Building a container class.

1. Questions?

2. From before: The Association class.

3. Building an extensible list, a Vector.

(a) Based on arrays of Objects. We allocate arrays with the following type of notation:

   Widget a[] = new Widget[n];

   Where n is the number of cells required. Note that this allocates an array of n object references; there is no need to think about how they are construct. Notice that an array can be allocated with a specific length. We can determine the length of the array with the length instance variable:

   if (a.length == n) doSomethingWith(a);

   The length cannot be changed. That's our motivation for a Vector.

(b) What would we need to keep track of, privately, to maintain state?

(c) What would be an appropriate constructor? For each constructor, we should be able to determine reasonable values of the state variables; the constructor should leave the Vector in a valid state.

(d) How do we implement int size()? Compare this with int capacity().

(e) How do we implement boolean isEmpty()? Is there a notion of full?

(f) How can we check to see if an Vector has the ability to hold n values? Thinking about how this is implemented is the most important part of the engineering of this structure.

(g) How do we implement void addElement(Object v)? Where should it be added? What is the cost of implementing this method?

(h) How do we implement Object elementAt(int n)?

(i) How do we implement boolean contains(Object v)? What happens when we have an array of Associations?

(j) Is it useful to add elements at a specific location? How might this be accomplished? What is the cost?

⋆