CS 162
Intro to CS II

More Big Three Concepts
Odds and Ends...

• Last week to demo Assignment #1!!!

Assignment #2

```cpp
int main() {
    game blackjack;
    int num_players;
    cin >> num_players;
    blackjack.set_players(num_players);
    blackjack.play_game();
}
```
What happens with dynamic memory?

```cpp
#include <iostream>
#include <string>
#include <cstring>
using namespace std;
const int x=20;
//x=20;
class date {
  private:
    //int month;
    char *month; //let's make month dynamic
    int year;
    int day;
    //static int i;
  public:
    const int m;
    date(char *, int, int); //constructor
    date(); //default constructor, no args.
    date(const date &); //copy constructor, arg is of same type
    ~date(); //destructor
    void operator =(const date &); //mutator
    void set_month(const int);
    void set_month(const char *);
```

```
assign up overload
d=(d1); d=d1
```
What happens with dynamic memory?

```cpp
#include "date.h"

// non-default constructor needs to create memory for member
date::date(char *ml, int d, int y) : m(d) {
    month = new char[strlen(ml) + 1];
    strcpy(month, ml);
    day = d;
    year = y;
}

// default constructor needs to set month to NULL
date::date() : m(4) {
    // m = 20;
    month = NULL;
    day = 0;
    year = 0;
}

// destructor needs to delete memory on heap
date::~date() {
    cout << "in destructor" << endl;
    delete [] month;
}

// copy constructor, once you define, you have to copy over all info
```
#include <iostream>
#include <string>
#include <cstring>
using namespace std;

class date {
    const int m;
    int day;
    int year;
    // encapsulation lets you hide the member type from user
    // int month=4;
    // char month[10];
    // string month;
    char *month;

public:
    date(); // default constructor
    // date(const int, const int, const int); // non-default constructor
    date(const char *, const int, const int); // non-default constructor
    date(const date &); // copy constructor
    ~date(); // destructor
    void operator=(const date &); // assignment overload
    // mutator, aka setter
    void set_month(char *); // set month using string
    // accessor, aka getter
    char * get_month() const;
};
```cpp
#include "date.h"

class date {  
    month=NULL;  
    day=0;  
    year=0;  
    
public:  
    date() : m(1) {  
    }  
    date(const char *m1, const int d, const int y) : m(d) {  
        month=new char[strlen(m1)+1];  
        strcpy(month, m1);  
        //month=m1;  
        day=d;  
        year=y;  
    }  
    date(const date &other) : m(other.day) {  
        cout << "in the copy constructor" << endl;  
        //month=other.month; //shallow copy  
        month=new char[strlen(other.month)+1];  
        strcpy(month, other.month);  
        day=other.day;  
        year=other.year;  
    }  
    ~date() {  
        cout << "in the destructor" << endl;  
        delete [] month; //delete memory when obj goes out of scope  
    }

    -- INSERT --
```

```cpp
20    year=other.year;
21 }
22 date::~date() {
23    cout << "in the destructor" << endl;
24    delete [] month; //delete memory when obj goes out of scope
25 }
26 //what needs to be in the assignment overload? Does it look just
27 //like the copy constructor? Do you need to delete memory?
28 void date::operator =(const date &other) {
29    cout << "in the operator =" << endl;
30    month=other.month;
31    day=other.day;
32    year=other.year;
33 }
34 //You need to delete before you reset the month because month
35 //could be pointing to something
36 void date::set_month(char *ml) {
37    //month=m; //don't do this!
38    delete [] month;
39    month=new char[strlen(ml)+1];
40    strcpy(month, ml);
41 char * date::get_month() const {
42    return month;
43 }
```

```
```cpp
#include "date.h"

//copy constructor called to create d2
void print(date d2) {
    //use accessor to get month
    cout << d2.get_month() << endl;
    d2.set_month("jan");
    //destructor called for d2
}

int main() {
    date d("april", 30, 2016);

    print(d); //copy constructor called to pass d object by value
    //d.set_month("april");

    //use accessor to get month
    cout << d.get_month() << endl;
    return 0; //destructor called for d and d1
}
```