# Stacks and Queues

**2-4-2002**

# Opening Discussion

- What did we talk about last class? Do you have any questions about the assignment?
- What is a stack? What is a queue? How are they different? How are they the same?

# Stacks

- A stack is a general LIFO data structure. That is, it is a way of storing data where when you take something out of it, what you get is always the last thing you put in.
- As stack only has to have 2 operations, though we often put in more to make things more functional. Your text gives a top method and has pop not return anything.

## An Array Based Stack

▌ Last class we wrote some code to implement an array based stack. Today we will play with that some more, but first we should discuss what it needs to do.

▌ With an array for a stack we keep a single integer value that tells us how many things have been pushed onto the stack. This is often called "top". Push increments top and pop decrements it.

## Queues

▌ The partner of a stack is called a queue. In a queue the thing you take out is the one added earliest. This is a FIFO structure.

▌ Queues also have two operations which most people call enqueue and dequeue. I'm fond of calling them push and pop but I'll try not to do that this semester.

## An Array Based Queue

▌ As with stacks we can write queues using arrays. They are a bit trickier though. This is because we want operations to happen in O(1) time.

▌ Because the queue has things happening at both ends we have it "wrap" around the array when it needs to. This can be implemented quite easily using modular arithmetic. We'll look at an example in a minute.

## Code

▌ Now let's look at the code we wrote last class and make it more functional.

▌ Let's also write a templated queue class. There are some more significant design decisions here that we didn't have to face with the stack.

---

## Minute Essay

▌ Hopefully you have seen stacks and queues before. Is this true? Do you have any questions about them? Can you think of any uses of stacks or queues that might come up in the project for this semester?

▌ Remember that the first assignment is due Friday. You should probably start the code soon so you can ask questions you might run into on Wednesday.