More Linked Lists

11/28/2007

Opening Discussion

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

Doubly Linked Lists

- Last time we drew singly linked lists where each node knows only about the next item in the list.
- In a doubly linked list, each node keeps a pointer to the next and the previous nodes.
- This allows us to delete a node when we have the node, unlike a singly linked list where you need the node before it.

Circular Lists and Sentinels

- You can make linked lists that link back around on themselves and these are called circular lists. They can be singly or doubly linked.
- Linked list code can be simplified by keeping an “empty” node in a circular list called a sentinel.
- In a singly linked list it replaces the head. It's even better for a doubly linked list where it is head and tail and it greatly simplifies the code.

Coding Linked Lists

- You've seen how to do an array based list with the basic functionality that we discussed.
- We also showed how to do linked lists in drawings.
- Now we need to write code for our linked lists.

Polymorphism in C

- C doesn't really have good polymorphic abilities, but we can get around that using void* and function pointers.
- Using void* allows you to use any type, but it requires lots of use of dynamic memory.
- The function pointers allow you to pass functions in to other functions. This is required in many cases where the code doesn't know what type it is working with.

Minute Essay

- Do you see any shortcomings with the way we code lists in C? If so, what are they?
- MattCoffer.age++; That means it is fountain time.
- Interclass Problem – Modify our add method so that the values stored in the list will be in sorted order. This means the insert method is no longer valid because you aren't allowed to pick where something goes in the list.