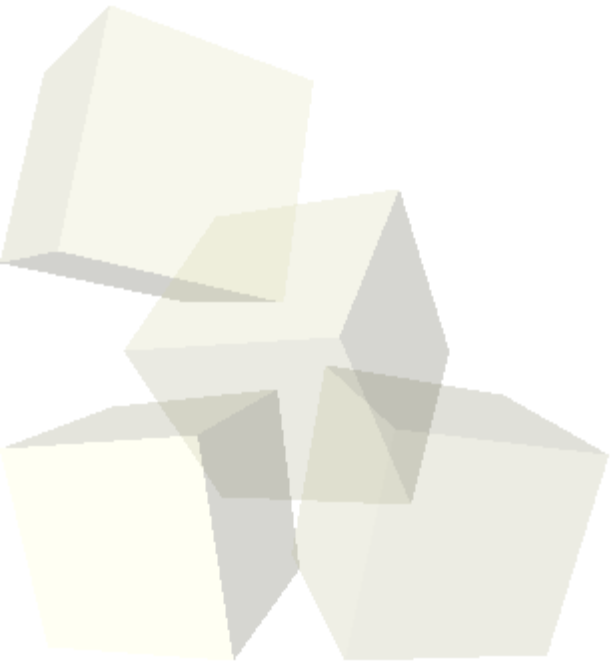
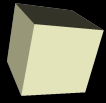


# Using Functions and Conditionals

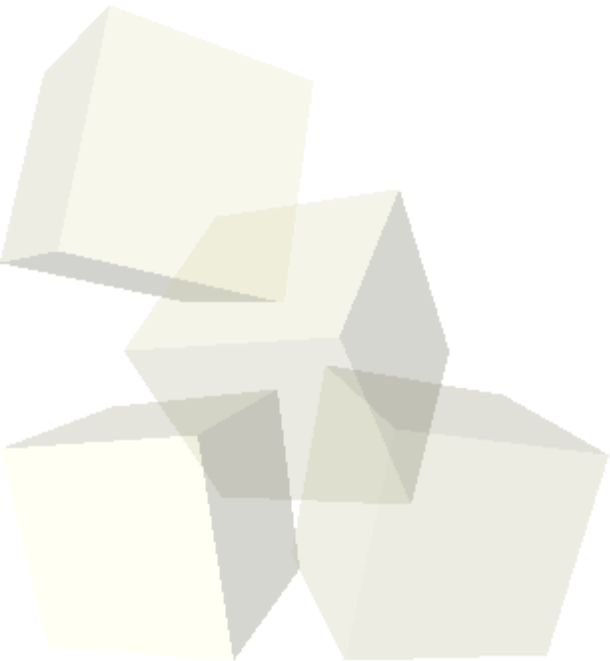
2/4/2008

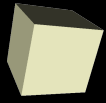




# Opening Discussion

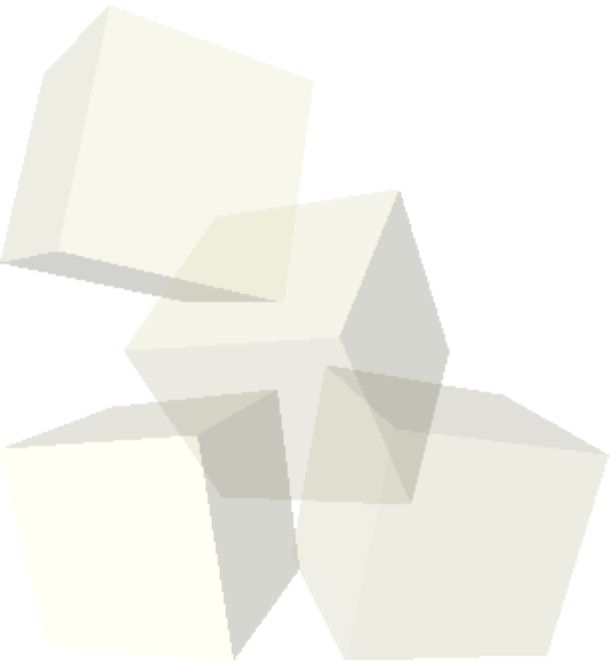
- Let's look at solutions to the interclass problem.
- Building exclusive or from and, not, and inclusive or.

