EECS 10: Computational Methods in Electrical and Computer Engineering Lecture 17

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering Electrical Engineering and Computer Science University of California, Irvine

Lecture 17: Overview

- Data Structures
 - Introduction
 - Arrays
 - Introduction
 - Indexing
 - Initialization
 - · Multi-dimensional arrays
 - · Operator associativity and precedence
 - Example
 - Histogram.c

EECS10: Computational Methods in ECE, Lecture 17

(c) 2008 R. Doemer

2

Data Structures

- Introduction
 - Until now, we have used (mostly) single data elements of basic (non-composite) type
 - · integral types
 - · floating point types
 - Most programs, however, require complex data structures using composite types
 - · arrays, lists, queues, stacks
 - · trees, graphs
 - · dictionaries
 - ANSI C provides built-in support for
 - arrays
 - · structures, unions, enumerators
 - · pointers

EECS10: Computational Methods in ECE, Lecture 17

(c) 2008 R. Doemer

3

Arrays

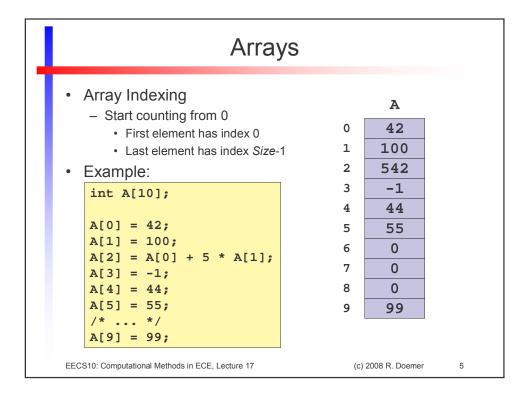
- Array data type in C
 - Composite data type
 - Type is an array of a sub-type
- (e.g. array of int)
- Fixed number of elements
 - Array size is fixed at time of definition (e.g. 100 elements)
- Element access by index (aka. subscript)
 - Element-access operator: array[index](e.g. A[42])
- Example:

```
int A[10];  /* array of ten integers */
A[0] = 42;  /* access to elements */
A[1] = 100;
A[2] = A[0] + 5 * A[1];
```

