

# **Sorting, Searching, Stacks, and Queues**



**9-28-2004**

# Opening Discussion



- Do you have any questions about the quiz?
- What did we talk about last class? Do you have code to show?
- Do you have questions about the assignment?
- Private not only protects your data, but parts of code as well. It reduces dependencies so you are more free to change without breaking other code.

# Back to Basics



- Let's look at what needs to go in a class in Java and what it means.
- The class collects methods for things instances of it should do and data for the state of the objects of that type.
- All classes in Java inherit from Object as a base class. That's why we can make polymorphic code by passing something of type Object.

# Abstract Data Types



- The main topics for today are Stacks and Queues. These are the simplest forms of what we call “Abstract Data Types”. An ADT is basically an entity that stores data in some unknown form, and provides us with a standard interface for dealing with it.
- Good data structures in general are ADTs. They demonstrate the separation of interface from implementation.

# The Stack and Queue ADTs

- The stack and queue are the simplest ADT because they each have only 2 methods in their interface (or 3-4 depending on the details of your definition).

- Stack

- Push: adds something new to the stack.
- Pop: removes the "newest" thing.

- Queue

- Enqueue: adds something to the queue

# LIFO vs. FIFO



- A stack is a last in, first out data structure.
- A queue is a first in, first out data structure.
- You will commonly here these referred to with the acronyms LIFO and FIFO.
- This really is the only difference between the two as for as a mental model of them is concerned.

# Array Implementation of a Stack



- Stacks can be implemented in many ways. There is even a simple implementation with an array.
- By just keeping an integer index to the “top” of the stack we can easily write the methods.
- The other methods often added are a peek method to look at the top of the stack without popping and an isEmpty method.

# Array Implementation of a Queue



- As with a stack, a queue can be fairly easily implemented with an array, but here there are a few more details to worry about.
- Now we need two indexes for the front and back of the queue. Array based queues also need to be “circular” or they run out of space quite quickly.
- Again, peek and isEmpty are helpful.



# Minute Essay



- Draw a picture of what the array for a stack would look like after these operations:
  - `push(5)`
  - `push(7)`
  - `pop()`
  - `push(2)`