Linked Lists

* Data structures
* Ways to store arbitrary amounts of data

Let’s look at ways to repackage the LinkedList into other types of data storage

* Stacks
* Queues

Both stacks and queues are “serial” data structures – you can insert single data elements one at a time, and you can remove single data elements one at a time

Queue:

* works just like waiting in line
* data objects are inserted into the queue, and the order of insertion is remembered
* the object that has been waiting the longest is the one that is removed first
* enter(), leave()

Stack:

* data objects are inserted into the queue, and the order of insertion is remembered
* the object that arrived last is the first one to leave
* Someone washing dishes
* push(), pop()

How do you implement these using a linked list?

CSE 1030 Final Exam

* Wednesday April 9 9:00 AM
* Written + labtest
  + Written part will be on paper
* 2 80-minute parts with a 20-minute break in between
* Written part: LAS A
* Labtest part: Prism lab
* ½ the class will write first, the other half will do the labtest first
  + family name is early in the alphabet: written part first
  + late in the alphabet: labtest first
* During the 20-minute break, switch
* For persons with accommodation, you will do the LABTEST part first

Material:

* All material is testable
* However, the emphasis will be on material covered since the 2nd labtest
  + Recursion, data structures