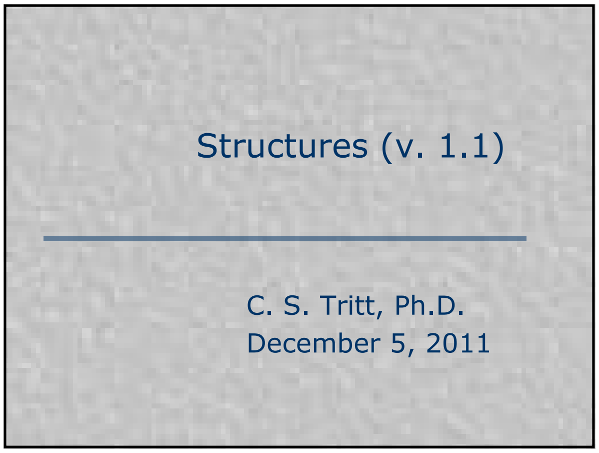
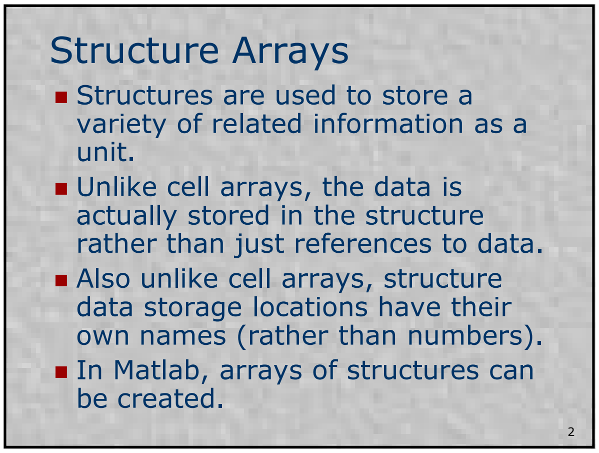


Structures (v. 1.1)

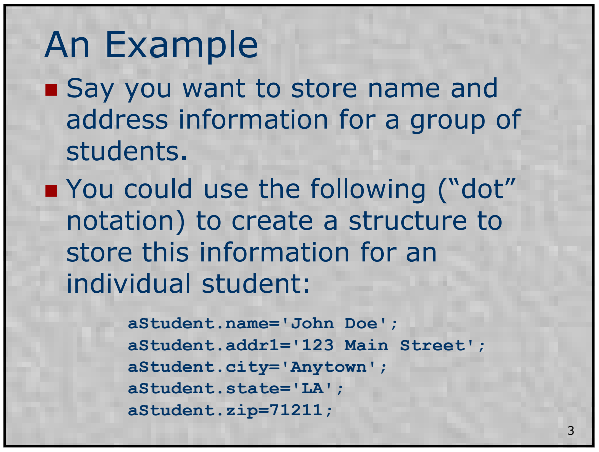
C. S. Tritt, Ph.D.
December 5, 2011



Structure Arrays

- Structures are used to store a variety of related information as a unit.
- Unlike cell arrays, the data is actually stored in the structure rather than just references to data.
- Also unlike cell arrays, structure data storage locations have their own names (rather than numbers).
- In Matlab, arrays of structures can be created.

2



An Example

- Say you want to store name and address information for a group of students.
- You could use the following (“dot” notation) to create a structure to store this information for an individual student:

```
aStudent.name='John Doe';
aStudent.addr1='123 Main Street';
aStudent.city='Anytown';
aStudent.state='LA';
aStudent.zip=71211;
```

3

Resulting Structure

The idea of *fields* and *values* is critical to understanding and using structures.

