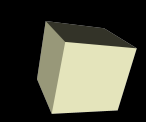
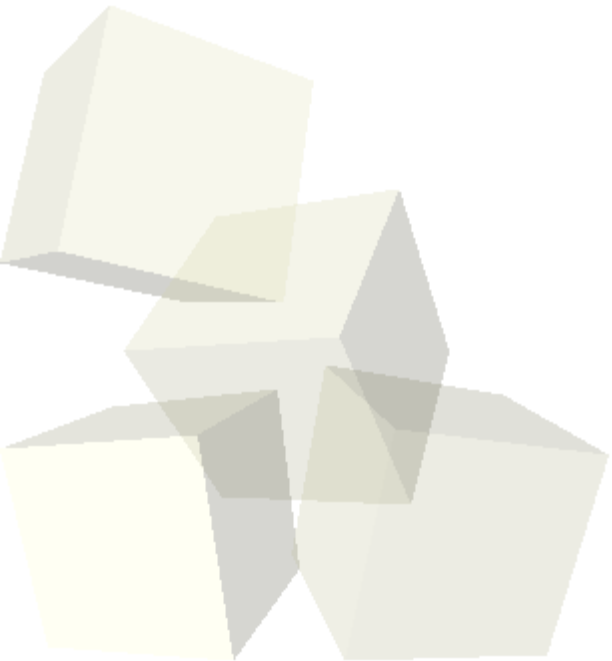


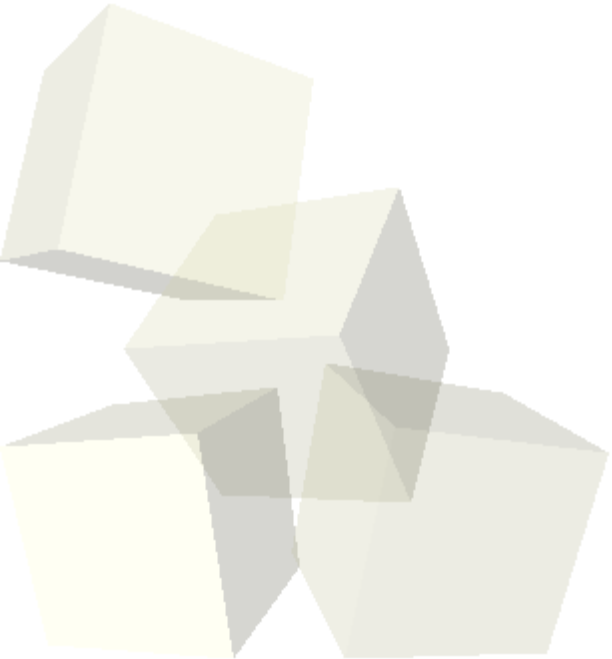
Enumerations and Structures

11-9-2006



Opening Discussion

- What did we talk about last class?
- Do you have any questions about the reading?
- Do you have any questions about the assignment?



User Defined Types

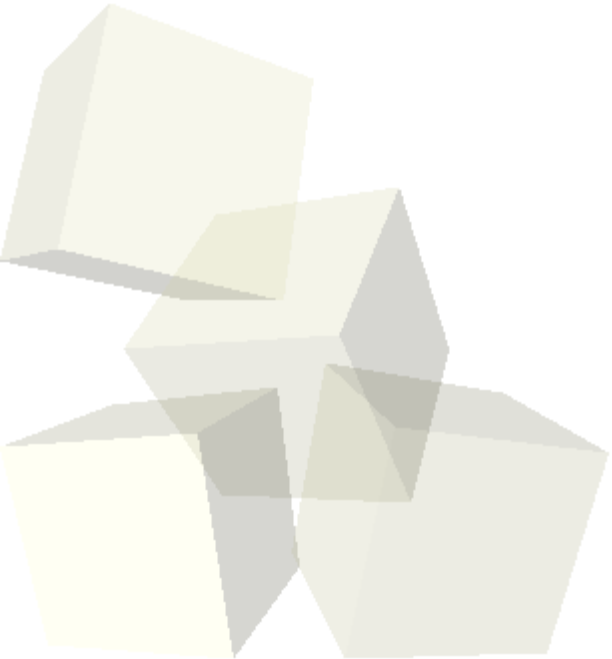
- We are now getting into a new area of programming. So far all the types that we have discussed were built into the language. That allows us to do anything we want, but it is much less than ideal for large programs.
- Our abilities to manage large programming projects take a significant step forward when we can define our own types. This lets us give names to things that have more meaning or to group information in ways that suit our particular application.
- What possible applications of this can you think of?

typedef

- The simplest activity that we play with the typing system is with typedef. A typdef statement simply creates a new name for an existing type.
- This might not seem too useful, but it can have significant benefits for creating code that is easily ported across multiple machines or when type names get really long (that's more of a C++ issue).
- The syntax for typedef is as follows.
  - typedef ExistingType NewName;

Enumerated Types

- There are times when you have items in a program that should only take on one of a small number of values. For example, you could have a variable representing the color of a street light.
- Given what you know, how would you implement a variable for a street light?
- What are some of the limitations of this method?



Syntax of enum

- An enum declaration creates a new type and various constants that are of that type. These are really just ints, but some type checking will be done to prevent you from mixing things up.
  - enum TypeName {list};
- The list needs to contain names separated by commas. Each name can optionally be given an integer value. If nothing is specified they will be numbered in order beginning at zero.
- You declare enum variables with enum followed by the TypeName.
- Because they are ints they can be used in switch statements.
- Anonymous enums define constants.

Structures

- We have seen how arrays allow us to group together many things of the same type and access them by numeric values. What do you do if you want to group things of different types?
- Consider the text adventure game option for assignment 7/8. Each room has several pieces of data associated with it. Many of those pieces are strings, but they have different lengths. There are also some ints mixed in.
- Another possible application would be a grade book. That application will need to keep track of students and their grades. Each student has a sting for a name and some numbers for grades.

Syntax of Structures

- Structures are declared with the keyword struct.
  - struct Tag { fields };
  - typedef struct {fields} typeName;
- In the first case the name of the type is struct Tag, for whatever you listed as the Tag.
- The fields are basically like variable declarations and they specify all the things that are in that structure.
- We refer to the things in a structure variable with a dot.
- If we have a pointer to a structure we can use -> as a shortcut.

Minute Essay

- For a graphics problem I want to have a structure that represents all my geometry. That can include multiple lights and spheres. Write a structure(s) you might use to represent this.
- Remember that nothing is due today, but that is because the assignment is rather large and if you don't start this week your life will be very unhappy next week.
- We have a quiz next Tuesday.