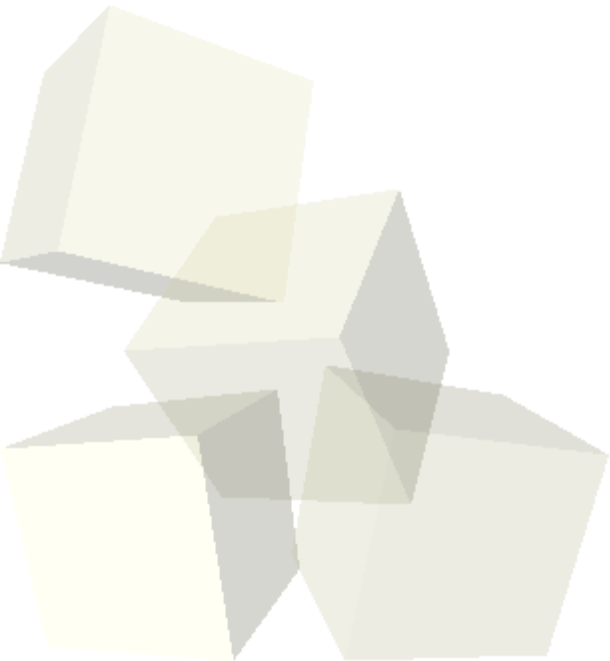


# Generics, Enums, and Inner Classes

1/30/2007





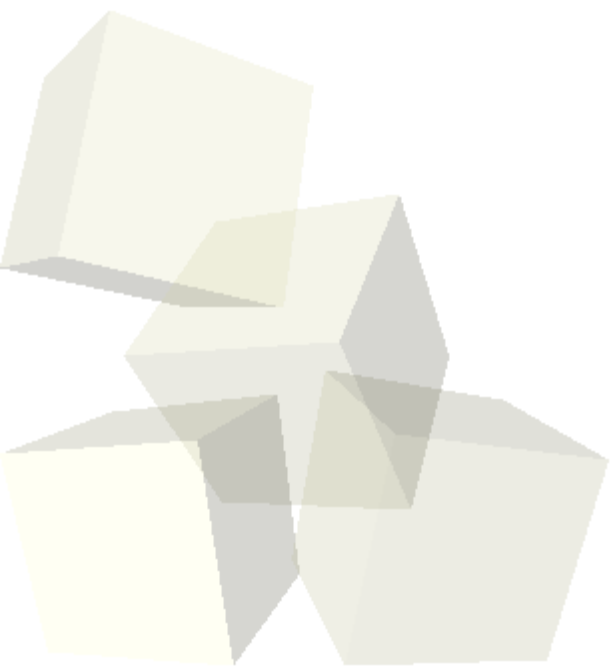
# Opening Discussion

- Do you have any questions about the quiz?
- Large inheritance hierarchies tend to be very brittle. It becomes basically impossible to edit anything at the top.
- What is the purpose of generics?
- What is the syntax of generics?
- What are enums? Did you cover enums in PAD1? How is the Java version different from the C version?
- What is the syntax of Java enums?
- What are inner classes? What makes them special? What types of inner classes are there?
- What is the syntax of inner classes?



# More Function Fun

- It turns out that our little MathFunction interface makes a perfect supertype for creating inner classes.
- We can demonstrate this by thinking of how we can create functions from other functions.
- Let's come up with a few of those and write code to do them.





# More General Functions

- The MathFunctions can be fun, but they are a bit limited because they can only work with doubles.
- Using generics we can create a type that represents a more general type of function that maps from one type to another.
- How can you combine these more general types of functions?





- How did you represent a string in C? How do we represent a string in Java?
- Let's go look at the API for String to see what the Java developers have provided us with.
- Strings are immutable so it is very inefficient to build big strings from a lot of little pieces. If you need to do that, use StringBuffer.
- It is the only class with an overloaded operator, + for string concatenation.
- Functions that looks they mutate the String return a new String instead.
- To get a single character, use charAt.



- Array syntax in Java is just like it was in C as far as using the arrays. There are significant differences though.
- Array types are made by putting [] after a regular type.
- Array types are object types so your variables are references and should be instantiated with new.
- Arrays have bounds checking and know their length.
- You can't make arrays of generics.



- Write the following method:
  - `String replaceAll(String s,char f,char t)`
- This method should return a new string where all instances of `f` in `s` have been replaced with `t`.
- The design for assignment #2 is due in a week.

