CS 161
Intro to CS I
Static vs. Dynamic Arrays
Odds and Ends...

• Assignment #4 due tonight
• Assignment #5 posted

3. read characters to the exact length grow the string
   no mem leaks

C strings / no C++ strings
soon by 5pm!!!

C strings

Why reinvent the wheel? 3 no C string function
no cstring/strings.h

1. C, strings
   pre define how big & limit
   the user to that # of characters

2. ask the user how many chars
   + create array!
#include <iostream>

using namespace std;

#define NUM_ELEM 4 // make a macro or constant for number of elements

int main() {
    int a[NUM_ELEM] = {1}; // will only initialize first element to 1
    for (int i = 0; i < NUM_ELEM; i++)
        cout << a[i] << endl; // print initial values of elements

    for (int i = 0; i < NUM_ELEM; i++)
        a[i] = 2; // initialize all elements to 2

    cout << a << endl; // print address of where array begins
    cout << &a << endl; // print address of where pointer lives
    cout << &a[1] << endl; // print address of 2nd element in array

    for (int i = 0; i < NUM_ELEM; i++)
        cout << a[i] << endl; // print contents of all elements

    return 0;
}
Static vs. Dynamic 1-D arrays...

- **Compile-time defined on stack**
- **Int array[3]**
- **Constant size, self-referential**

Dynamic runtime:
- **Int* array**
- **Array = new int[Num-Players]**
- **Array points to heap memory**

Memory layout:
- **Stack**
  - 0 array[0]
  - 32 array[1]
  - 64 array[2]

- **Heap**
  - 256 array[0]
  - array[1]
  - array[2]
How does freeing memory work?

```c
int *p, *q;
p=new int;
q=new int[5];
delete p;
delete [] q;
```
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>
#include <string.h>  // or cstring
using namespace std;
#define NUM 5

int main() {
    // char str[NUM];  // create a static array
    char *str;  // create a dynamic array
    int num;

    while(1){
        cout << "how many chars: ";
        cin >> num;
        str = new char[num];  // on the heap

        cout << "enter a message: ";
        cin >> str;  // automatically put '\0' on end

        cout << &str << endl;  // pointer is on stack
        cout << str << endl;  // should be address, but cout gets smart
        cout << (void *)str << endl;  // don't get smart and give address
        cout << strlen(str) << endl;  // length of str based on where null is

        delete [] str;  // delete array off heap for no mem leak
        cout << str << endl;  // str contents and array not cleared on delete
}
    return 0;
}
```
Passing a 1-D Array (Static/Dynamic)

```c
int main() {
    int array[5];
    ...
    pass_1darray(array);
    ...
}
void pass_1darray(int *a) {
    cout << “Array at zero: ” << a[0] << endl;
}
OR
void pass_1darray(int a[]) {
    cout << “Array at zero: ” << a[0] << endl;
}
```