Lab 08 (Due: Monday, March 21, 2016, 11 : 59 : 00pm)

CSCE 155*N*

1 Lab Objectives

- Write simple cell arrays
- Sting manipulations
- Understand how structures can be used to represent complex information
- Perform basic operations on structures

2 Prior to Laboratory

- Review the laboratory handout
- Read Chapter 7 and Chapter 10 in Attaway

3 Topics Covered in Lab

- Input and output arguments as cell arrays
- Cell arrays
- Structures

4 Activities/Exercises

Before You Begin

- **Download the files from** http://cse.unl.edu/~cse155n/labs/07 **to your** `Z:\csce155nm` **directory**

4.1 `increasingStrings`

- Modify `increasingStrings.m` so that the function returns a cell array of three (3) strings where the first string has length 2, the second has length 3, and the third has length 4
- The provided function `randString` will generate a string with a length between 0 and 6 (inclusive)
- If `randString` returns a string of a length that is already in the cell array, keep the string that is already in the cell array

4.1.1 Example

```
>> increasingStrings

ans =

    'vp'    'hkn'    'IGZx'
```

4.2 randomSentence

- Modify `randomSentence.m` so that the function returns words that can form a random sentence that is generated from banks of names, verbs, and nouns
- The banks will be stored as cell arrays of strings

4.2.1 Example

```
>> for i = 1 : 5
randomSentence
end

ans =

    'Stewart'    'milks'    'bicycles'


ans =

    'Patrick'    'rides'    'cows'


ans =

    'Patrick'    'herds'    'sheep'


ans =

    'Stewart'    'rides'    'bicycles'


ans =

    'Ryan'    'herds'    'bicycles'
```

4.3 printRecord

- Modify `printRecord.m` so that the function returns a string that displays information about the record

- The record will be provided as a cell array
- Use `sprintf`

4.3.1 Example

```
>> record = { 0 , 'Ryan' , 43 }

record =

    [0]    'Ryan'    [43]

>> printRecord( record )

ans =

Ryan (ID: 0): 43 points
```

4.4 printInfo

- Modify `printInfo.m` so that the function returns a cell array of strings that reports the names, heights, and final weights of the provided experiment subjects
- The subjects' information will be provided in an array of structures
- Use `sprintf`

4.4.1 Example

```
>> subjects.name = 'Ryan';
subjects.id = 0;
subjects.weight = [ 134 145 ];
subjects.height.feet = 6;
subjects.height.inches = 0;
for s = 2 : 4
subjects( s ).name = char( randi( 26 , 1 , randi( 10 ) ) + 96 );
subjects( s ).id = randi( 100 );
subjects( s ).weight = randi( 100 , 1 , 2 ) + 100;
subjects( s ).height.feet = randi( 4 ) + 3;
subjects( s ).height.inches = randi( 12 ) - 1;
end
>> printInfo( subjects );
>> ans'

ans =

    'Ryan: 6'0", 145 pounds'
    'l: 4'5", 180 pounds'
    'qsthr: 5'11", 112 pounds'
    'pftg: 7'6", 190 pounds'
```

5 Code Documentation

Remember to document your files in the way that we did for the previous labs. It will come in handy when you look back at code after a long time, or when someone else is trying to understand what your code does.

6 What to Submit

You will be submitting six (6) files (`increasingStrings.m`, `randomSentence.m`, `printRecord.m`, `contributions07lab.txt`, `members07lab.txt`, and, `printInfo.m`).

7 Additional Resources

Online MATLAB Documentation
CSE Webhandin
CSE webgrader

8 Point Allocation

Component	Points
`increasingStrings.m`	20
`randomSentence.m`	20
`printRecord.m`	20
`printInfo.m`	20
`members07lab.txt`	5
`contributions07lab.txt`	5
webgrader PDF	10
Total	100