EECS10: Computational Methods in ECE, Lecture 17

(c) 2008 R. Doemer

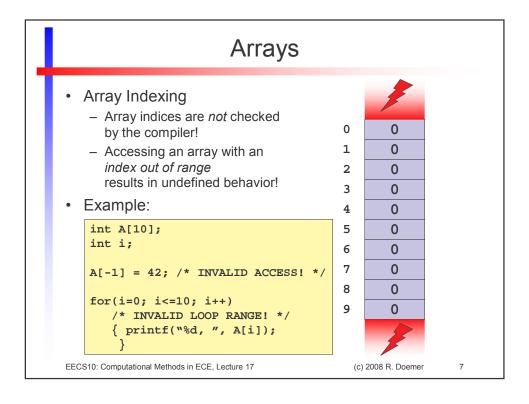
4

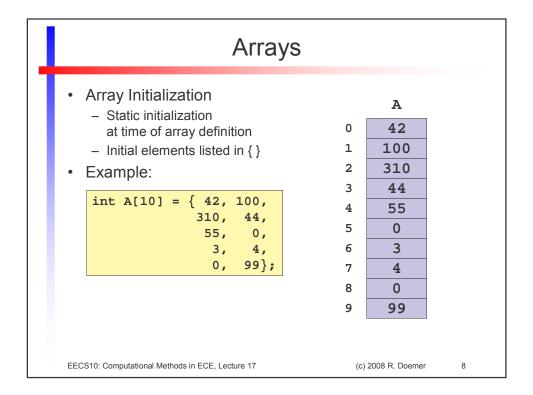


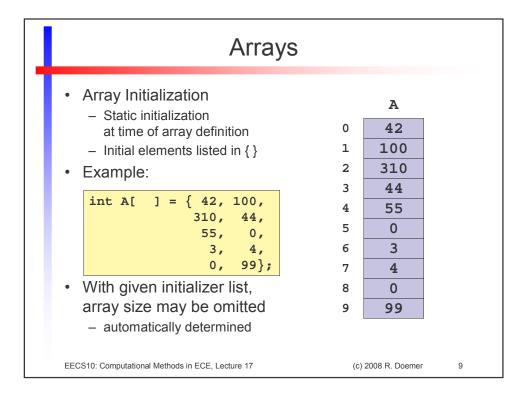
```
Arrays
  Array Indexing
                                                    Α

    for loops are often very helpful

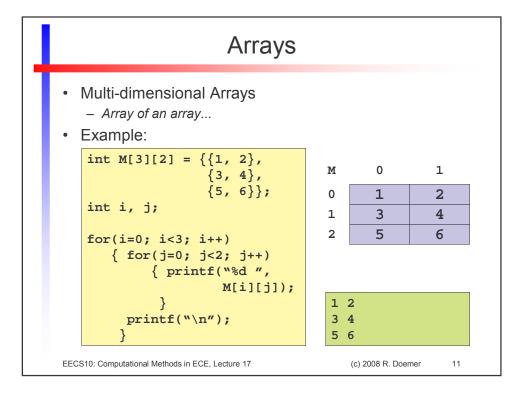
                                                     0
                                             0
        • for(i=0; i<N; i++)</pre>
             {...A[i]...}
                                                    11
                                             1
                                                    22
   Example:
                                             2
                                             3
                                                    33
    int A[10];
    int i;
                                                    44
                                             4
                                             5
                                                    55
    for(i=0; i<10; i++)
       {A[i] = i*10 + i;}
                                             6
                                                    66
                                                    77
                                             7
    for(i=0; i<10; i++)
                                                    88
                                             8
       { printf("%d, ", A[i]);
                                             9
                                                    99
   0, 11, 22, 33, 44, 55, 66, 77, 88, 99,
EECS10: Computational Methods in ECE, Lecture 17
                                               (c) 2008 R. Doemer
                                                                6
```







Arrays				
 Array Initialization Static initialization at time of array definition Initial elements listed in { } Example: 	0 1 2 3	A 1 2 3		
 int A[10] = { 1, 2, 3}; With given initializer list and array size, unlisted elements are zero-initialized array is filled up with zeros 	4 5 6 7 8 9	0 0 0 0 0 0		
EECS10: Computational Methods in ECE, Lecture 17	(c)	(c) 2008 R. Doemer 10		



```
Arrays
   Operator associativity and precedence
    - parentheses, array access
                                                             left to right
                                            (),[]

    unary operators

                                            +, -, !, ++, -- right to left
    - type casting
                                            (typename)
                                                             right to left

    multiplication, division, modulo

                                           *, /, %
                                                             left to right

    addition, subtraction

                                                             left to right
                                           +, -
    - shift left, shift right
                                                             left to right
                                           <<, >>
    - relational operators
                                                             left to right
                                           <, <=, >=, >
    equality
                                           ==, !=
                                                             left to right

    logical and

                                           &&
                                                             left to right
    - logical or
                                                             left to right
                                            - conditional operator
                                                             left to right
    - assignment operators
                                           =, +=, *=, etc. right to left

    comma operator

                                                             left to right
EECS10: Computational Methods in ECE, Lecture 17
                                                                       12
                                                    (c) 2008 R. Doemer
```

```
Arrays
• Program example: Histogram.c
   - Display a simple bar graph for 10 integer values
 Desired output:
  % Histogram
  Please enter data value 1: 111
  Please enter data value 2: 222
Please enter data value 3: 33
Please enter data value 4: 333
  Please enter data value 10: 111
   1: 111 *********
   2: 222 ******************
   3: 33 ****
   4: 333 **************************
  [...]
  10: 111 *********
  %
EECS10: Computational Methods in ECE, Lecture 17
                                             (c) 2008 R. Doemer
```

```
Arrays
  Program example: Histogram.c (part 1/3)
    /* Histogram.c: print a histogram of data values
    /* author: Rainer Doemer
   /* modifications:
   /* 11/02/04 RD initial version
   #include <stdio.h>
   /* constants */
   #define NUM_ROWS 10
   /* main function */
   int main(void)
       /* variable definitions */
      int Data[NUM_ROWS];
      int i, j, max;
      double scale;
EECS10: Computational Methods in ECE, Lecture 17
                                               (c) 2008 R. Doemer
                                                               14
```

Program example: Histogram.c (part 2/3) ... /* input section */ for(i = 0; i < NUM_ROWS; i++) { printf("Please enter data value %2d: ", i+1); scanf("%d", &Data[i]); } /* rof */ /* computation section */ max = 0; for(i = 0; i < NUM_ROWS; i++) { if (Data[i] > max) { max = Data[i];} } }

EECS10: Computational Methods in ECE, Lecture 17

} /* rof */
scale = 70.0 / max;

} /* fi */

(c) 2008 R. Doemer

15

Arrays

• Program example: Histogram.c (part 3/3)

EECS10: Computational Methods in ECE, Lecture 17

(c) 2008 R. Doemer

16

