CS2100

Vectors (cont)

Recap

- No lab/hw due this week
- Next HW - up later today, due next week
- HWs 1-3 are graded
  (go pull + look for grades.txt in each folder)
  ↳ check for HW3 in particular

Last lecture

Vectors:

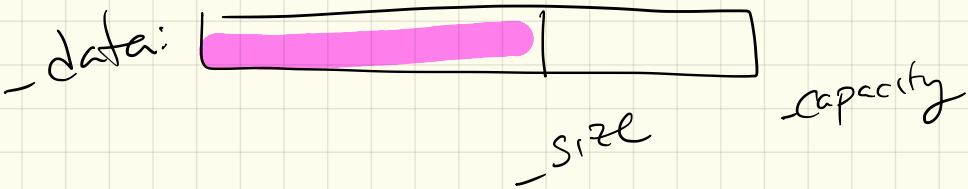
- basically a more robust array
- incorporates list-like functionality
  - no max size
  - Python class

We coded:

- constructor
- destructor
- size & empty
- insert ⚠ double the array anytime size == capacity then no maximum size

Picture:

3 private variables:



Goals:

myvec.insert(10, 'A')

↳ put 'A' in slot 77 & make room

myvec[10] = 'A';

↳ overwrite slot 10

```
Object& operator[](int index) {
    //check if valid
    return _data[index];
}
```

Today:

- erase (i) → copy data down
- push_back
- popback

- housekeeping ✡
  - ↳ copy cons.
    operator=

Next HW: adding more

Run times:

insert: $O(n)$

Why? for loops that in the worst case go 1 to n or n → 1 ≈ size plus O(1) of other lines

size & empty: $O(1)$

[] fcn: $O(1)$

erase: $O(n)$

push_back:

if size < cap: $O(1)$

if size == cap: $O(n)$

Amortized analysis:
"average" case

n push_backs in empty vector:

lots of O(1)

10 → single long one

10.0 [lots of O(1)

20 → single O(n)

20 [⋮

We doubled cap when size = cap