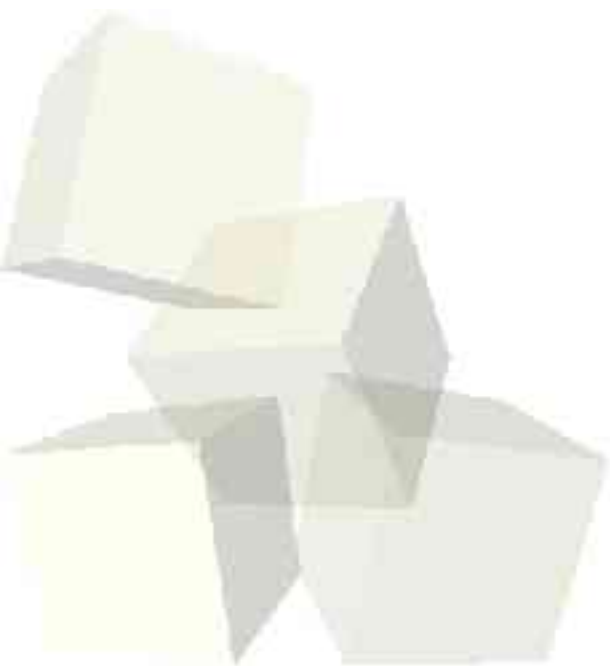




The ML Module System

11-12-2004





Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?
- How do you build modules (groups of related functions a structures) in the languages you have used previously?





Modules in ML

- Modules serve two primary roles in languages. Prevent others from messing with things they shouldn't, and increase code reuse.
- ML provides a number of different constructs that help us to implement modules efficiently.
- These constructs focus around the concept of a structure. We have used structures previously because many of the basic functions of ML are in modules.



Structures

- We can define our own structures in the following way.
 - ◆ `structure <ident> = struct <elements> end`
- The elements can be function definitions, exceptions, constants, or types.
- Let's go see how this works by putting our binary search tree in a structure. Note the response that ML gives to us when we load it in.
- Structures can be nested.



Signatures

- A signature is, roughly speaking, a type for a structure.
- We can specify a signature with the form
 - ♦ sig <specs> end
- The specs can include the keywords type (type ('key,'data) bnode), eqtypes (works like type, but says it must be an equality type), exception (providing the name), and val (followed by an identifier, colon, and the type expression). They are separated by semicolons.



Names for Signatures

- We can give a signature a name using the signature keyword
 - ♦ signature <ident> = sig <specs> end
- We are given the ability to name signatures so that we can easily specify a particular signature for a given structure. We do this so that we can hide certain parts of a structure. Only the parts in the signature are usable.
- The syntax is
 - ♦ structure <ident> : <sig> = <struct>



Details

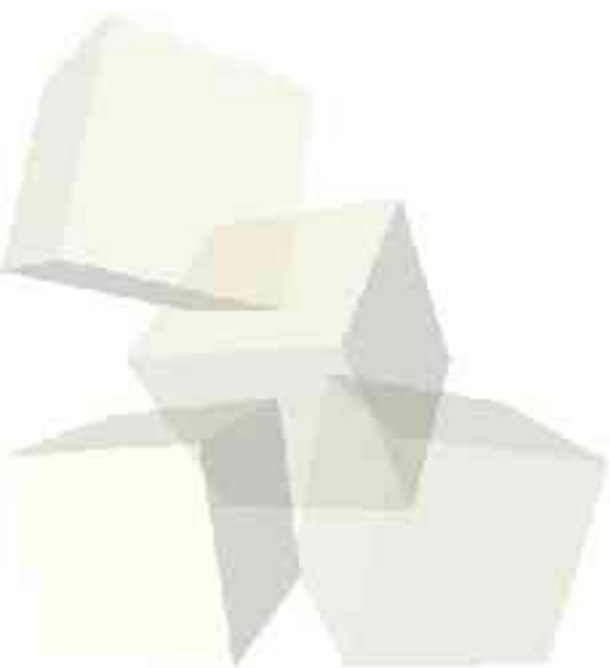
- Notice the difference between structure and struct and also signature and sig. The first in each pair begins a declaration. The second in each pair goes with end to bracket the elements in them.





Code

- Let's go ahead and write some other code that can be put into structures. One possible example is vector math libraries.





Minute Essay

- What are the benefits of using structures?

