CS 161
Intro to CS I

Continue Arrays
Odds and Ends...

• Assignment 5 design due Sunday by 11:59pm on Canvas!

• Questions?

Can make better after this day/time, but to get critiques.
```cpp
#include <iostream>

using namespace std;

int main () {
    //int a[1000000000]; //can't put a billion ints on the stack'
    int *a; //make pointer to capture address of where array is on heap

    //can put lots of billions of ints on heap
    while(1) {
        a = new int[1000000000];
        cout << *a << endl;
    }

    return 0;
}
```
What are the similarities/differences?

• **String Object vs. C String**
  – Which library to include?
    `<string>` **VS.** `<string.h>` or `<cstring>`
  – How do we create it?
    `string str_obj;` **VS.** `char str_arr[20];`
  – How do we access it?
    `str_obj.at(3)` or `str_obj[3]` **VS.** `str_arr[3]` or `*(str_arr+3)`
  – How do we get the length?
    `str_obj.size()` or `str_obj.length()` **VS.** `strlen(str_arr)`
  – How is length of string determined?
    Size member variable **VS.** `\0`, null character at end
```cpp
#include <iostream>

using namespace std;

int main () {
    char str[20];

    cout << "Enter 19 chars: ";
    cin >> str; //null gets put into the char array, so I take by that

    cout << "String beginning address: " << (void *)str << endl;
    cout << "String contents: " << str << endl; //see contents of string

    return 0;
}
```
```cpp
#include <iostream>
#include <cstring> //string.h

using namespace std;

int main() {
    //char str[20]; //statically allocated can only hold 19 chars entered
    //instead of limiting the number of chars entered by the user, we can
    //grow our array to hold the right amount of chars
    char s; //read a char at a time
    char *str=new char[1]; //create smallest c-string, which is just null char '\0'
    char *temp; //need to capture old string in pointer
    //add '\0' to smallest c-string with length zero, but with one element
    str[0]=\0; cout << "Length: " << strlen(str) << endl;
    //read one character from input and add to string if it is not the newline/enter
    cin.get(s);
    while(s!='\n'){
        temp=str; //point to old string, so we don't lose it
        str=new char[strlen(temp)+2]; //add space to the array, strlen doesn't count null char + extra
        strcpy(str,temp); //copy old string into new string (destination, source)
        str[strlen(temp)]=s; //add new character where null was in the old string
        str[strlen(temp)+1]='\0'; //add null character to the last element in array
        delete [] temp; //get rid of old array, so we don't have memory leak
        cin.get(s); //get another character to make sure it isn't '\n' to stop reading input
    }

    return 0;
}
```