CS 161
Intro to CS I

Static/Dynamic Arrays
Odds and Ends...

• Test next Monday, 11/21
• Keep working on Assignment 5
• Get Assignment 4 demoed
• Critiques due by end of week
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;
}
```
How does freeing memory work?

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

int main() {
    string *b; //static array on stack
    int num;
    cout << "enter number of elements: ";
    cin >> num;

    b=new string[num]; //dynamic array on the heap

    //fill b
    for (int i=0; i<num; i++)
        b[i]="hi";

    delete [] b; //delete the array I point to

    return 0;
}

What are the similarities/differences?

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

using namespace std;

int main()
{
    char str[256];

    cout << "give string: " << endl;
    cin >> str; //puts '\0' on end of string

    cout << strlen(str) << endl; //get length of string
    cout << str << endl; //see the whole string
    cout << (int *)str << endl; //see the address of where string is

    return 0;
}
```
Revisit Creating Memory in Functions

Advantages to Dynamic Memory
int *i=NULL; //created in main function

create_mem(&i); //call in main void
void create_mem(int **m) {
    *m = new int[5];
}
OR
i = create_mem(); //call in main
int * create_mem() {
    return new int [5];
}
```cpp
#include <iostream>
#include <string.h>

using namespace std;

void create_1d(char **s, int l)
{
    *s = new char[l];
}

int main(int argc, char *argv[])
{
    char str[10];
    char *str2;
    int len;

    cout << "num args: " << argc << endl;
    cout << argv[0] << endl;
    cout << argv[0][0] << endl;
    if(argc > 1)
        cout << argv[1] << endl;

    cout << "how many chars to enter: ";
    cin >> len;
    create_1d(&str2, len);
    cout << "give str2 string: " << endl;
    cin >> str2;
    cout << "str2 len: " << strlen(str2) << endl;
    delete [] str2;
}
int main(int argc, char *argv[])
{
    char str[10];
    char *str2;
    int len;

    cout << "num args: " << argc << endl;
    cout << argv[0] << endl;
    cout << argv[0][0] << endl;
    if(argc > 1)
        cout << argv[1] << endl;

    cout << "how many chars to enter: ";
    cin >> len;
    create_id(&str2, len);
    cout << "give str2 string: " << endl;
    cin >> str2;
    cout << "str2 len: " << strlen(str2) << endl;
    delete [] str2;

    cout << "give string: " << endl;
    cin >> str; //puts '\0' on end of string
    cout << strlen(str) << endl; //get length of string
    cout << str << endl; //see the whole string
    cout << (int *)str << endl; //see the address of where string is
    return 0;
}