CS-21 LECTURE #1

Day 1 Agenda:
- Course overview
- Canvas info
- Class notes
- Assignment #1

Second Half
- Linked list review
- Stacks review
- Queues review

Third Half
- Pengo compile environment
- Vim & emacs for
- STDIN, STDOUT
- command line
- C++. Java on pengo server.

PENGO USER: mginella

command "passwd" to change password

- **REQUIRED PROGRAMMING ENVIRONMENT:** VIM & EMACS

- **Pengo** = Server where Java or C++ programs will be turned in & tested. Ubuntu server.
  - **PASSWORDS** = USER + "p" **ex:** sjohnp

- **What doesn't count for Lab Work:** Reading textbook, coding assignment time, both don't count
  - ↑ Unless the reading isn't assigned. Then it does count!

- **YOU HAVE 3 FLEX DAYS NOT TWO:** You have max 24hrs per flex day to turn in late assignments for full credit.

- **NO SUBMIT STEP. PROGRAM WILL BE COPIED FROM DIRECTORY**

- 2 EXAMS.
  - 1 MIDTERM, 1 FINAL.
    - primarily writing memorized code. Paper written.

- ASSIGNMENT #1 DUE FEB 13th!
  - Create a makefile w/ executable named p1
  - Create a double linked list program. You can use the program from Jeffs Class.
  - Stuill class uses composition to take features from List to make things easier.

A REVEW OF LINKED LISTS, QUEUES, & STACKS

Standard OOP Linked List

- two classes

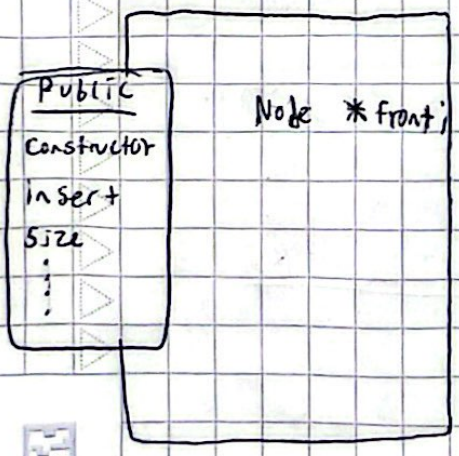
List object - Exactly one

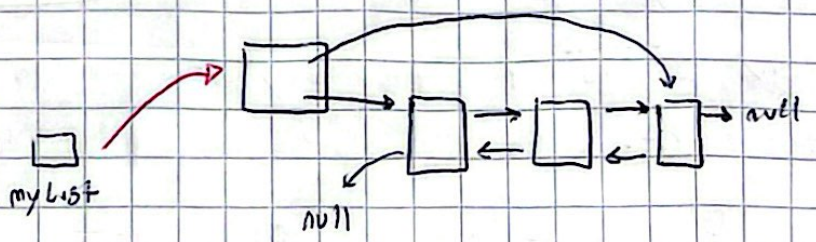
Node object - One or more

(C++)

List Object

EX List myList;





Basic List Traversal

C++

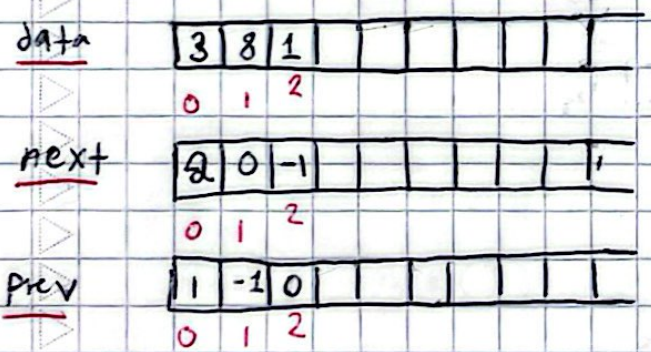
```
void List::print() {
    Node *temp = front;
    while (temp) {
        cout << temp->data << endl;
        temp = temp->next;
    }
}
```

will keep going until temp is NULL

dereferences data in Node class...

Parallel Arrays

front 1
back 0



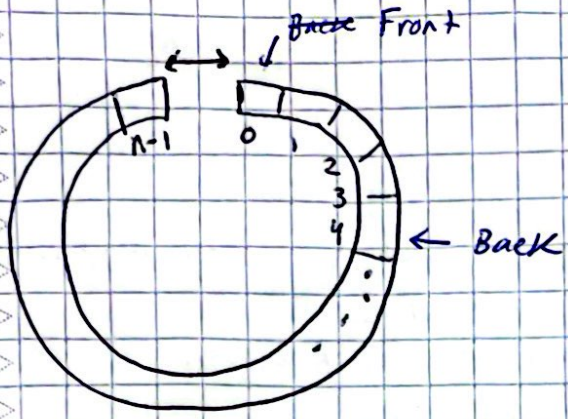
STACKS & QUEUES

interface:

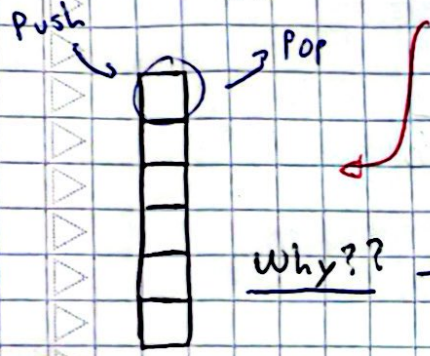
Array — partially filled array.

size : 6

Circular Buffer



Stack data structure where you can only add or remove from one end



you can only pop or push to top of ~~stack~~ queue.
Its like a stack of trays, you can only access the top.

Why?? → TIME!

Queue dequeue ↖

↖ enqueue

↳ FIFO "FIRST IN FIRST OUT"

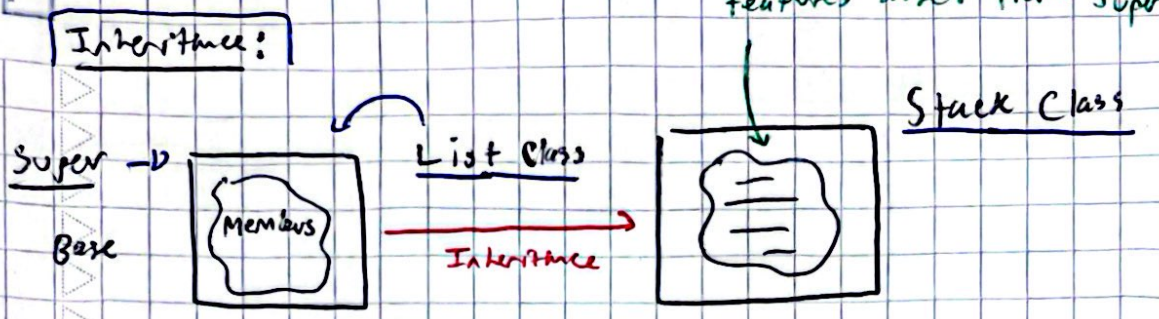
COMPOSITION VS INHERITANCE

```
Class List {
    public:
        Insert_Front(x) { → }
        Insert_rear(y) { → }
}
```

Composition: "has a" ↙ "I have a Stapler"

Inheritance: "is a" ↖ "I AM the Stapler"

Inheritance:



Inheritance C++ ?

```
" Class Stack : public List { "

public Push (int x) {
        insertFront (x) ;
}
```

Function from List class used in stack class due to inheritance.

Composition:

Java...

```
class Stack {
    :
    Stack() { myList = new List() ; }
    :
private List myList;

public Push (int x) {
        myList. insertFront(x);
}
```

you gotta explicity mention that its from list