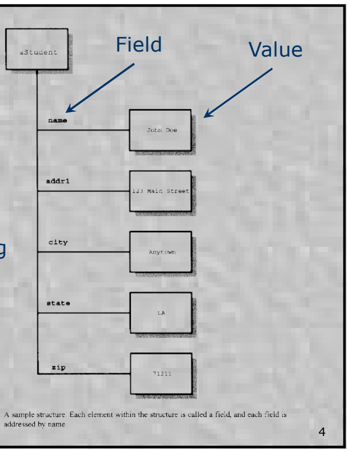


Figure 7.6 (Modified) A sample structure. Each element within the structure is called a field, and each field is addressed by name

Creating Structure Arrays

- In Matlab, structure arrays can be created "on the fly," but code executes faster if you pre-allocate them using the following syntax:

```
structName(size) = struct(field1, value1, field2,
value2, …)
```

For example,

```
student(100) = struct('name', [],'addr1', [], ...
     'city', [], 'state', [], 'zip', [])
```

Resulting Structure (after population)

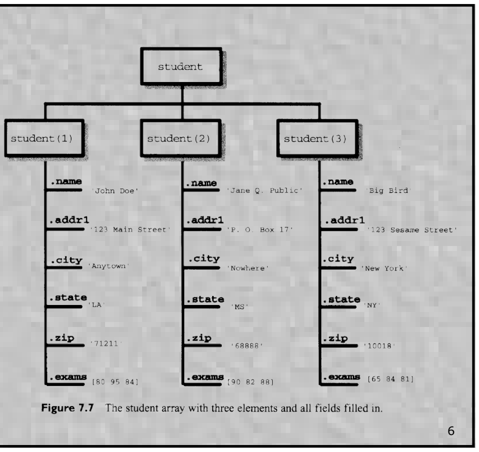
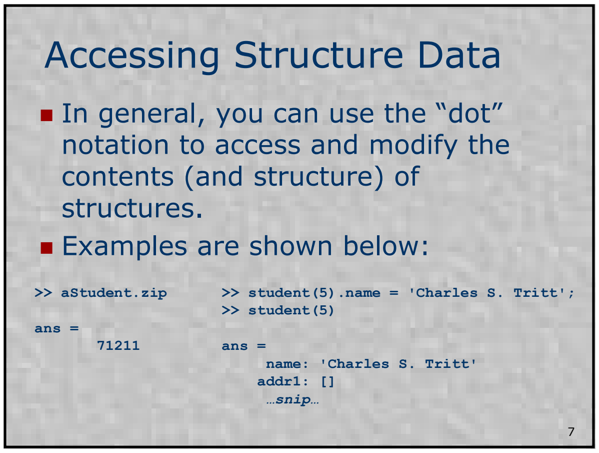


Figure 7.7 The student array with three elements and all fields filled in.



Accessing Structure Data

- In general, you can use the “dot” notation to access and modify the contents (and structure) of structures.
- Examples are shown below:

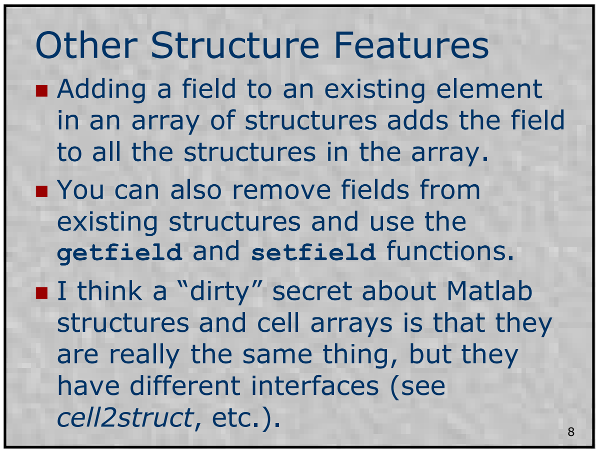
```
>> aStudent.zip

ans =
       71211
```

```
>> student(5).name = 'Charles S. Tritt';
>> student(5)

ans =
     name: 'Charles S. Tritt'
    addr1: []
     …snip…
```

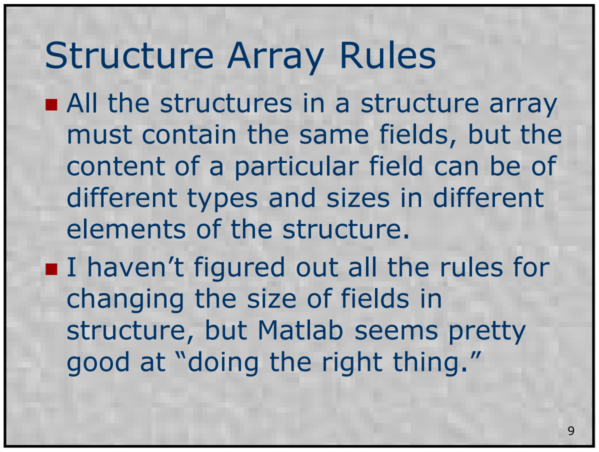
7



Other Structure Features

- Adding a field to an existing element in an array of structures adds the field to all the structures in the array.
- You can also remove fields from existing structures and use the **`getfield`** and **`setfield`** functions.
- I think a “dirty” secret about Matlab structures and cell arrays is that they are really the same thing, but they have different interfaces (see *cell2struct*, etc.).

