Computer Programming 143 – Lecture 26
File Processing II

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

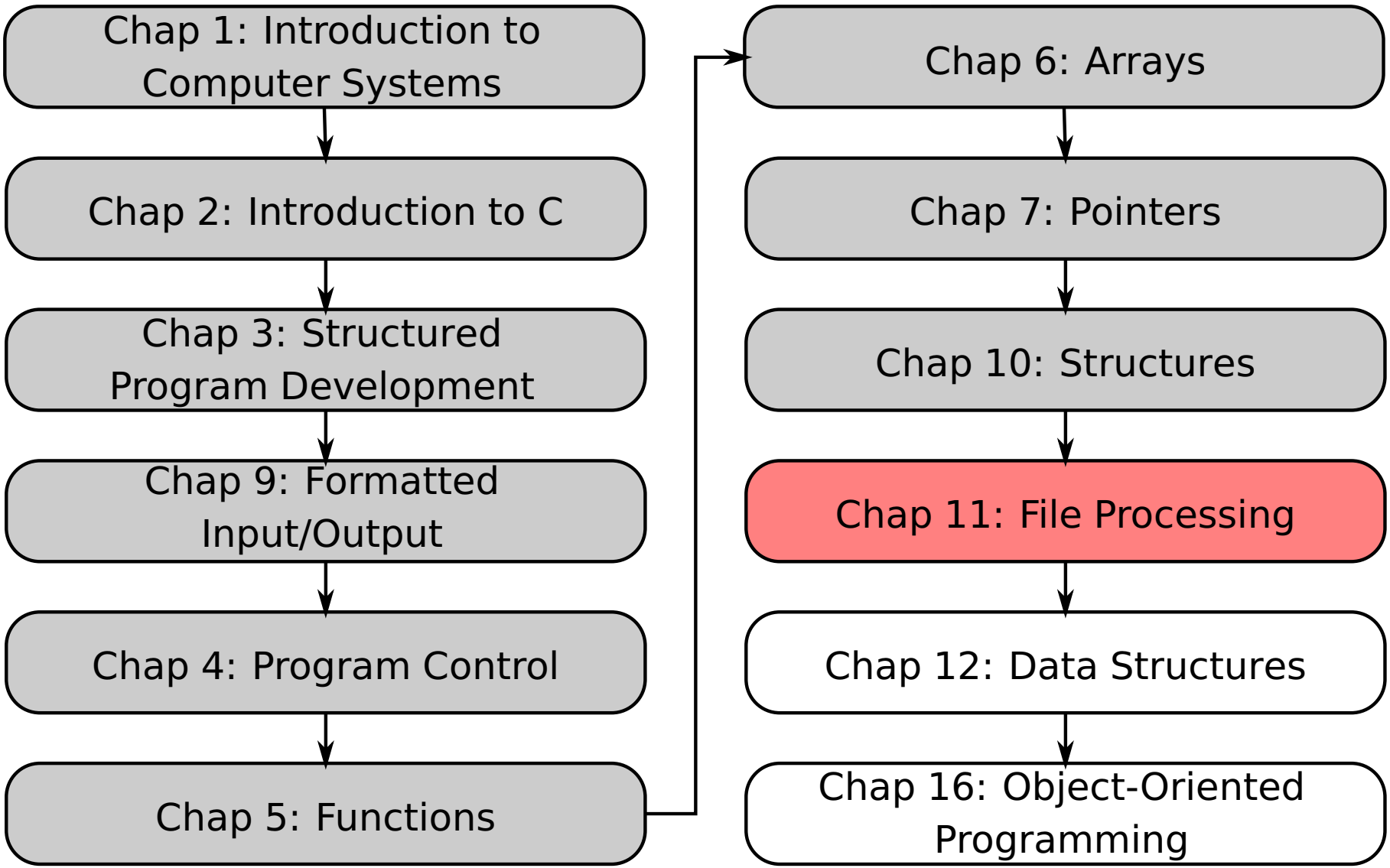
Disclaimer

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

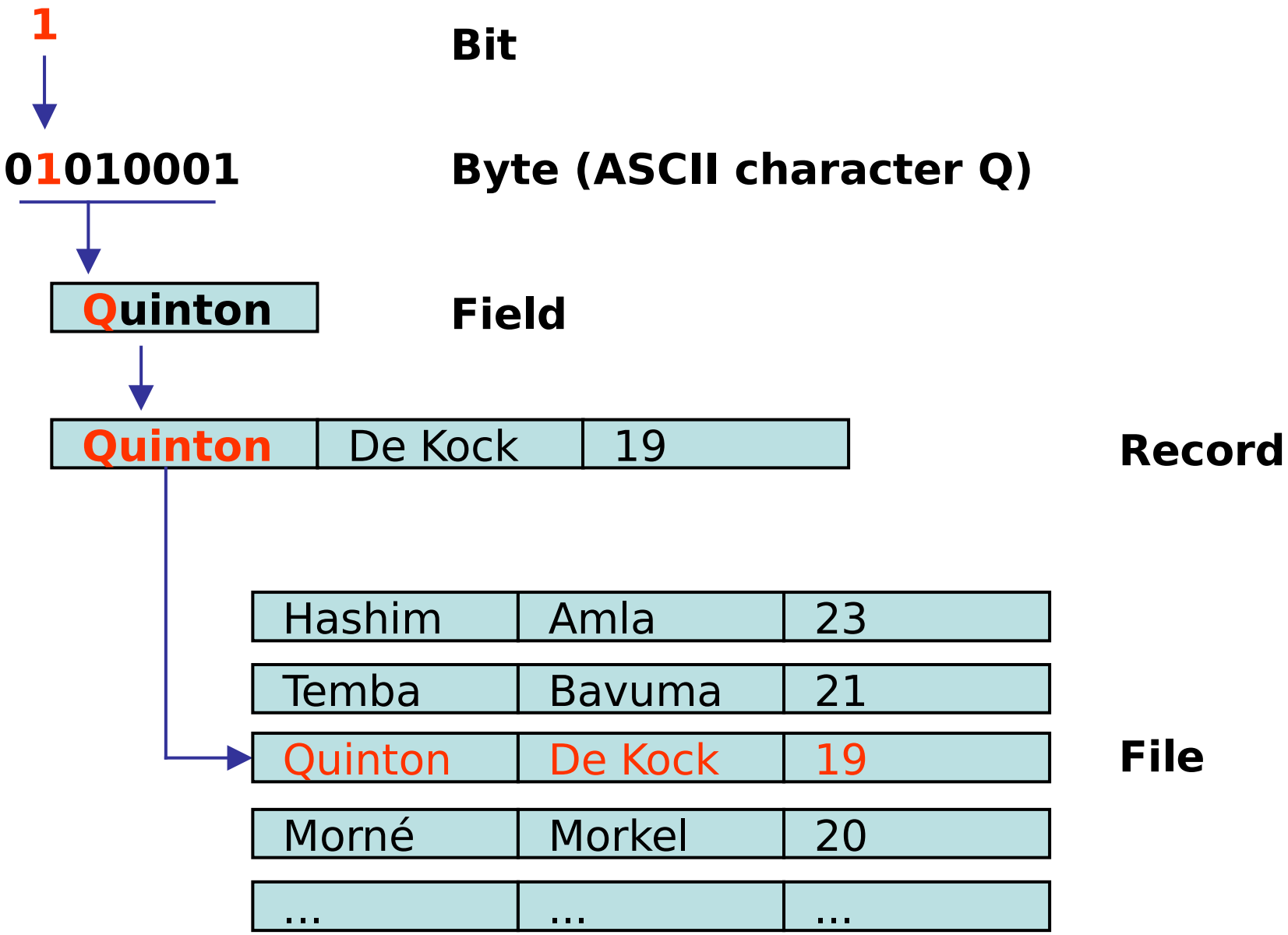
1. The Data Hierarchy (1.3)
2. Write to/Read from Binary File (11.6-11.8)

1.3 The Data Hierarchy I

Data Hierarchy

- Bit – smallest data item
  - Value of 0 or 1
- Byte – 8 bits
  - Used to store a character
    - Decimal digits, letters, and special symbols
- Field – group of characters conveying meaning
  - Example: your name
- Record – group of related fields
  - Represented by a `struct` or a **`class`**
  - Example: In a payroll system, a record for a particular employee that contained his/her identification number, name, address, etc.
- File – group of related records
  - Example: payroll file
- Database – group of related files

1.3 The Data Hierarchy II



1.3 The Data Hierarchy III

Data files

- Record key
  - Identifies a record to facilitate the retrieval of specific records from a file (eg. ID)

ID	NAME	SURNAME
253	Hashim	Amla
254	Temba	Bavuma
255	Quinton	De Kock
256	Morné	Morkel
257	...	...

