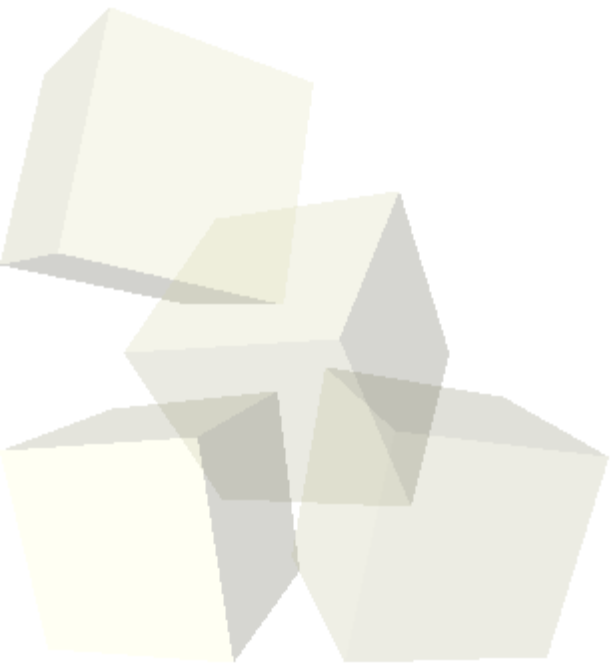




# Binary I/O and Pointers in Files

11/16/2007





# Opening Discussion

- Let's look at solutions to the interclass problem.
- Do you have any questions about the assignment?





# Using Binary Files

- Working with binary files we use `fopen` and `fclose` just like normal, but you need to specify that it is a binary file in the `fopen`.
- Instead of `fprintf` or `fscanf` we use `fread` and `fwrite`.
- These functions directly copy from disk to memory or from memory to disk.
- Let's look at the man pages for these functions.



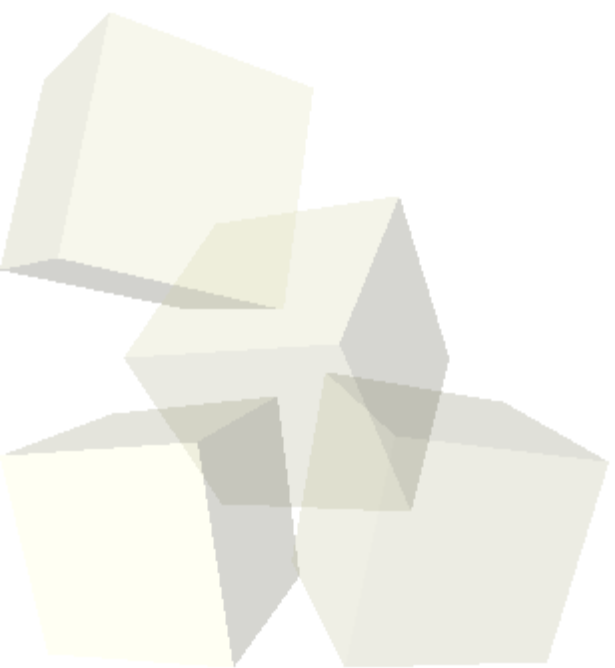


# Random Access

- What really makes binary files significant is the ability to jump to any place in the file to read or write.
- This random access ability makes them very fast for accessing large amounts of data. Databases are built around this ability.
- The simplest form of direct access involves fixed length records. This is basically like having an array on disk.
- The `fseek` and `ftell` functions allow us to know where we are in a file and jump to new locations.



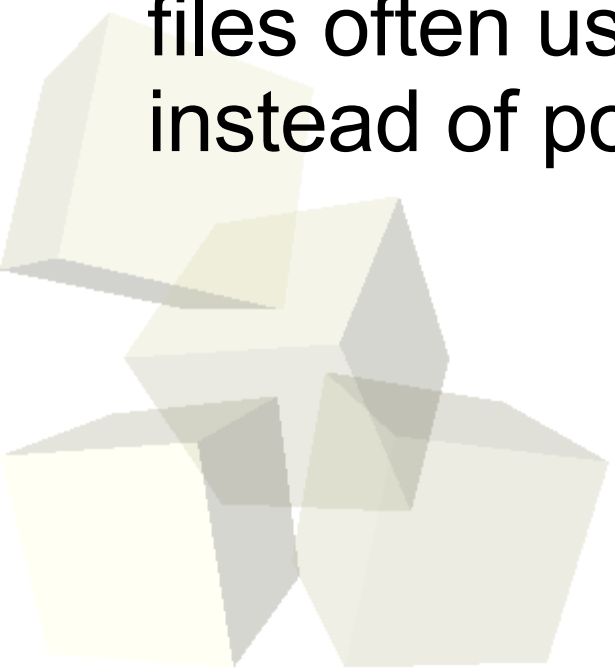
- Let's write some code to see how this works.





# Pointers in Files

- Structures that include pointers cause problems for files just as they can for assignments and pass-by-value.
- Writing a pointer to file serves no point because that pointer will not point to a valid memory address the next time you load in the program.
- For this reason data structures that are written to files often use integers that reference into arrays instead of pointers.





# Minute Essay

- Not all records work well as fixed length. What do you think you need to do to efficiently implement a file with variable length records?
- Interclass Problem – Do problem 39 on page 877.

