CS 161
Intro to CS I
Continue Arrays
Odds and Ends...

- Demo Assignment 4
- Assignment 5 posted
Static vs. Dynamic 1-D arrays...

constant int array[3]

int * array = new int[3];
delete [] array;

Allocation

Stack

Heap
How does freeing memory work?

```c
int *p, *q;
p = new int;
q = new int[5];
delete p; p = NULL;
delete [] q;
```
```cpp
#include <iostream>
using namespace std;

int main() {
    int n=5;
    //int a[5]={1,2,3,4,5}; //creates an array on stack
    int a[n]; //creates a variable length array on stack, not heap
    //int **p=&a; //can't do this, &a is int *[] not int **
    int *p; //make pointer on the stack
    p=new int[n]; //allocate new space for n ints on the heap
    cout << "Array pointer's address: " << &p << endl;
    cout << "Array pointer's contents: " << p << endl;
    cout << "Array element address: " << &(p[0]) << endl;
    cout << "Array element contents: " << (p[0]) << endl;
    cout << "Array element 2 address: " << &(p[1]) << endl;
    delete [] p; //always delete before you point to new place
    n=7;
p=new int[n]; //we grew it to 7 instead of 5 by deleting and recreating
    cout << "Array pointer's address: " << &a << endl;
    cout << "Array pointer's contents: " << a << endl;
    cout << "Array element address: " << &(a[0]) << endl;
    cout << "Array element contents: " << (a[0]) << endl;
    cout << "Array element 2 address: " << &(a[1]) << endl;
    //delete [] a; //cannot free/delete memory off stack
    return 0;
}
```