CS 161
Intro to CS I

Static vs. Dynamic Arrays
Odds and Ends...

• Exercise #6 due tonight
• Assignment #4 due Monday night

Hint: 2 strings
string str1, str2;
getline(cin, str1);
//copy C++ strings
str2 = str1;

for all non-space characters put a ‘-‘
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

Why pointers?
1. change a variable/value inside the function
2. access heap
3. cache efficient

lists vs.

C/C++
Create 1-D Array

```c
int student_grades[5];
```

- How do you access each item?
- What does the array name represent?
- Why is the array name the address of 1\textsuperscript{st} element?
- What are the initial values?

- How many of the type to create declared?
- Address \\
  \begin{align*}
  \text{address} &= \text{student\_grades[0]} \\
  &= (\text{student\_grades + 0})
  \end{align*}

- Brackets where the array begins in memory.
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
  
  ```cpp
  for(i=0; i<5; i++)  {
      cout << "Enter final grade for student: ";
      cin >> student_grades[i];
  }
  ```

• Print Values
  
  ```cpp
  for (i=0; i<5; i++)  {
      cout << "Student\'s final grade is " << student_grades[i] << endl;
  }
  ```
```cpp
#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;
}
```

Demo

OSU Oregon State University
Static vs. Dynamic 1-D arrays...

- Compile-time: on stack
- Int array[3]
- Constant, self-referenced
- Stack

- You cannot change where array points

- Dynamic runtime:
- int* array
- array = new int[NumPlayers]
How does freeing memory work?

```c
int *p, *q;
p = new int;
q = new int[5];
delete p;
delete [] q;
```