Structs/classes
Ch 10.1 - 10.3

How does computer programming work?

MAGIC.
Highlights

- classes

```cpp
class blah2 {
    public:
        void foo();
    private:
        std::string word;
};
```

- structs

```cpp
struct position {
    int x;
    int y;
};
```
Recursion

How would you solve a sudoku problem?
Rules:
1. Every row has numbers 1-9
2. Every column has numbers 1-9
3. The nine 3x3 boxes have numbers 1-9

Reduce problem?
Stopping case?

(see: sudokuSolver.cpp)
Recursion

Do not try to solve chess in this manner!

You will segfault
(you will also not finish computing before the sun burns the earth to a crisp)
Miscellaneous notes

Try googling “recursion” and click on the spelling suggestion

Recursion is very powerful and used in many advanced algorithms

It will give you a headache for a while... =(
class vs array

Arrays group together similar data types (any amount you want)

Classes (and structs) group together dissimilar types that are logically similar
classes

classes are outlines/blue prints of an organization structure

Thus when you create a variable of your class's type, you create an instance
classes

Suppose you wanted to write a function to find the maximum element in an array. How would you return both an index and the element?
Suppose you wanted to write a function to find the maximum element in an array. How would you return both an index and the element?

1. Use a global variable to share between functions
2. Use call-by-reference (See: findMax.cpp)
classes

A class is a grouping of (similar) objects

```cpp
class closet{
public:
    string belts[10];
    string shoes[20];
    string shirts[40];
    string pants[30];
    string dresses[20];
};
```

(See: findMaxV2.cpp)
classes

You just made your own data type (just like int/double/char/etc.)

You can make as many variables of this type as you want.

The dot operator tells the computer to go inside the object/container.

class twoInts{
public:
    int first;
    int second;
};

twoInts x = findMax(numbers);
cout << "Maximum is numbers["<<x.second<<"] = "<<x.first<<endl;
You need the dot to differentiate between two different variables.

You can also think of the dot as possessive in English ( . → 's )

```c
struct date
{
    int day;
    int month;
    int year;
};
```

```c
int main()
{
    date today;
    date midterm2;
    today.month=3;
    midterm2.month=4;
}
```
classes

Suppose we are planning to redo all the classroom name plates in Keller hall.

How would you store all the room information...
... without classes?
... with classes?

(See: room.cpp)