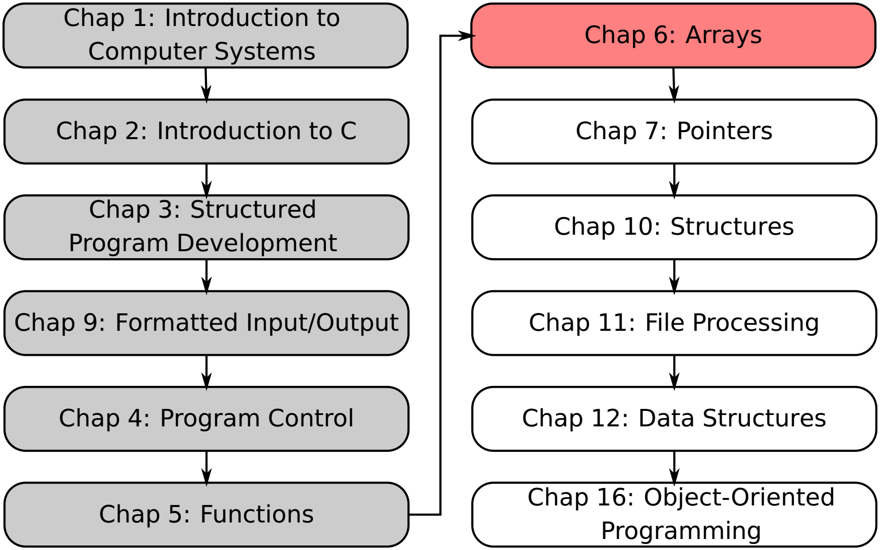
Computer Programming 143 – Lecture 14
Arrays I

Electrical and Electronic Engineering Department
University of Stellenbosch

Prof Johan du Preez
Mr Callen Fisher
Dr Willem Jordaan
Dr Hannes Pretorius
Mr Willem Smit

Copyright & Disclaimer

Module Overview



Chap 1: Introduction to Computer Systems
Chap 2: Introduction to C
Chap 3: Structured Program Development
Chap 9: Formatted Input/Output
Chap 4: Program Control
Chap 5: Functions
Chap 6: Arrays
Chap 7: Pointers
Chap 10: Structures
Chap 11: File Processing
Chap 12: Data Structures
Chap 16: Object-Oriented Programming

Lecture Overview

1. Introduction (6.1)
2. Arrays (6.2)
3. Declaring Arrays (6.3)
4. Examples Using Arrays (6.4)

6.1 Introduction

Definition of array

- Structure that stores related data items of the same data type
- Fixed size throughout program execution
  - Dynamic data structures discussed in Chapter 12

6.2 Arrays I

Array

- Group of consecutive memory locations
- Same name and type

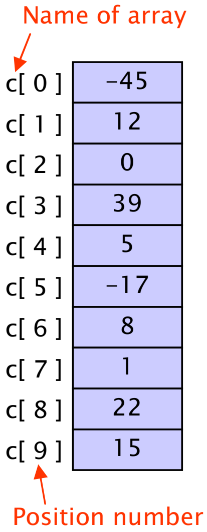
To refer to a specific element

- Array name
- Position number (index)

Format:

name[position number];

- First element is at position 0
- **n** element array named **c**:
  - **`c[0], c[1] ... c[n-1]`**



6.2 Arrays II

Array elements are like normal variables

- Assign a value of 3 to the first element of array **c**:

```
c[ 0 ] =  3;
```

- Display the sum of the first 3 elements of **c**:

```
printf( "%d", c[ 0 ] + c[ 1 ] + c[ 2 ] );
```

- Halve the 7th element of **c** and store the result in variable **x**:

```
x = c[ 6 ] / 2;
```

- Can perform operations in index. If **a = 3** and **b = 2**, then

```
c[ a + b ] += 2;
```

  would increment the 6th element of **c** by 2

6.3 Declaring Arrays

When declaring arrays, specify

- Type of array
- Name
- Number of elements

  arrayType arrayName[numberOfElements];

  Examples:

```
int c[ 10 ];
float myArray[ 3284 ];
```

Declaring multiple arrays of same type

- Format similar to regular variables

  Example:

```
int b[ 100 ], x[ 27 ];
```

6.4 Examples Using Arrays I

Initialising arrays

- Like other variables, array elements are not initialised automatically at declaration
- Initialise elements by assigning values one by one (Fig. 6.3 in D&D)
- Or use an initialiser list:

```
int n[5] = { 1, 2, 3, 4, 5 };
```

- If initialiser list shorter than array – remaining elements become 0

```
int n[5] = { 0 };
```

  - All 5 elements of **n** are initialised to 0
- If size omitted, initialisers determine it

```
int n[ ] = { 1, 2, 3, 4, 5 };
```

  - 5 initialisers, therefore 5-element array

6.4 Examples Using Arrays II

Problem

- Initialize an array with the even integers from 2 to 20 and then display the array.

