CS 161
Intro to CS I
Finish Dynamic Memory and 1-d Arrays
What is an Array?

• **Array (ar·ray) n.** An ordered arrangement of related items.
  
  — Example: Array of colors in a rainbow.
    • Related items?
    • Ordered arrangement?
  
  — Class examples?

  — Computer Science
    • Same data type/data structure
    • Contiguous memory locations
int student_grades[5];

- How do you access each item?
- What does the array name represent?
- Why is the array name the address of 1st element?
- What are the initial values?
Initialize/Assign Values

- **Declaration**
  
  ```c
  int student_grades[5] = {0, 0, 0, 0, 0};
  ```

- **Individual Elements**
  
  ```c
  student_grades[0]=0;
  ...
  student_grades[4]=0;
  ```

- **Why is this incorrect?**
  
  ```c
  student_grades={0, 0, 0, 0, 0};
  ```
Initialize/Assign Values...

• Using a Loop
  While Loop Example:
  i=0;
  while (i<5) {
    student_grades[i]=0;
    i++;
  }

  For Loop Example:
  for(i=0; i<5; i++)
    student_grades[i]=0;

• Which is better to use with arrays and why?
Read/Print 1-D Array Values

• Read Values From User
  
  ```
  for(i=0; i<5; i++)  {
      cout << "Enter final grade for student: ";
      cin >> student_grades[i];
  }
  ```

• Print Values
  
  ```
  for (i=0; i<5; i++) {
      cout << "Student\'s final grade is " << student_grades[i] << endl;
  }
  ```
Static vs. Dynamic 1-D arrays...
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?

```cpp
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