

Arrays

Last Time

- We
 - Learnt about passing arguments by value and by reference
- Today we will
 - Start looking at arrays

Arrays

- Arrays are a way to store more than one value for a variable
- Example: Store 10 grades
 - Using single variables--need 10 variables
 - Using array--need one array of size 10
- Declaration:
 - `float grade[10];`
 - `string name[2];`

Using Arrays

- Declaration
`double temp[5];`
- Assign
`temp[0] = 32.0;`
`temp[1] = 55.3;`
`temp[2] = 72.1;`
`temp[3] = 85.0;`
`temp[4] = 20.0;`

`for (int i = 0; i < 5; i++)`
`temp[i] = 0.0;`

`float temp[] = {32.0, 55.3, 72.1, 85.0, 20.0};`

Memory Storage

- Single variable--one memory location. Size?

- Array--multiple memory locations. Size?

Accessing Arrays

- After initialized, can read array

- Make sure you don't try to access array outside of bounds

- Indices should be 0 through size -1

General Form

datatype identifier[size];

- Same as regular variable declaration, but need to give size of array
- Size must be constant integer expression greater than zero
- Index array through []. Indices must be integers (or integer expressions) in the range 0 through size - 1
- There is no array[size] element!!

11/4/05

CS150 Introduction to Computer Science 1

7

19.1 What is the Output?

```
const int SIZE = 5;
int i, j, k;
int vals[size] = {10, 9, 8, 7};
char grades[] = {'A', 'B', 'C', 'D', 'F'};

i = 2;
j = 3;
k = 7;

cout << vals[2] << vals[j] << vals[j-i];
cout << grades[j] << grades[j % i] << grades[i % j] <<
    grades[k / i];
cout << int (grades[i]) << grades[vals[1] % 4];
cout << grades[vals[1]];
```

11/4/05

CS150 Introduction to Computer Science 1

8

Problems

- 19.2 Write a C++ program segment that will switch the values in the first and third elements of the array values
- 19.3 Write a C++ program segment that will declare an integer array nums of size 100. Then place the first 100 positive even integers into the array nums starting with the first element and proceeding to the end of the array
- 19.4 A data file of grades exists with an unknown number of characters. Write a C++ program segment that will read these characters into a character array. Assume no more than 1000 characters exists in the data file

11/4/05

CS150 Introduction to Computer Science 1

9

Passing by Reference

```
int i, j;
int arry1[] = {1,2,3,4,5};
int arry2[] = {5,4,3,2,1};

i = 2;
j = 3;
...

void swap (int& num1, int& num2)
{
    int temp;

    temp = num1;
    num1 = num2;
    num2 = temp;
}
```

• 19.5 What happens?

a) swap(i, arry1[1]);

b) swap(arry1[2], arry1[3]);

c) swap(arry1[i], arry2[i+1]);

d) swap(i, arry1[1]+i);

e) swap(arry1, arry2);

11/4/05

CS150 Introduction to Computer Science 1

10

Passing arrays to functions

- Can pass individual elements
- If pass whole array, it is automatically passed by reference. Why do you think that is?
- The address of the array is passed to the function so that any element of the array can be accessed
- The address of an array is the memory location of the first element

11/4/05

CS150 Introduction to Computer Science 1

11

19.6 Problem

- Write the function definition for a 'large' function that stores the larger of each element in arry1 and arry2 in arry3. If the program works, arry3 should have {5,4,3,4,5}.

11/4/05

CS150 Introduction to Computer Science 1

12

Solution

```
void large (int size, int arry1[], int
  arry2[], int arry3[]);

void main()
{
  const int size = 5;
  int arry1[] = {1,2,3,4,5};
  int arry2[] = {5,4,3,2,1};
  int arry3[size];

  large (size, arry1, arry2, arry3);
}
11/4/05 CS150 Introduction to Computer Science 1 13
```

Arrays used as input

- What happens when want to use array only as input? We can't pass it by value...
 - void large (int size, const int arry1[], const int arry2[], int arry3[]);
- We can protect array arguments by putting const in front of them in prototype and function definition

11/4/05 CS150 Introduction to Computer Science 1 14

19.7 Problem

- Assume you have two arrays of floats called vals1 and vals2. They both contain maxels elements. Write a C++ bool function identical that will accept both arrays and return true if both arrays are identical; otherwise, return false. The call to your function might be by a statement of the following form:

```
if (identical (maxels, vals1, vals2))
  cout << "Arrays are identical" << endl;
else
  cout << "Arrays are not identical" << endl;
```

11/4/05 CS150 Introduction to Computer Science 1 15

Summary

- In today's lecture we covered
 - Arrays
- Readings
 - P. 170 - 180

11/4/05 CS150 Introduction to Computer Science 1 16