push_back(i);                  // push_back grows the vector as needed
      }
}
template<typename T>                         // template function
void output(std::vector<T> v) {              // print a vector of type T
      for(T e: v) {                          // ranged for loop
            std::cout << e << " ";           // print each element in the vector
      }
}

int main() {
      std::vector<int> v;                    // create a vector of integers
      init(v);                               // send the vector v to an initialize function
      output(v);                             // send the vector v to a print function
}
```

# Containers and Iterators

Example

```
int main() {
	std::vector<int> v;				// create a vector of integers
	v.push_back(5);					// append 5 to the vector
	v.push_back(10);				// append 10 to the vector
	v.push_back(15);				// append 15 to the vector

	std::vector<int>::const_iterator iter;	// create an int vector iterator object
	iter = v.begin();				// associate this object with vector v

	while( iter != v.end() ) {			// iterate until the end of the vector
		std::cout << *iter << " ";		// print the current vector value
		++iter;					// advance the iterator to the next value
	}
}
```

# Containers and Algorithms

Reference http://www.cplusplus.com/reference/algorithm/

Example

```
#include <vector>
#include <algorithm>

int main() {
      std::vector<int> v;                          // create a vector of integers
      v.push_back(321);                            // append 5 to the vector
      v.push_back(4);                              // append 10 to the vector
      v.push_back(64);                             // append 15 to the vector

      std::sort( v.begin(), v.end() );             // stl sort of a vector

      for(int e: v) {
            std::cout << e;                        // print a sorted vector
      }
}
```