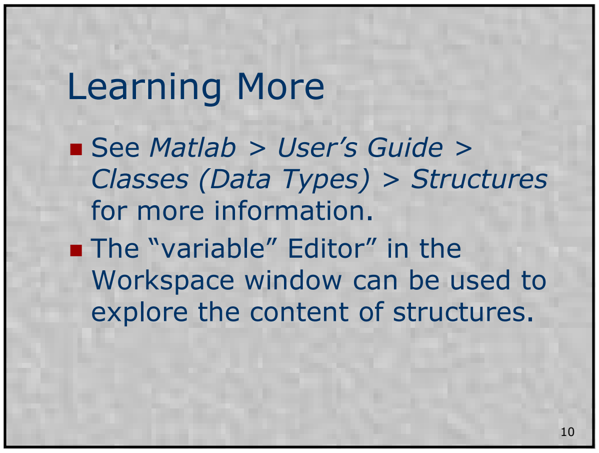
8



Structure Array Rules

- All the structures in a structure array must contain the same fields, but the content of a particular field can be of different types and sizes in different elements of the structure.
- I haven’t figured out all the rules for changing the size of fields in structure, but Matlab seems pretty good at “doing the right thing.”

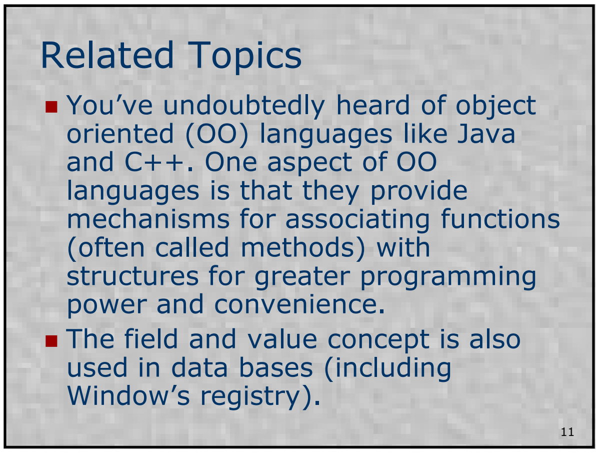
9



Learning More

- See *Matlab > User’s Guide > Classes (Data Types) > Structures* for more information.
- The “variable” Editor” in the Workspace window can be used to explore the content of structures.

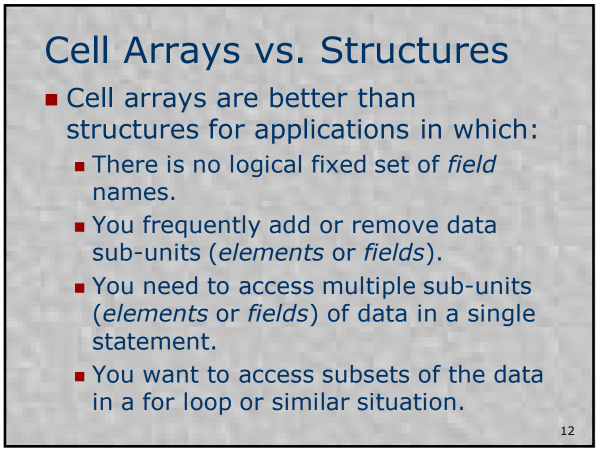
10



Related Topics

- You’ve undoubtedly heard of object oriented (OO) languages like Java and C++. One aspect of OO languages is that they provide mechanisms for associating functions (often called methods) with structures for greater programming power and convenience.
- The field and value concept is also used in data bases (including Window’s registry).

