STL Lists

CPE 212 -- Lecture 19 continued

** Notes based on
The C++ Standard Library: A Tutorial and Reference, by Niicolai M. Josuttis

Lists

- Doubly-linked list of nodes
- No random access (i.e. no at() or [ ])
  - Must traverse nodes to reach desired position
- Insertions/deletions are constant time regardless of position since relocation of existing elements not required
- Must use iterators
- #include <list>

Selected List Operations - 1

- list<T> someList;
  - Creates list with no elements
- list<T> someList(int someSize);
  - Creates list with someSize elements, each created using the default constructor for type T
- list<T> someList(int someSize, T value);
  - Creates list with someSize elements of type T, all initialized to value
- ~list<T>()
  - Destructor

Selected List Operations - 2

- size()
  - Number of elements currently stored
- empty()
  - Returns true if empty, false otherwise
- front()
  - Returns first element but does not check to see if it exists
- back()
  - Returns last element but does not check to see if it exists

Selected List Operations - 3

- push_back(T someValue)
  - Adds someValue to end of list
- push_front(T someValue)
  - Adds someValue to front of list
- pop_back()
  - Removes last element from end of list but does not return it
- pop_front()
  - Removes last element from front of list but does not return it
- insert(list<T>::iterator k, T someValue)
  - Inserts at iterator position k the value someValue

- There are many more methods available for inserting, erasing, removal of duplicate values, sorting, etc.