**Lab: Data Structures**

This would be a Models II lab assigned pretty late in the term, after they've gotten some practice with functions. The point of this lab is to teach the students how to create a structure, how to access it both to read and write data, and how to create an array of structures. Using structured data is a huge part of programming, and this lab will be especially helpful for students who would like to extend their programming knowledge beyond Matlab. A lot of complaints around Matlab arise because it's not "real programming," but this lab will introduce a very real and necessary programming concept.

**Part 1:**

Students create a structure Student, which contains following data: age, major, and year. It would look like this

Student = struct('age', 0, 'major', '0', 'year', 0)

This initializes the values to 0 so students can practice accessing them with "."

Students will then enter their own data using the . method:

Student.age = 20; Student.major = 'Computer Science'; Student.year = 3

Then they will print out the data using the same . format

**Part 2:**

Now students will create an array of structures using the people at their table. The format for entering data will be the same except now it will be

Student(1).age = ?; Student(1).major = ?; Student(1).year = ?

Student(2).age = ?; Student(2).major = ?; Student(2).year = ?

Etc.

They will then print out the data in the same fashion.

**Part 3:**

This part will focus on nested structures. The students will redefine their structure to include a field that is also a structure. The data is initialized to a null array.

Student = struct('major', {}, 'year', {}, 'Info', {})

Info is going to include age and M#. To make it into a structure as well, students can do the following:

Student.Info = struct(age, {}, M_num, {})

Now when students want to write data, they must use two dots:

Student(1).major = ?; Student(1).year = ?; Student(1).Info.age = ?; Student(1).Info.M_num = ?

Students would enter the information of the people at their table, making up M#s.