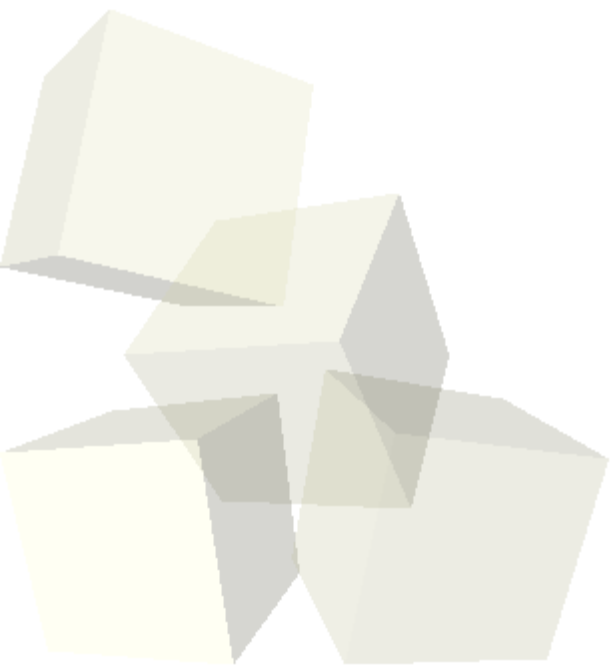




Exceptions and Using Files

4/7/2008





Opening Discussion

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





Problems at Runtime

- Sometimes when you run programs things don't go as you expect. We saw this before with the username program.
- In the case of files, lots of things can go wrong that are outside of our control. For example, the file might not be there or it might not have what we were expecting in it. Networked files can be lost because of network problems. A good program will deal with these things instead of just dying.
- So how should a method tell you if something went wrong and it didn't work the way it was supposed to?



Exceptions

- The way that error conditions are dealt with in Java is with exceptions.
- We can have our own code produce exceptions or code that we call can produce exceptions.
- We saw the latter case yesterday when we were trying to use files.
- As the name implies, exceptions are to be used in exceptional situations. You only use them when something out of the ordinary occurs.
- There are many types of exceptions in Java that are part of the API. Later we will see how we can create our own.



Throwing Exceptions

- If you write a method and something goes wrong in it, you can have the method throw an exception.
 - ◆ `throw new ExceptionType(message);`
- Most of the time you want to use exception types that already exist. That's all we can do with what we know now.
- For example, if we had a method that was supposed to read five ints from a file and there weren't five ints to read we could throw an `IOException` with an appropriate message.





Throws Clauses

- Many exceptions in Java have to be dealt with in some way. `IOException` is an example. If we throw one of these we have to explicitly say that the method will throw it.
- We do this by adding a throws clause after the argument list of the method.
 - ◆ `public returnType methodName(args) throws ExceptionType {}`
- A method should include a throws clause if it knows the exception can occur, but doesn't know how to fix the situation.



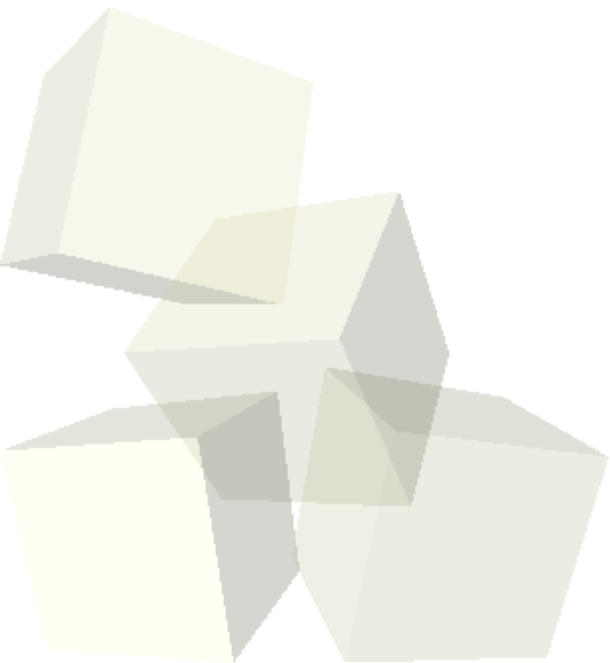
try/catch Blocks

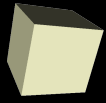
- When a method does know what to do to deal with a particular exception, the code that can cause the exception can be put in a try block.
- This is what we did yesterday with the files though we didn't do much to handle the problems.
- A try block includes code you want to try to do. It is followed by catch blocks that catch each of the exception types that can occur in the try block.

- ◆ try {
 → Statements
- ◆ } catch(ExceptionType1 name1) {
 → Statements
- ◆ } catch(ExceptionType2 name2) {
 → Statements
- ◆ } ...



- Let's play around a bit more today with files. We could finish off the program we started last class and then consider how we would put files into the bank program.





- Exceptions give your program the ability to recover from problems. How does this compare to what happened in Alice programs if things went wrong?
- Note that there is no class on Friday. Do you want to have Mike here to meet and go over questions you might have?
- Interclass Problem – Do problem 11.6.