Pseudocode

```
for each integer j from 1 to 10 (j is loop counter)
    set array element = 2*j

for each integer j from 1 to 10
    print array element j
```

6.4 Examples Using Arrays III

```c
/* Fig. 6.5 in Deitel&Deitel : Initialising the elements of array s
to the even integers from 2 to 20 */

#include <stdio.h>
#define SIZE 10 // maximum size of array
int main( void )
{
   int s[ SIZE ];  // array has SIZE elements
   int j; // counter
   for ( j = 0; j < SIZE; j++ ) { // set the values
      s[ j ] = 2 + 2 * j;
   } // end for
   printf( "%s%13s\n", "Element", "Value" );
   for ( j = 0; j < SIZE; j++ ) {
      printf( "%7d%13d\n", j, s[ j ] );
   } // end for
   return 0; // indicates successful termination
} // end main
```

6.4 Examples Using Arrays IV

Output

```
     Element          Value
           0              2
           1              4
           2              6
           3              8
           4             10
           5             12
           6             14
           7             16
           8             18
           9             20
```

6.4 Examples Using Arrays V

Symbolic constants

```
#define SIZE 10
```

- Is a C preprocessor directive
- Defines a symbolic constant **SIZE** of value 10
- Preprocessor replaces all occurences of **SIZE** in the text with 10 before program is compiled
- Makes program scalable – value is changed in only one place in the text

6.4 Examples Using Arrays VI

Survey example: problem statement

Forty students were asked to rate the quality of the food in the student cafeteria on a scale of 1 to 10 (1 means awful and 10 means excellent). Place the 40 responses in an integer array and summarise the results of the poll.

6.4 Examples Using Arrays VII

Pseudocode

```
Declare and initialise an array called bins to keep count of each rating
   score. Each element of bins will represent a score, and will count the
   number of times the score occurs.

Declare and initialise an array called responses to capture all the
   responses

For each response in the responses array
    Determine the rating score
    Increment the corresponding element in the bins array

For each rating score (i.e. 1 to 10)
    Print the corresponding element in the bins array.
```

6.4 Examples Using Arrays VIII

```c
/* Student poll program; Modified from Fig. 6.7 in Deitel & Deitel */
#include <stdio.h>
#define RESPONSE_SIZE 40 // number of responses
#define BIN_SIZE 10 // number of bins (number of possible ratings)

int main( void ) // function main begins program execution
{
     int resp_cnt; // counter to loop through responses
     int rating_cnt; // counter to loop through ratings 1-10
     int rating; // temporary, stores rating of current response
        /* bins array to keep count of number of occurrences per rating
      * bin[0] will keep track of the number of "1" responses,
      * bin[n] will keep track of the number of "n+1" responses */
     int bins[ BIN_SIZE ] = {0}; //initialise bins (set all to 0)
     // responses hard-coded in the responses array for this example
     int responses[ RESPONSE_SIZE ] = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
            1, 6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7, 5, 6,
            6, 5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };
```

6.4 Examples Using Arrays IX

```
    /* for each response, get rating and increment appropriate bin*/
    for ( resp_cnt = 0; resp_cnt < RESPONSE_SIZE; resp_cnt++ ) {
         rating = responses[ resp_cnt ]; //get response from array
         //note:rating-1 since array starts at 0 and ratings at 1
         bins[rating-1]++; //select bin, add one to the bin
    } // end for
    // display results
    printf( "%s%12s\n", "Rating", "Occurrences" );
    // output the frequencies in a tabular format
    //note:rating_cnt + 1 since array starts at 0 and ratings at 1
    for ( rating_cnt = 0; rating_cnt < BIN_SIZE ; rating_cnt++ )
         printf( "%6d%12d\n", rating_cnt + 1, bins[ rating_cnt ]);
    return 0; // indicates successful termination
} // end main
```

6.4 Examples Using Arrays X

Output

```
    Rating Occurrences
         1           2
         2           2
         3           2
         4           2
         5           5
         6          11
         7           5
         8           7
         9           1
        10           3
```

Perspective

Today

Arrays I

- Definition of arrays
- Declaration of arrays
- Examples

Next lecture

Arrays II

- Passing arrays to functions
- Searching arrays

Homework

1. Study Sections 6.1-6.4 in Deitel & Deitel
2. Do Self Review Exercises 6.1(a)-(c), 6.2(a)-(d), 6.3, 6.5(a)-(d)
3. Do Exercises 6.8(a)-(e), 6.12 in Deitel & Deitel