



# Boolean Expressions and if

9/14/2007





# Opening Discussion

- Let's look at solutions to the interclass problem.
- Minute essay questions.
  - ◆ Do variables, parameters, properties, and functions get more complicated?
  - ◆ Moving to different scenes.
  - ◆ Functions and their role.
  - ◆ How do you make one object pick up another object?
  - ◆ When do you use a variable/parameter/property?





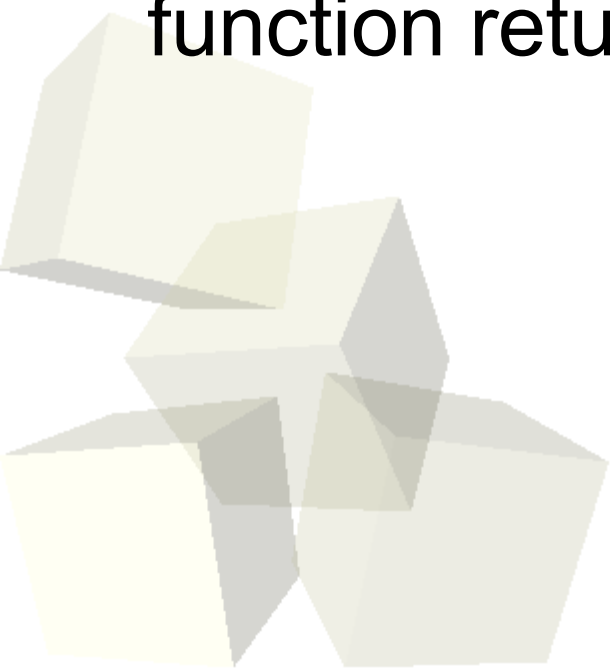
# Conditional Execution

- So far all of our programs have been fairly simple in that the same lines of code happen in the same order every time we run the program.
- Changes in user input or initial conditions might have changed certain details, but basically the same thing happened every time we ran the code.
- Real programs need to have the ability to do certain things only in certain situations. Last time we discussed the skater falling in a hole only when she got too close.
- Having control over what pieces of code execute gives us a lot more power.



# Boolean Type

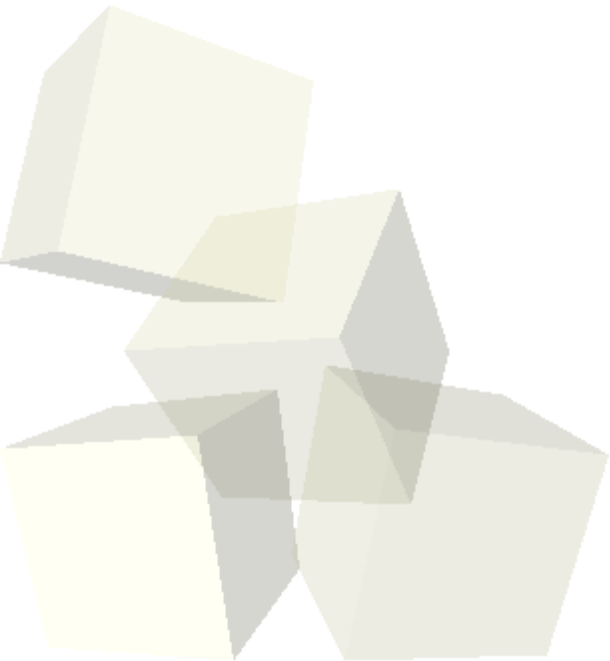
- The Boolean type is an extremely simple type. It can hold one of two values. It is either true or it is false.
- This type is simple, but it is the foundation of simple logic.
- We have the ability to specify Boolean as the type of our variables, parameters, properties, or function returns.





# Boolean Functions

- Alice objects have many functions that return Boolean values. They all start with the word “is”.





# Boolean Expressions

- We can also build complex Boolean expressions using the relational operators and Boolean operators provided in the functions under the World.
- The relational ones are fairly clear other than == for equality and != for inequality.
- The Boolean operators are:
  - ◆ ! - not
  - ◆ && - short-circuit and
  - ◆ || - short-circuit or (not exclusive)





# The if Statement

- The first place we will use Boolean values and Boolean expressions is in the if statement.
- This allows us to check if something is true or not and execute different pieces of code.
- We have already mentioned the possibility of an object falling into a hole if the distance is too small. Similarly we could ask the user if they want certain things to happen in the world.





- Most of the time when you say “or” in English you mean the exclusive form where we don't accept both being true. How would you make an exclusive or using and and inclusive or?
- Interclass Problem – Take your animated object and have it check the distance to another object. If it is more than 3 meters away have it turn toward that object and take a step. Do this three times so the first object walks toward the second unless it gets close in to it.

