Agenda / Learning Objectives:

1. Learn about vector. Run the following commands to extract lab26.tar in your venus account (note the dot):

   cp ~ctse/cs211/lab26.tar . ; tar xvf lab26.tar

2. Recursion review: Go through chap13cpp6th_abridged.pdf and Ch13TestQuestions.pdf to get ready for quiz #4 next week.

3. Read the key points and pitfall (for chapter 7.3) on page 2.

From teacher's note for Absolute C++:

Key Points

Vectors. The STL vector container is a generalization of array. A vector is a container that is able to grow (and shrink) during program execution. A container is an object that can contain other objects. The STL vector container is implemented using a template class (Chapter 16). The text's approach is to define some vector objects and use them prior to dealing with templates and the STL in detail (Chapter 19). Unlike arrays, vector objects can be assigned, and the behavior is what you want. Similar to an array, you can declare a vector to have a 10 elements initialized with the default constructor by writing

```
vector<baseType> v(10);
```

Like arrays, objects stored in a vector object can be accessed for use as an l-value or as an r-value using indexing.

```
v[2] = 3;
x = v[0];
```

Unlike arrays, however, you cannot (legally) index into a vector, say v[i], to fetch a value or to assign a value unless an element has already been inserted at every index position up to and including index position i. The `push_back(elem)` function can be used to add an element to the end of a vector.

Efficiency Issues for Vectors. Most implementations of vector have an array that holds its elements. The array is divided into that used for elements that have been inserted, and the rest is called reserve. There is a member function that can be used to adjust the reserve up or down. Implementations vary with regard to whether the reserve member can decrease the capacity of a vector below the current capacity.

Pitfalls

There is no array bounds checking done by a vector. There is a member function, `at(index)` that does do bounds checking. When you access or write an element outside the index range 0 to (size-1), is undefined. Otherwise if you try to access v[i] where I is greater than the vector's size, you may or may not get an error message but the program will undoubtedly misbehave.