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

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

• Questions?
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
Passing a 1-D Array (Static/Dynamic)

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;
}
Creating Memory in Functions

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

create_mem(&i);       //call in main void
create_mem(int **m) {
   *m = new int[4];
}
OR
i = create_mem();    //call in main
int * create_mem() {
   return new int[4];
}
What About Memory Leaks?

- What happens here...

...  

```c
int main () {
    int *i=NULL; /*created in main function
    while(1) {
        i = create_mem(); /*call in main
    }
}
```

```c
int * create_mem() {
    return new int[4];
}
```
Fixing Memory Leaks...

• What happens here...

...  
int main () {
    int *i=NULL; //created in main function
    while(1) {
        i = create_mem(); //call in main
        delete [] i; //free memory that i points to, preventing mem leaks
    }
}

int* create_mem(){
    return new int[4];
}
Demo...
In-class Exercise

• Get into groups of 4-5.
• Write a function that takes two arrays, x and y, stores the sum of each element from x and y in a new array, z, and returns the new array z.
  – What information does your function need?
  – What does the code inside the function look like?
  – How are you going to call this function?
Demo...
Multidimensional Arrays

- data_type array_name[rows][cols];
  - int array[2][3];
  - int array[4][2][3];
  - int array[2][4][2][3];

- What are examples of these?
  - 2-D – Matrices, Spreadsheet, Minesweeper, Battleship, etc.
  - 3-D – Multiple Spreadsheets, (x, y, z) system
  - 4-D – (x, y, z, time) system
Initializing 2-D Arrays

• **Declaration:** int array[2][3] = {{0,0,0},{0,0,0}};

• **Individual elements:** array[0][0]=0; array[0][1]=0; array[0][2]=0; array[1][0]=0; array[1][1]=0; array[1][2]=0;

• **Loop:**
  
  for(i = 0; i < 2; i++)
    for(j = 0; j < 3; j++)
      array[i][j]=0;

• **Why do we need multiple brackets?**
Reading/Printing 2-D Arrays

• Reading Array Values
  
  for(i = 0; i < 2; i++)
  
  for(j = 0; j < 3; j++) {
    cout << “Enter a value for ” << i << “, ” << j << “: ”;
    cin >> array[i][j];
  }

• Printing Array Values
  
  for(i = 0; i < 2; i++)
  
  for(j = 0; j < 3; j++)
    cout << “Array: ” << array[i][j] << endl;
Demo...