STL Associative containers

* Provide direct access to store & retrieve data (such as structs, classes, or any other type).
* Each data item has an associated value known as the ***key***
* Four Associative containers are: *set, multiset, map* and *multimap.*
* Example: if the data is an Employee’s record, the key might be the employee’s social security number.
* Items are retrieved using the key.
* The key type and the type of data to be stored need not be related to one another but they often are.
* A simple case is when each data item is its own key. A **set** is an example where every element is its own key.
* A set is a template class which stores elements without repetition.
* Each associative container stores the keys is *sorted order*.
* Both set and multiset provide operations to manipulate sets of values *where the values are the keys*.
* The difference between a set and a multiset is that a set does not allow duplicate values whereas as multiset does.
* Classes *map* and *multimap* provide operations for manipulating data values using keys.
* The difference between a map and multimap is that a map allows only a unique key with a data value whereas a multimap allows duplicate keys associated with data values to be stored.
* In order to work efficiently, a set object stores data in sorted order.
* The sorting order for elements can be specified as in

Set <T, Ordering> S;

Some member functions: (s is a set object)

s.insert(elem); //insert a copy of element in the set. If

// element is already in the set, this has

// no effect.

s.erase(elem); // Removes elem from the set. If the elem js not in // the set, this has no effect.

s.find(elem); // returns a mutable iterator located at the

// copy of the elem in the set. If elem is

// not in the set, s.end() is returned.

s.erase(iterator) // Erases the elem at the location of the

// iterator

s.size() // Returns the number of elements in the set

s.empty() // Returns true if the set is empty, else

// false

s1 == s2 // Returns true if the sets contain the same

// elements, otherwise returns false.