

## MATH.2720 Introduction to Programming with MATLAB

### Cell Arrays and Structures

#### A. Cell Arrays

Cell arrays are like numeric arrays, but their elements can be any MATLAB construct, even another cell array. They are useful for combining disparate types of information into a single variable. For example, suppose you wanted to keep track of weather information, including the date, low and high temperatures, cloud cover (clear, partly cloudy, cloudy), and amount of precipitation. Here is an example of one way you could do that.

```
weather_info = cell(3, 4); %This creates an empty 3x4 cell array
weather_info(1, :) = {'July 1, 2015', [65, 78], 'clear', 0}; %Notice the curly bracket notation
weather_info(2, :) = {'July 2, 2015', [60, 72], 'cloudy', 1};
weather_info(3, :) = {'July 3, 2015', [65, 83], 'partly cloudy', 0};
```

Try typing the following commands to see the difference in the output.

```
weather_info(1, :)
weather_info{1, :}
```

#### B. Structures

Structures are similar to cell arrays, but elements are indexed by name rather than by numerical index. If you wanted to record the same information in a structure array that you recorded in the cell array `weather_info` here is one way you could do it.

```
weather_data(3) = struct('date', 'July 3, 2015', 'temps', [65, 83], 'cover', ...
    'partly cloudy', 'precip', 0); %Starting with the last array entry causes MATLAB
    %to preallocate space for the whole array.
weather_data(1) = struct('date', 'July 1, 2015', 'temps', [65, 78], 'cover', ...
    'clear', 'precip', 0);
weather_data(2) = struct('date', 'July 2, 2015', 'temps', [60, 72], 'cover', ...
    'cloudy', 'precip', 1);
```

A second way to generate the structure `weather_data` given that the cell array `weather_info` already exists is to use the `cell2struct` command

```
weather_data = cell2struct(weather_info, {'date', 'temps', 'cover', 'precip'}, 2);
```

Here is how you can reference the `temps` field in the structure `weather_data(2)`:

```
weather_data(2).temps
```

You can also convert structures to cell arrays using the command `struct2cell`. For example, the following command would convert the structure `weather_data` to the cell array `UML_weather_data`:

```
UML_weather_data = struct2cell(weather_data);
```

#### C. References

1. Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving*, 2nd ed., Elsevier, 2012.
2. Driscoll, *Learning MATLAB*, SIAM, 2009.
3. Knoesen, Amirtharajah, Vahid, and Lysecky, *Programming in MATLAB*, zybooks.com, 2015.

Practice Problems (from Attaway, *MATLAB: A Practical Introduction to Programming and Problem Solving*)

1. Create the following cell array: `ca = {'abc', 11, 3:2:9, zeros(2)}`  
Use the `reshape` function to make `ca` a  $2 \times 2$  matrix. Then write an expression that refers to the last column of this cell array. (You might want to look at the handout on Working with Arrays.)

2. Write a script file defining the following three cell array variables:

```
names = {'Harry', 'Xavier', 'Sue'}  
verbs = {'loves', 'eats'}  
nouns = {'baseballs', 'rocks', 'sushi'}
```

Have your script print a sentence using one random element from each cell array (e.g. `Xavier eats sushi`). Hint: The command `rule = randi([1 3])` will generate a 1, 2, or 3 with equal probability and the command `rule = randi([1 2])` will generate a 1 or 2 with equal probability.

3. Create an array of structures to store information on students' quiz scores. Each structure in the array contains 4 numbers: the student's ID number, quiz 1 score, quiz 2 score, and quiz 3 score. Use the following data:

idNo	q1	q2	q3
44	7	7.5	8
33	5.5	6	6.5
37	8	8	8
24	6	7	8