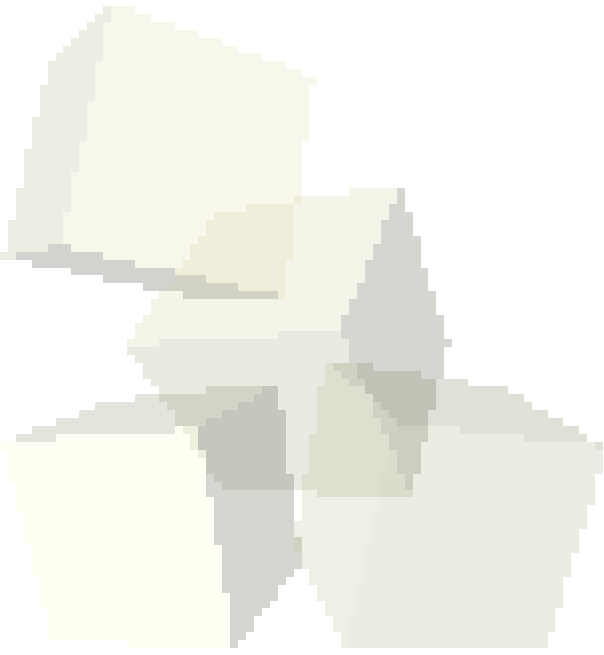




Strings and Arrays

2-4-2010

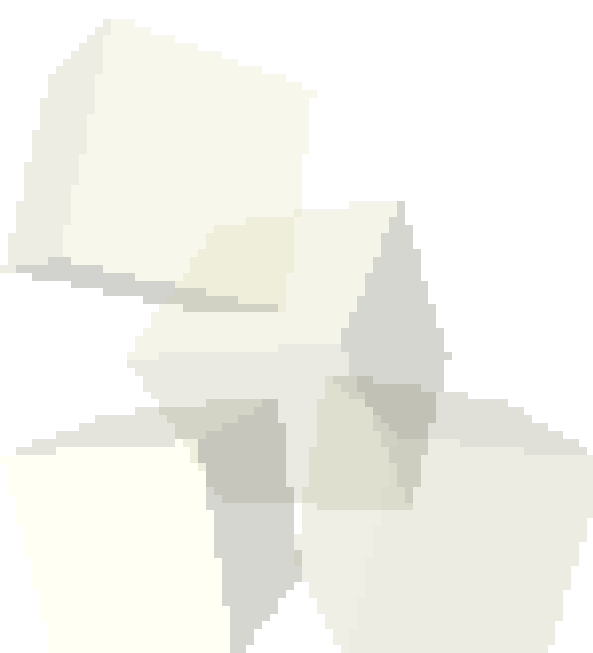




- Do you have any questions about the assignment?
- Minute Essay comments
 - ◆ Seeing the big picture.
 - ◆ What is `Arrays.asList`?
 - ◆ List of common Java syntax?
 - ◆ How are programs like Eclipse written? In Java?
 - ◆ Value of different languages in real life. Using Java for the game/graphics. Sprites?
 - ◆ Does Eclipse only work with Java?
 - ◆ Why not start in Java?
 - ◆ Alternate information sources.
 - ◆ Do you use the Java libraries a lot?
- Javadocs and designs

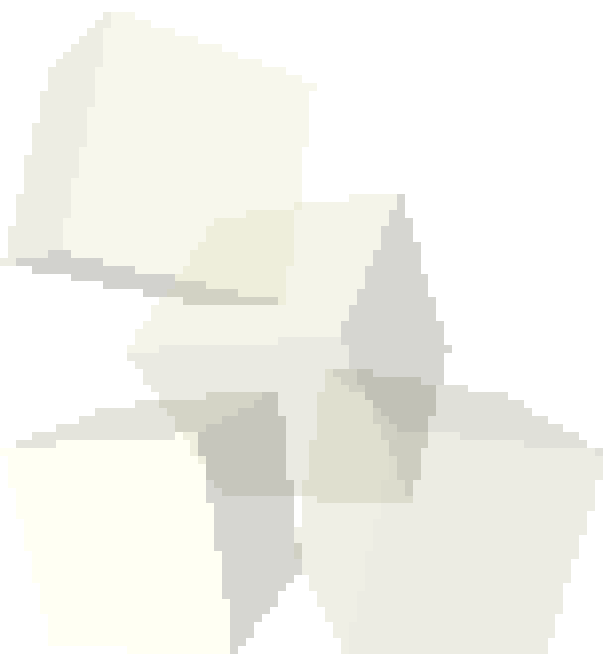


- C had enums. What were they supposed to do? What was the problem with them?
- Java includes enums as well. They serve the same goals, but lack the pitfalls.
- Java enum syntax can get quite complex, but the basic form is simple and very similar to C.