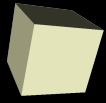




# Letter Grades

- Today let's create a classroom scene. A kid in the class will ask us what grade he got and the user will input a number grade. The kid should then respond with the proper letter grade.
- How are we going to do this?

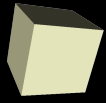




# Nesting ifs vs. &&

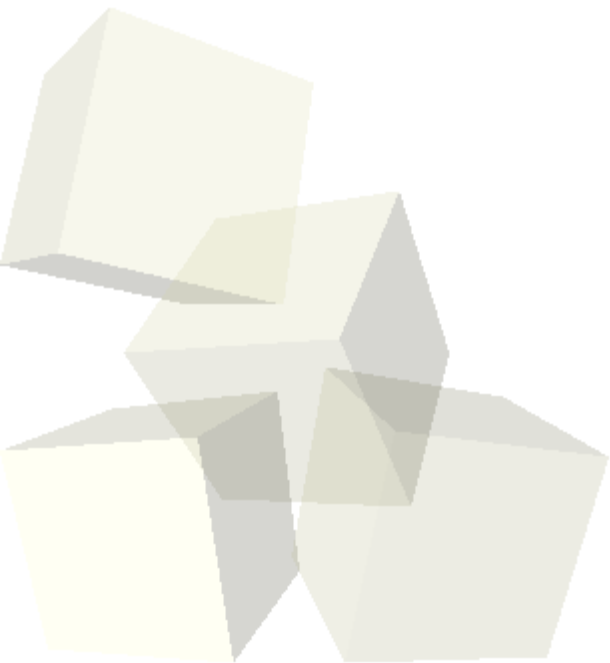
- There are two ways we can get the behavior we want from if statements.
  - ♦ One is to nest one if statement in the else of another one.
  - ♦ The other is to have the if statements separate then use && to narrow things down.
- The former is more efficient and often easier to read. In Alice though the indentation can get a bit extreme.





# Using a Function Instead

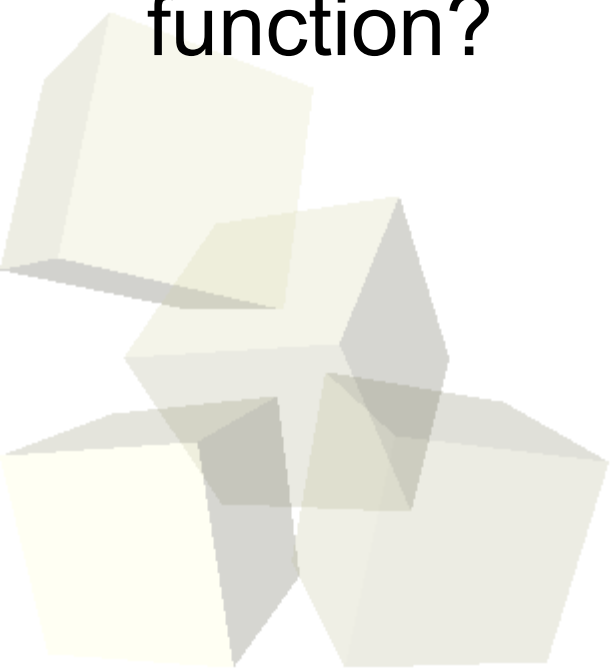
- Let's solve this a slightly different way by putting the logic for deciding what to print in a function instead of in the method.
- This way we just pass to say the results of the function call.

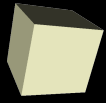




# Validating Parameters

- Fairly often you can have methods/functions that take parameters where not all input should be considered valid.
- Checking to make sure that the input is valid is a common use of if statements.
- We could do exactly that for our grade program. What range of grades should we accept into the function?





- Does the if statement make sense? Do you feel you know how to use it? What do you feel we are still lacking in our ability to make Alice do things for us?
- Quiz #2 is next class. Your ideas for your first project are due a week from today.
- Interclass Problem - Do problem 4.1.