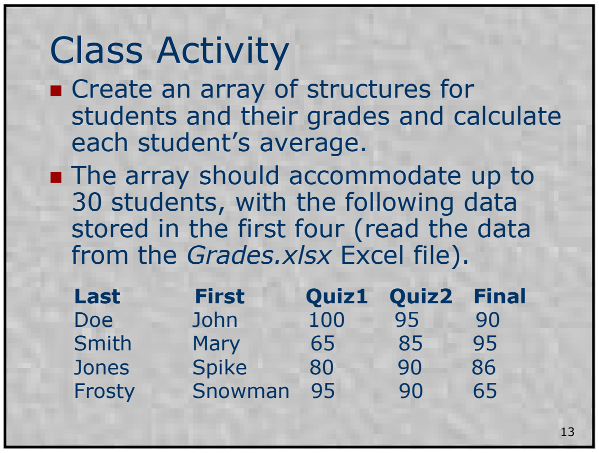
11



Cell Arrays vs. Structures

- Cell arrays are better than structures for applications in which:
  - There is no logical fixed set of *field* names.
  - You frequently add or remove data sub-units (*elements* or *fields*).
  - You need to access multiple sub-units (*elements* or *fields*) of data in a single statement.
  - You want to access subsets of the data in a for loop or similar situation.

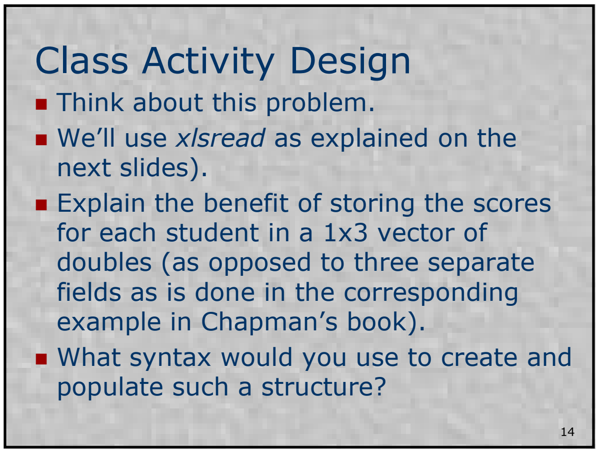
12



Class Activity

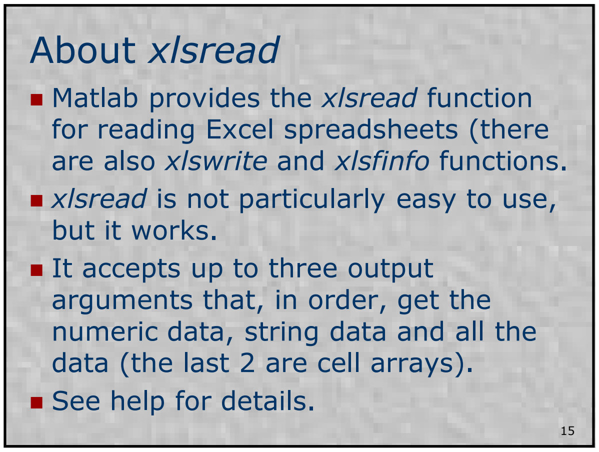
- Create an array of structures for students and their grades and calculate each student's average.
- The array should accommodate up to 30 students, with the following data stored in the first four (read the data from the *Grades.xlsx* Excel file).

Last	First	Quiz1	Quiz2	Final
Doe	John	100	95	90
Smith	Mary	65	85	95
Jones	Spike	80	90	86
Frosty	Snowman	95	90	65



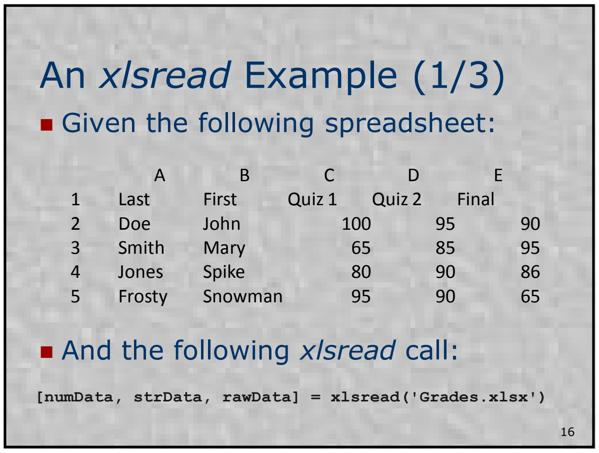
Class Activity Design

- Think about this problem.
- We'll use *xlsread* as explained on the next slides).
- Explain the benefit of storing the scores for each student in a 1x3 vector of doubles (as opposed to three separate fields as is done in the corresponding example in Chapman's book).
- What syntax would you use to create and populate such a structure?