- Sequential file
  - Records typically sorted by key

11.6 Write to/Read from Binary File I

Unformatted I/O functions

- `fwrite`
  - Transfer bytes from a location in memory to a file
- `fread`
  - Transfer bytes from a file to a location in memory
- Example:

```
fwrite( &number, sizeof( int ), 1, fPtr );
```

  - `&number` – Location to transfer bytes from
  - `sizeof( int )` – Number of bytes to transfer
  - 1 – For arrays, number of elements to transfer. In this case, “one element” of an array is being transferred
  - `fPtr` – File to transfer to or from

11.6 Write to/Read from Binary File II

Writing structures

```
fwrite( &myObj, sizeof( struct myStrct ), 1, fPtr );
```

- `sizeof` – returns size in bytes of object in parentheses

Reading structures

```
fread( &myObj, sizeof( struct myStrct ), 1, fPtr );
```

- `sizeof` – returns size in bytes of object in parentheses

Writing and reading array of structures

```
fwrite( &myObj, sizeof( struct myStrct ), numElements, fPtr );
fread( &myObj, sizeof( struct myStrct ), numElements, fPtr );
```

- `numElements` – is the number of array elements

11.7 Sequential Write to Binary File I

```c
/* Example of writing sequentially to a binary file */
#include <stdio.h>
#include <stdlib.h>

struct coord { // structure to store coordinates
   double x;
   double y;
}; // end structure coord
typedef struct coord Coord;

int main( void )
{
   char filename[ 20 ]; // string to store filename
   FILE *cfPtr;         // file pointer
   int i;               // counter
   Coord vertex;        // coordinates of a vertex of triangle

   printf( "Enter file name: " );
```

11.7 Sequential Write to Binary File II

```
   scanf( "%19s", filename ); // reads filename from user
   cfPtr = fopen( filename, "wb" ); // opens binary file for writing
   if ( cfPtr == NULL ) { // exits if file cannot be opened
      printf( "ERROR: File could not be opened!\n" );
   }
   else {
      printf( "\nPlease specify the vertices of the triangle\n" );
      for ( i = 1; i <= 3; i++ ) {
         printf( "Enter the coordinates of vertex %d: ", i );
         // reads a vertex from user
         scanf( "%lf%lf", &vertex.x, &vertex.y );
         // writes vertex to file
         fwrite( &vertex, sizeof( Coord ), 1, cfPtr );
      }
      fclose( cfPtr ); // closes file
   }
   return 0; // indicates successful termination
} // end main
```

11.8 Sequential Read from Binary File I

```c
/* Example of reading sequentially from a binary file */
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

struct coord { // structure to store coordinates
  double x;
  double y;
}; // end structure coord
typedef struct coord Coord;

int main( void ) {
  char filename[20]; // string to store filename
  FILE *cfPtr;       // file pointer
  int i;             // counter
  Coord vertices[3]; // array for triangle coordinates
```

11.8 Sequential Read from Binary File II

```
  printf( "Please enter filename: " );
  scanf( "%s", filename ); // reads filename from user
  cfPtr = fopen( filename, "rb" ); // opens file for reading
  if ( cfPtr == NULL ) { // exits if file cannot be opened
  printf( "ERROR - file could not be opened!\n" );
  }
  else {
    printf( "\n" );
    for ( i = 0; i < 3; i++ ) {
      // reads one set or coordinates
      fread( &vertices[i], sizeof( Coord ), 1, cfPtr );
      printf( "Vertex %d: ( %.1lf, %.1lf )\n", i + 1,
      vertices[i].x, vertices[i].y );
    }
    fclose( cfPtr ); // closes file
  }
  return 0; // indicates successful termination
} // end main
```

Perspective

Today

File Processing II

- Data hierarchy
- Binary files

Next lecture

File Processing III

- Random-access files

Homework

1. Study Sections 1.3, 11.5-11.8 in Deitel & Deitel
2. Do Self Review Exercises 11.1, 11.2, 11.4 in Deitel & Deitel
3. Do Exercises 11.5, 11.16 in Deitel & Deitel