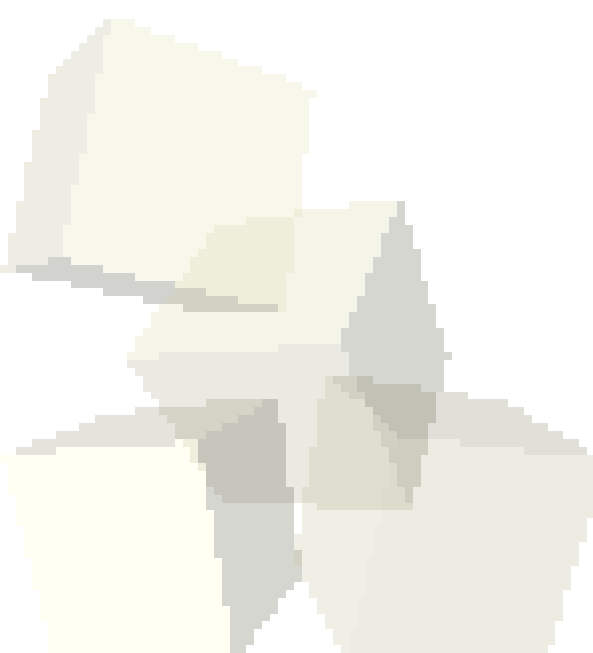


- How did you handle errors in C? (Consider the fopen function.)
- What are some problems with this method?





- Error handling in Java is done with exceptions, not return values or flags.
- Normal exceptions can't be ignored and they don't propagate. Runtime exceptions don't propagate.



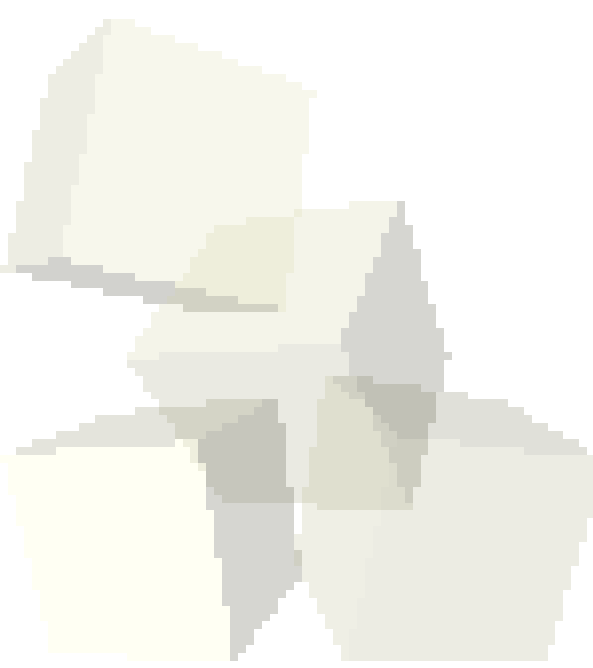


- For anything that isn't a RuntimeException you have to include handling code. For RuntimeExceptions it is optional.
- If you want to deal with a possible exception in the current method do this:
 - ◆ try {
 - statements
 - ◆ } catch(ExceptionType1 e) {
 - statements
 - ◆ } [catch(ExceptionType2 e) { ...} ...]
- If this method can't handle it you add a throws clause to the method and it will go up to the calling method.
 - ◆ Type name(args) throws ExType[,...] {...}



Additional Information

- Exceptions also have the advantage that they can provide additional information.
- Stack trace.
- Informative message.
- You can create your own exception classes. Strive to have them provide sufficient information for debugging.



