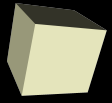




9/24/2007





Opening Discussion

- Let's look at solutions to the interclass problem.





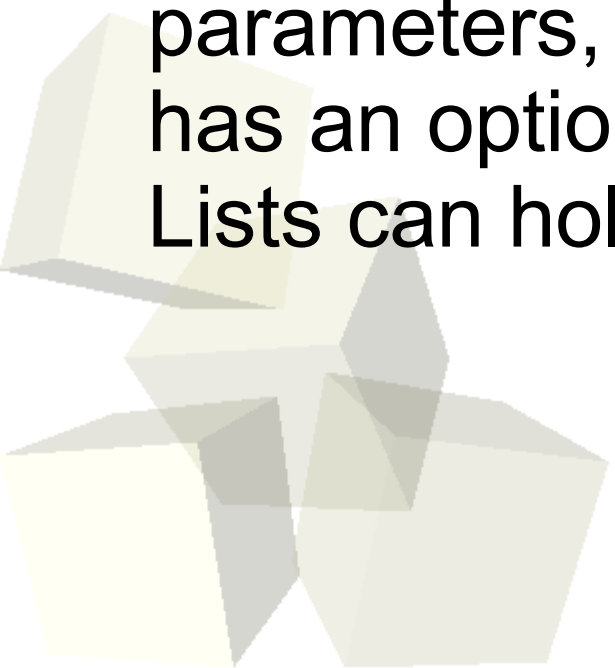
Working with Multiple Objects

- While we have given ourselves a lot of power in Alice, there are still some things that are difficult. Working with a large number of different object is an example.
- To do this we need some construct in the language that represents a group of things so that we can manipulate the group or give commands to elements in the group.
- In Alice we accomplish this with lists and arrays.





- A list is a type of abstract data type with certain operations defined on it that mimic what we generally think of as things we can do to a list.
- You use a list in Alice when you have a collection of objects that you want to add to and remove from.
- The dialog box that comes up for variables, parameters, properties, and function return types has an option that allows you to make it into a list. Lists can hold any of the Alice types in them.





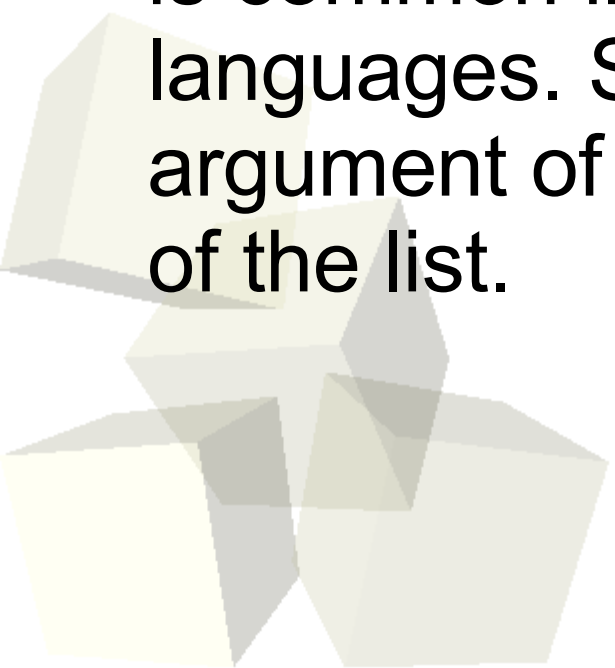
List Operations

- Alice provides the operations for `AllInOrder` and `forAllTogether` to allow you to do something to all of the elements of a list.
- The loop will create a variable that stores each item from the list and executes the code in the loop body.
- At this point we need to make a list and demonstrate using each of these constructs on the items of the list.





- Lists have methods in them that we can call to modify the contents of the list.
 - ◆ Three versions of add put new items in the list as different locations.
 - ◆ Versions of remove take something out of the list.
 - ◆ The clear method takes everything out of the list.
- Note that the index of a list starts at 0, not 1. This is common in many modern programming languages. So the add method with a first argument of 0 places a new item at the beginning of the list.



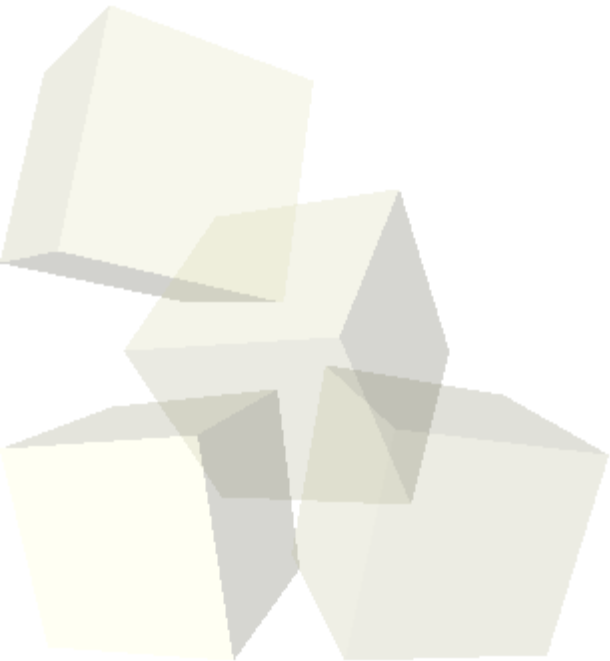


- Lists also have functions that we can call to get information from the list.
 - ◆ size tells us how many elements are on the list.
 - ◆ firstIndexOf searches for something in the list and gives the index it first appears at.
 - ◆ lastIndexOf is similar to firstIndexOf, but starts searching at the end.
 - ◆ [] get an item from the list.
 - ◆ getLastItem and getRandomItems return the respective items.





- Let's actually do something with a list in our program.





- How might you use a list in your project?
- Interclass Problem - Place four objects in a world. Three should be characters you can animate and one should be a stationary object. Put the three characters on a list and have them walk to the stationary object. Each should stop when it gets close to the object.

