



# Lists and Linked Lists

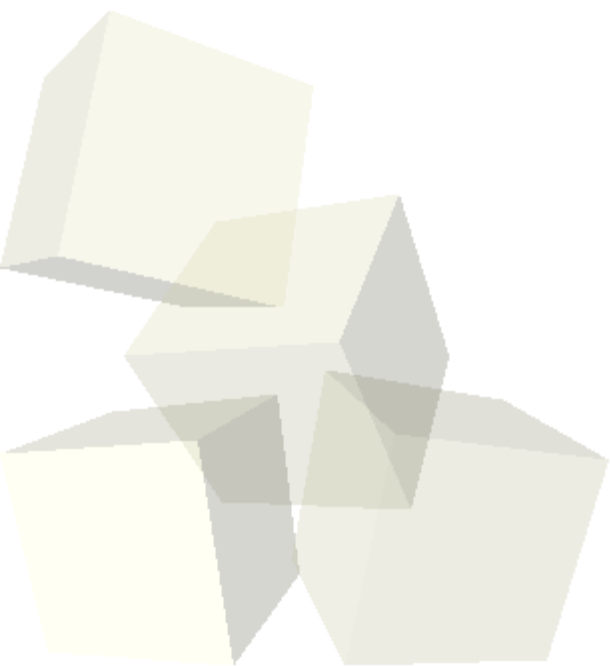
11/26/2007





# Opening Discussion

- Let's look at solutions to the interclass problem.
- Do you have any questions about the assignment?





# Abstract Data Types (ADTs)

- One of the most powerful concept in computer science is the abstract data type.
- This is something that hold information for us and defines a certain set of operations we can do on that data.
- The ADT does not tell you how it will do those operations. This is critical as most operations can be done in different ways.
- Some common ADTs include the following.
  - ◆ List
  - ◆ Map
  - ◆ Set



- Today we will concern ourselves with the list ADT. Conceptually this is quite simple. Think of the lists you write yourself to help organize.
- You have certain operations.
  - ◆ Get element from list
  - ◆ Add element to list
  - ◆ Insert into the list
  - ◆ Remove from the list
  - ◆ Get the size of the list
- There might be others, but those are the basics.
- Once again, the list ADT doesn't tell you how to store the elements or do these things. It just says that a list must do them.



# Array Based List

- You could write all of these functions using an array as the storage mechanism for your list.
- Let's think about the different functions we have to implement.
  - ◆ Getting an element is just looking up by index
  - ◆ Adding an element at the end is storing it and then keeping track of the fact we have one more.
  - ◆ Inserting into the list requires moving other elements down to make a hole in the list.
  - ◆ Removing from the list requires moving elements down to fill up the hole.
  - ◆ Return a stored value for the size.
- Some of these are fast, but others aren't. The slow ones require a lot of memory movement.



- The solution to this is to use a different implementation of the list: a linked list.
- A linked list is made out of nodes that store data. Each node knows about one or two of its neighbors.
- If it knows about one it is called a singly linked list. If it knows about two it is a doubly linked list.
- To build a linked list we need a structure for the nodes and most of the time we'll have a structure for the list to hide the nodes in.
- The linked list is efficient for adding and removing, but random access is slow.



- What do you have to do to return a random element from a linked list?
- Quiz #6 is at the beginning of next class.
- Interclass Problem – Write an implementation for an array based list with the five operations mentioned in this lecture.