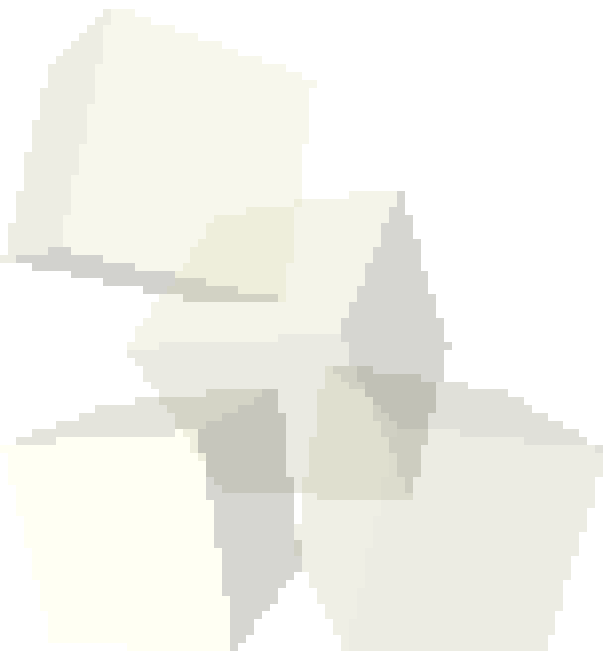


- How did you represent a string in C? How do we represent a string in Java?
- Let's go look at the API for String to see what the Java developers have provided us with.
- Strings are immutable so it is very inefficient to build big strings from a lot of little pieces. If you need to do that, use `StringBuilder`.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



Text/In Class Project

- In class we will be going through the same project that is developed in the textbook. It is a drawing program with a somewhat different structure than paint.
- In addition to drawing, the application will also have command processing that we can use to extend the power of the application.





- To give us a chance to play with strings some let's go ahead and write the beginnings of the command processor.
- We want a class that will take a string, break off the first word as the command, then execute the command with the rest of the string as the arguments.
- We should go ahead and make two commands, echo and add, that we can test it with.





Regular Expressions

- You'll notice that some of the methods in `String` say that they take regular expressions (regex). This is a topic that we won't directly cover in this class, but it is a very powerful technique for you to learn.
- The `java.util.regex` package contains the two main classes that are used for regular expressions. The `Pattern` class has a lot of description about them at the top.

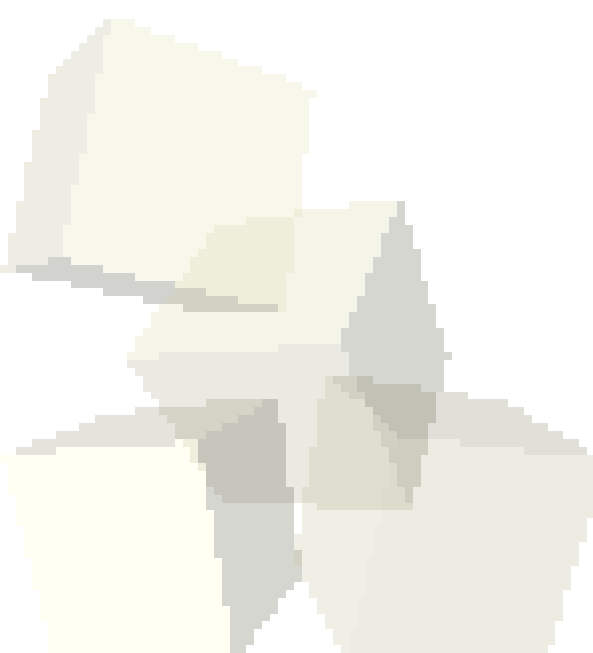




- Array syntax in Java is just like it was in C as far as using the arrays. There are significant differences though.
- Array types are made by putting [] after a regular type.
- Array types are object types so your variables are references and should be instantiated with new.
- Arrays have bounds checking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



- Our command parser was pretty lame in how it picked the command. It definitely isn't very flexible. I want to write a class called `ArrayMap` and use it to make the code more flexible.





- 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`. Don't use the method in `String` to do exactly this.
- Remember to submit your design for assignment #1 today.
- Interclass Problem – Write a program where a user types in a line of text and you print out each word in the line and the number of characters in it.