CS 2100

Vectors

Recap

- Due tomorrow:

  HW3

- Due Sunday:

  Lab 5

- Due Monday:

  Reading (2 sections) (by 2pm)

- Next HW: let's take a look at it!

  may work w/ partner

  submit .h & readme

- Git instructions: See webpage, & either try today or this weekend!

& ~~question: is Curtis Dawkins here?~~

Last time: Queues

Operations:

- push (not stack)
- pop
- front
- empty + size

Trade-off / uses:

simple + fast: O(1)

not much access to data

Today: Vectors

Similar to lists in Python

(will see in 3.5 of zyBook)

Our implementation:

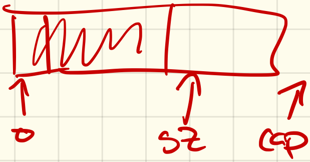
- array based

Main functions -

see STL, because there are a lot of them!

Private data (in C++ version):

```
Object * A;
int capacity;
int sz;
```

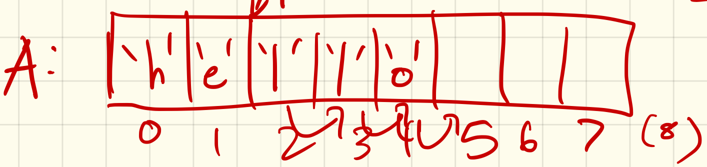


To think through:

```
myvec.insert(2, 'c');
myvec[2]='c';
```

How to insert, if we don't
want to lose data?

put 'c' between 1 & 2



size = 5
cap = 8

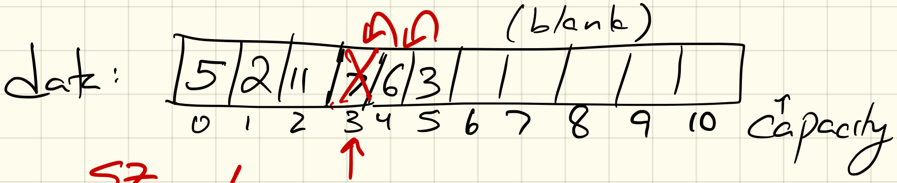
in class

```
Object& operator[](int i){
   return A[i];
}
void insert(int i, Object o){
   for (int j=sz; j>i; j--)
      A[j]=A[j-1];
   A[i]=o;
   sz++;
}
```

Similarly, erase:

myvec.erase(3);

Underneath:



~~Size~~ sz = 6

capacity = 11

```
erase(int index){
    for (int i=index; i<(sz-1); i++)
        A[i] = A[i+1]
    sz--;
}
```

Forgot: error handling

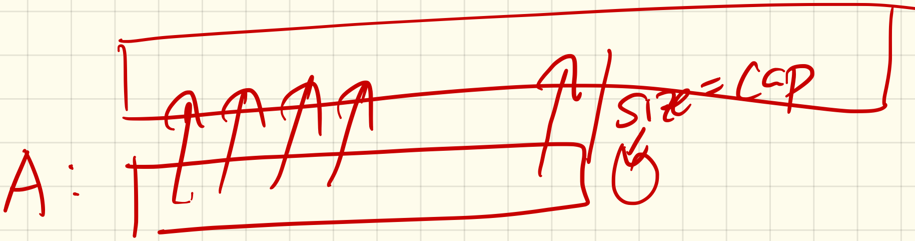
Another issue:
& what if what if we exceed the capacity?

Increase capacity automatically.

push_back & insert will double array size

in fcn:

```
if (sz == capacity) {
    capacity = capacity * 2;
    Object* temp = new
        Object[capacity];
    // for loop to copy data
```



Finally, don't forget housekeeping!

Will look like ArrayStack or ArrayQueue

Next time:

Reading on both:

- array-based lists
- doubly-linked lists

Implementaton:

- Vector.h