About *xlsread*

- Matlab provides the *xlsread* function for reading Excel spreadsheets (there are also *xlswrite* and *xlsfinfo* functions.
- *xlsread* is not particularly easy to use, but it works.
- It accepts up to three output arguments that, in order, get the numeric data, string data and all the data (the last 2 are cell arrays).
- See help for details.



An *xlsread* Example (1/3)

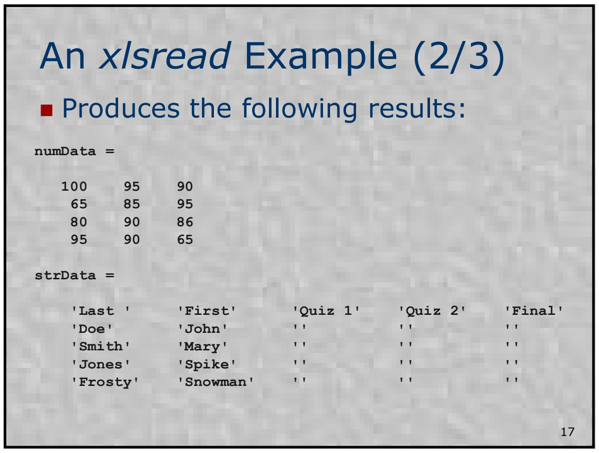
- Given the following spreadsheet:

	A	B	C	D	E
1	Last	First	Quiz 1	Quiz 2	Final
2	Doe	John	100	95	90
3	Smith	Mary	65	85	95
4	Jones	Spike	80	90	86
5	Frosty	Snowman	95	90	65

- And the following *xlsread* call:

```
[numData, strData, rawData] = xlsread('Grades.xlsx')
```

16



An *xlsread* Example (2/3)

- Produces the following results:

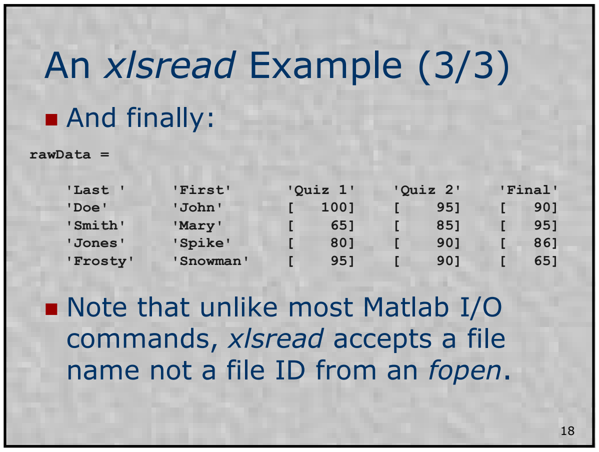
```
numData =

   100    95    90
    65    85    95
    80    90    86
    95    90    65

strData =

    'Last '     'First'      'Quiz 1'     'Quiz 2'     'Final'
    'Doe'       'John'       ''           ''           ''
    'Smith'     'Mary'       ''           ''           ''
    'Jones'     'Spike'      ''           ''           ''
    'Frosty'    'Snowman'    ''           ''           ''
```

17



An *xlsread* Example (3/3)

- And finally:

```
rawData =

    'Last '     'First'      'Quiz 1'     'Quiz 2'     'Final'
    'Doe'       'John'       [   100]     [    95]     [   90]
    'Smith'     'Mary'       [    65]     [    85]     [   95]
    'Jones'     'Spike'      [    80]     [    90]     [   86]
    'Frosty'    'Snowman'    [    95]     [    90]     [   65]
```

- Note that unlike most Matlab I/O commands, *xlsread* accepts a file name not a file ID from an *fopen*.

18