Computer Programming 143 – Lecture 17
Arrays IV

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

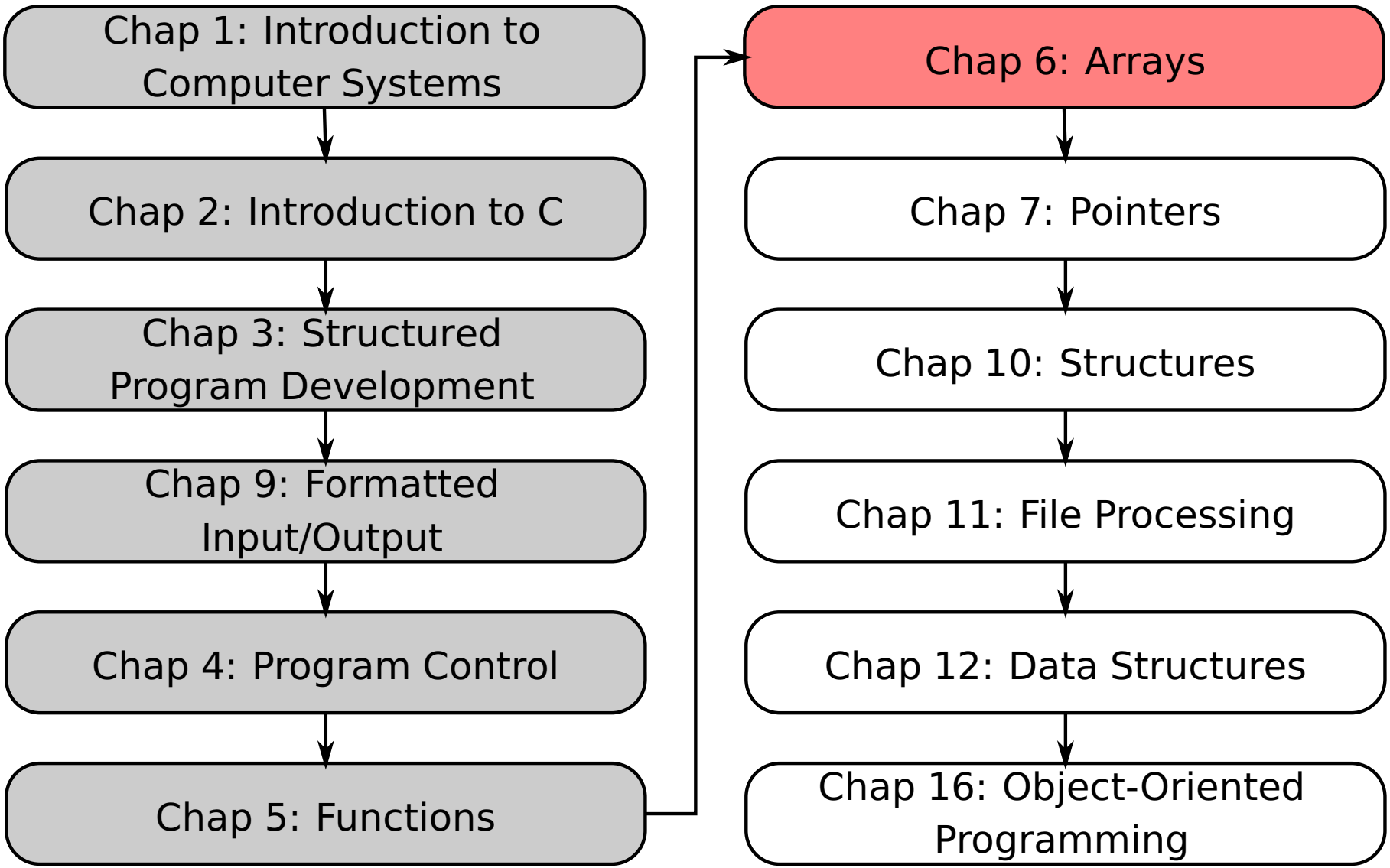
Disclaimer

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

Module Overview



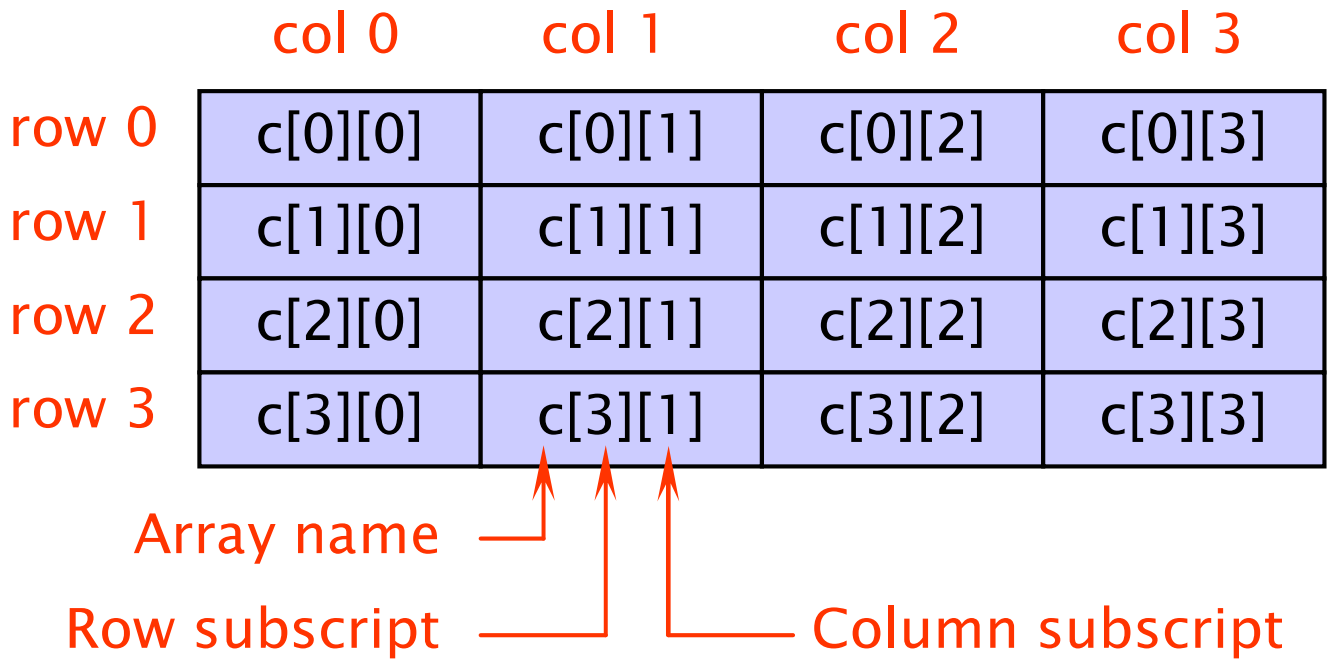
Lecture Overview

1. Multiple-Subscripted Arrays (6.11)
2. Debugging
3. A1 Test Information

6.11 Multiple-Subscripted Arrays I

Multiple-Subscripted Arrays

- Tables with rows and columns (**m** by **n** array)
- Like matrices: specify row, then column



6.11 Multiple-Subscripted Arrays II

Initialisation

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

- Initialisers grouped by row in braces
- If not enough, unspecified elements set to zero

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

1	2
3	4

1	0
3	4

Referencing elements

- Specify row, then column

```
printf( "%d", b[ 0 ][ 1 ] );
```

Refer to Fig. 6.21 and 6.22 in Deitel & Deitel for examples

Debugging

What is debugging?

- It is a systematic process of identifying and fixing errors (bugs) in a computer program
  - Errors may lead to wrong results or code that does not compile

Debugging methods

Using printf () statements

- Print out the data in a particular variable to see if it is as expected
- To see if a certain line is executed:

```
printf("%s:%d\n", __FILE__, __LINE__);
```

- To interrupt the program at specific point:

```
exit(-1);
```

Use *.....*\ comment blocks

- Block out large sections that may be leading to an error
- Unblock single statements / small sections until the error occurs

Best practice - use the debugger

- Those red arrows etc at the right top in Code::Blocks
- Install if not available - see SUNLearn prac 00 under Week 1

A1 Test Information I

Format of A1 test

- 2 hours, 60 marks
- Quiz format on SUNLearn

A1 Test Information II

Example question formats

Similar to that asked in practical tests

- Write C code that would give the following output...
- Complete the C code...
- Write C code that would solve the following problem...
- Use the ... selection/repetition structure to do...
