Administrivia

• Partner survey is out, please fill out **EXACTLY ONCE** per pair (due midnight tonight)

• Homework 5 is due midnight tomorrow
  
  • Come to office hours, we can help

  • I’ll have extra office hours today at 3:30pm to help
types in C

• At its core, a type is nothing more than how to interpret some number of bytes
  • an int is just a signed number interpretation
  • a pointer is just a memory address interpretation
  • a struct is just a box which holds other types
typedef struct point {
    int x;
    int y;
} point;

point* a = (point*)malloc(sizeof(point)+512);
//a is still a point
//with some empty space after

a->x = 5;
a->y = 3;
malloc review

```c
point* a = (point*)malloc(sizeof(point));
point* b = (point*)malloc(sizeof(point));
```

What can I say for sure about `a` and `b`?

What can I say if after I make sure they’re not NULL?
structs

where should you *declare* them?

where should you *define* them?
Questions?
List Example

- Emphasizing proper abstraction