CS 161
Intro to CS I
Finish Pointers/Start Recursion
Odds and Ends

• Last week to demo Assignment 3!!!
• Assignment 4 questions

```cpp
// toupper and tolower

string message = guessed_mssg;

cin >> message

guessed_mssg = message;

for all chars in guessed_mssg

if (lowercase, at(i) != )

at(i)
```
```cpp
#include <iostream>
#include <string>
using namespace std;

void fun(string *s, string *t) {
    cin >> *s;
    cin >> *t;
}

void fun(string &s, string &t) {
    cin >> s;
    cin >> t;
}

int main() {
    string str, str1;
    
    fun(&str, &str1);
    cout << str << " " << str1;
    
    fun(str, str1);
    cout << str << " " << str1;
    
    return 0;
}
```
More Understanding Pointers

- What if you made a pointer that points to a pointer to an int, i.e. `int **p2`? Now, set p2 to point to p, and use p2 to print the address and contents of the int s variable!!!

```
cin >> *p2;
p2 = &s;
```

```
int s = 30;
int *p;
```

```
*p = &s;
```

**compiler error:**

```
10 0x400
0x300
```

```
never cavalier unless you have set *p
```
Recursion

• **What is it?**
  – Function that calls itself 1 or more times (directly or indirectly)
  – Has 1 or more base case for stopping
  – Inductive reasoning: general case must eventually be reduced to a base case
Example: Drawing Rectangles

• Iterative Solution:

```cpp
void draw_rect(int i) {
    for(; i > 0; i--){
        cout << "********" << endl;
        cout << "*         *" << endl;
        cout << "********" << endl << endl;
    }
}
```
Example: Drawing Rectangles

• Recursive Solution

```cpp
void draw_rect(int i) {
    if(i>0) {  // Base case
        draw_rect(--i);  // Recursive call
        cout << "******" << endl;
        cout << "* " << endl;
        cout << "******" << endl << endl;
    }
}
```
What is different when we call after?

- Recursive Solution

```cpp
void draw_rect(int i) {
  if(i>0){     //Base case
    cout << "******" << endl;
    cout << "*         *" << endl;
    cout << "******" << endl << endl;
    draw_rect(--i);    //Recursive call
  }
}
```