- Write C code that would implement the following flow diagram/pseudocode...
- Identify the errors in following code...
- Name three of the six phases a C program typically goes through to be executed.

A1 Test Information III

Resources

- Textbook
- Weekly video lectures (SUNLearn)
- Weekly lecture slides and example programs (SUNLearn)
- Homework problems (in lecture slides)
- Weekly practical programming exercises and memos (SUNLearn)

A1 Test Information IV

List of topics covered

Refer to the class notes for a complete list

- Chapter 1: Introduction to computers - *hardware, software, computer organisation, computer languages, C standard library, structured programming, object technology, C development environment*
- Chapter 2: Introduction to C programming - *program structure, input ('scanf') and output ('printf'), variables, arithmetic, relational operators*
- Chapter 3: Structured programming development - *algorithms, pseudocode, flow diagrams, control structures, 'if', 'if...else' and 'while' statements, algorithm design using top-down, step-wise refinement, assignment, increment and decrement operators*
- Chapter 9: Formatted input/output - *'printf', 'scanf', precision, field width*

A1 Test Information V

List of topics covered (cont...)

Refer to the class notes for a complete list

- Chapter 4: Program control - *counter- and sentinel-controlled repetition, 'for', 'do...while' and 'switch' statements, logical operators, 'break'*
- Chapter 5: Functions - *components of a function, function calls, math library functions, function definitions and prototypes, casting, header files, passing arguments by value and by reference, random number generation, storage classes, scope rules, recursive functions*
- Chapter 6: Arrays - *declaration, initialisation and use of arrays, strings, symbolic constants, passing arrays to functions, linear and binary search of arrays, sorting arrays (bubble sort), multiple-subscripted arrays*

Perspective

Today

- Multiple-subscripted arrays
- Debugging

Next

- A1

Homework

1. Study Section 6.11 in Deitel & Deitel
2. Do Self Review Exercise 6.4 in Deitel & Deitel
3. Prepare for A1