COGS 119/219
MATLAB for Experimental Research

Fall 2014 – Week 3

Structures and

Functions

Structure

- Cell arrays can hold multiple data classes of items together but are hard to use sometimes.

- Structures or structs are another useful way to organize your data.

Structure example

- Suppose that you want to organize information about your experimental subject.
- Let's say you want to collect each subject's number, initials, and age into a single variable.
- You can use a struct.
- A struct is a variable just like what we used before, but is has subvariables called fields.

Structure example

Open a new .m file called struct_example.m

```
>> subject.no = 203;
>> subject.init = ‘AVC’;
>> subject.age = 22;
>> subject.data = rand(5, 4);
```

Structure example

```
>> middle_init = subject.init(1,2);
>> mean_col = mean(subject.data,1);
>> mean_row = mean(subject.data,2);
```

Structure example

```
>> middle_init = subject.init(1,2);
>> mean_col = mean(subject.data,1);
>> mean_row = mean(subject.data,2);

>> subject.no = input('Enter subject number:  ');
>> subject.init = input('Enter subject initials ', 's' );
>> subject.age = input('Enter subject age: ');
```

Struct example: multiple elements

- Open a new .m file called struct_example_multsubjects.m

```
nsubs = 3;
for i = 1:nsubs
      subjects(i).no = input('Enter subject number: ');
      subjects(i).init = input('Enter initials: ', 's');
      subjects(i).age = input('Enter age: ');
      subjects(i).data = rand(5,4);
      % don't make up data in real life!
end;
```

Data of 3 subjects

Struct example: multiple elements

- Be careful about indexing.
- If your struct has more than one element, the index comes before the field.

e.g. `subjects(1).init`

not

`subjects.init(1)`

Struct example: multiple elements

- Be careful about indexing.
- If your struct has more than one element, the index comes before the field.

e.g. `subjects(1).init`
`not subjects.init(1)`

Other Examples:

```
a = subjects(1,2).init;
% b = subjects.init(1,2); % won't work
c = subjects(1,3).init(1);
d = subjects(1,3).data(2,3);
e = subjects(1,2).data(2:5,3:4);
f = subjects(1,2).data(1,1) > subjects(1,3).data(1,1);
```

Struct example: multiple elements

- Calculate the mean of the `data(1,1)` for those subjects who are 25 years old or older.

```
count = 0; selected = [];
for i = 1:length(subjects)
     if subjects(i).age >= 25
          count = count +1;
          selected = [selected, subjects(i).data(1,1)];
     end;
end;
meandata = num2str(mean(selected));
disp (['Mean of (1,1) for people 25 and older: ' meandata]);
```

Run and play with this kind of program

disp([‘There were ’ num2str(count) ‘ subjects over 25’]);