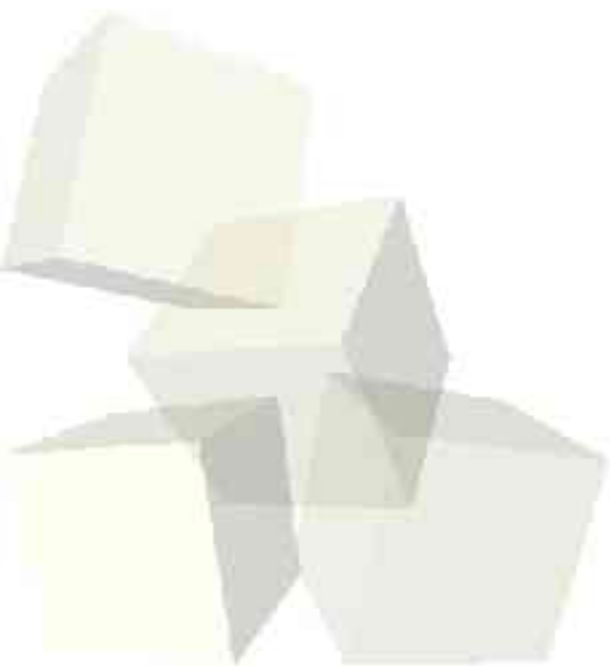




# Using Streams

10-6-2004





# Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?
- The review session will be on Thursday since I won't be on campus this afternoon. We'll do it right after the ACM meeting so I'll come to the lab between 5:30 and 6:00.
- We were going to talk about macros today because your book talks about them, but it turns out that isn't standard Scheme so we won't.



# Using Character Data

- In Scheme we denote individual characters with tokens like `#\A` for a capital A.
- There are also special forms `#\space` and `#\newline`.
- There are some standard functions that we can use when working with character.
  - ◆ `char->integer` – converts to ASCII values.
  - ◆ `integer->char` – does the inverse.
  - ◆ `char=?`, `char<?`, `char>?`, `char<=?`, `char>=?`
  - ◆ `char-ci=?`, `char-ci>?`, ... - ignore case.
  - ◆ `char-upcase`, `char-downcase` – change case.
  - ◆ `char-upper-case?` - check case.
  - ◆ `string->list` – turns a string to a list of chars.



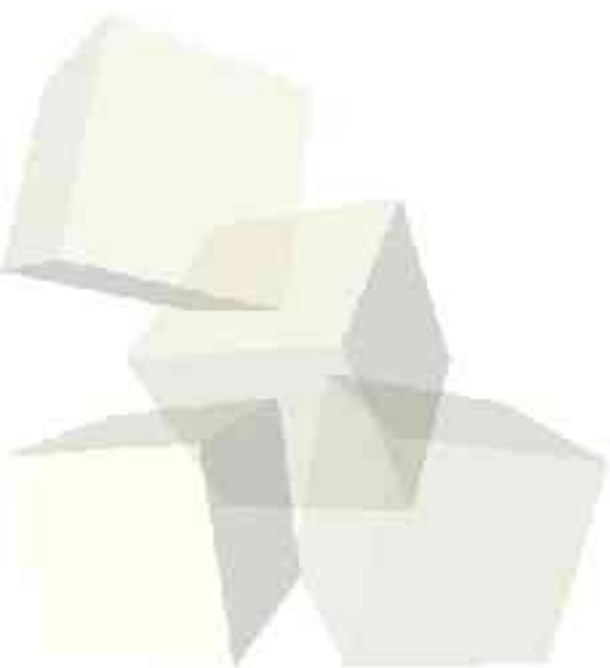
# Files

- Files in Scheme are referred to as input and output ports.
- We create an input port like this
  - ♦ (open-input-file “name.txt”)
- We can read from it with the read command, but providing the port as an argument.
  - ♦ (read port)
- When done reading we need to close the port.
  - ♦ (close-input-port port)



# Input Details

- The predicate eof-object? Tells us if a read object is the end of file marker.
- We can read single characters with read-char.





# Output

- Output works much like input. We use `open-output-file` to get a port.
- We can write with `display` or `write` giving it a 3<sup>rd</sup> argument of the port or write single characters with `write-char`.
- We also need to make sure to close the port when we are done using `close-output-port`.





# Minute Essay

- We start using ML next week after the midterm. Given what you have seen of Scheme, what do you think are its greatest strengths? What are its greatest weaknesses? How would you correct the weaknesses if you designed a new functional language?
  - Don't forget the midterm is Friday and that we have a review session tomorrow.
- 