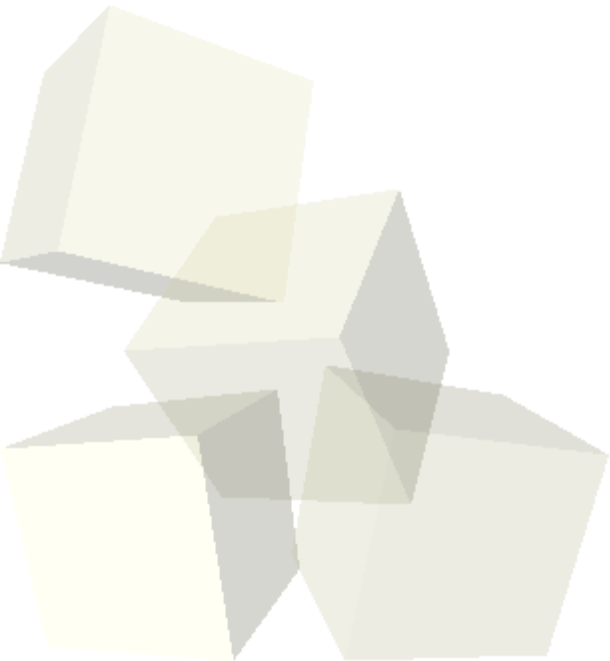




Libraries and Instance Methods

3/10/2008





Opening Discussion

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





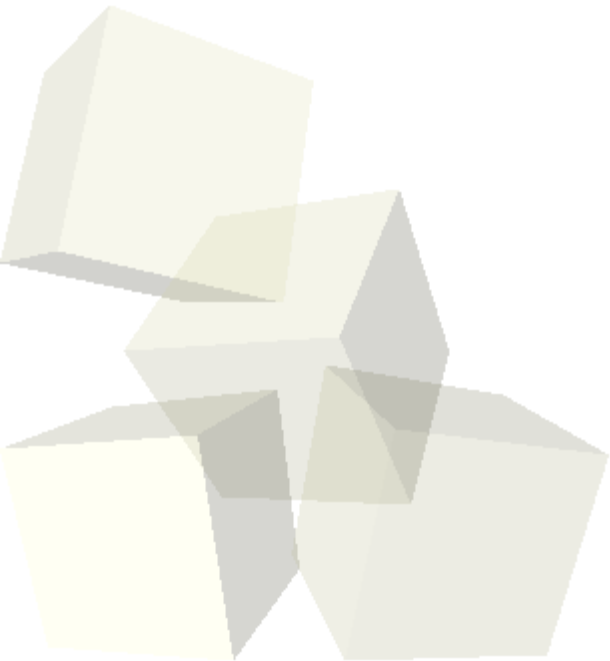
Testing and Unit Tests

- In order to be certain that a program actually works, we have to test it.
- A unit test is a test written to check the functionality of one particular class. We typically create them by making another class that uses the first one and does things we can verify the answers to.
- The assert directive is a way of checking whether the answer to something came out the way we expected it.
- There are also tools like JUnit that can help with building and running unit tests.



Problem

- I'd like each and every bank account we create to have a new account number. This way no two accounts have the same number.
- I'd also like to have a method that will validate if a particular account number is allowed.
- How can we do this? What is the challenge we run into?





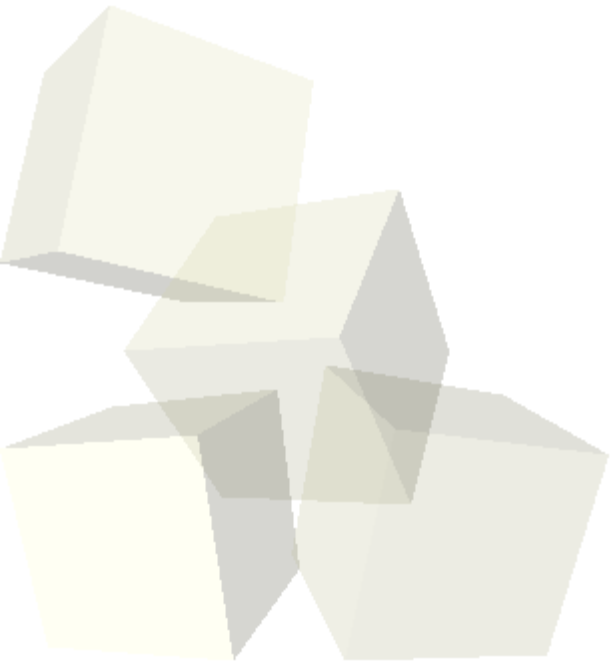
Instance vs. Static Methods/Data

- The solution to our problem lies in the use of static. We need something that is associated with all bank accounts, not just one.
- To accomplish our task, let's put a static int into the BankAccount class. This will store the number of the next bank account.
- Every time we make a BankAccount (remember that calls a constructor) we change that number in predictable way.
- The balance is non-static. Every BankAccount gets its own. On the other hand, nextAccountNumber is static. All BankAccounts share it.



Money and Customers

- Let's finish what we started last time and make methods that give us strings that nicely format the value of the balance.
- I'm also unhappy with our way of keeping track of who the account belongs to. If one person has more than one account with our bank we run into some odd situations.





- Imagine a class called Student. What properties and methods might you put in that class?
- Interclass Problem – Do problem 9.10.

