CISC106 Fall 2009 Lab06

- Review the code examples from class.
- Some programs below are associated with a question. **Answer the questions** using comments below your code in the m-file.
- The office hours of the TAs and the instructor are on the class website. Visit us!
- **NOTE:** Every function comment section should contain, at a minimum, *three examples* of the function being called and the result of evaluating the call. These examples must include boundary conditions (as discussed in class). Your test files must cover *at least* these exact examples (otherwise, why did you choose them?) and possibly more. Testing is important.
- **Every** M-file you write or modify must be demonstrated, either by running a script test file in a diary or by testing at the command line. Note that if you write function **foo.m** and test script **fooTest.m**, you can demonstrate both by running only fooTest.m (assuming fooTest works).

Problems

1. Go to the mathworks.com website. Click on $Support \rightarrow Product \rightarrow Matlab \rightarrow How\,To\,(function\,list) \rightarrow Functions\ (on\ left) \rightarrow Graphics \rightarrow AnnotatingPlots \rightarrow rectangles$. Read about this built-in function, then use it to make three rectangles on a single figure that do not touch each other, and one circle. Save the plot as a .png and put it on your webpage, then call your TA over to look at it during lab before proceeding. Show your TA how you can delete one of the shapes using its handle (yes, this info is on the webpage). The plot title should **include your name** (see 2.11 and 3.5 in Chapman).

2. We can change the size of a matrix we are using in Matlab. Try the following sequence of commands:

```
x = []
x(2) = 5
x(2,3) = 80
```

   What is the size of x after each command? Answer for yourself only, and check with the size command.

   Being able to "grow" a matrix to fit our needs is a powerful tool, but not without cost.

   Write a function "grow" that uses a loop to grow a vector one element at a time, from size one to size 10,000. Place a zero in each element.

   Write a second function "preallocate" that uses the "zeros" function to create a 1x10000 vector.

   Determine a number of calls of the "preallocate" function that will give you measurable time. Do three timing runs of each function with a timing function. After the timing runs your script should display the size of the vector created by each function (not the vectors!).

   Plot your data, showing each point (of course!). Post your labeled plot on your website. The plot title should **include your name** (see 2.11 and 3.5 in Chapman).

3. Try this in the interpreter (at the command prompt). When you can do it without mistakes, record it all in a diary.

   Create a turtle structure with two fields: 4 legs and a color of your choice.

   Now add a second turtle structure, using the fact that the first turtle was really an array of one turtle, and that the array can be extended the same way we grow a matrix, using a new index and assigning a new value. Give the second turtle both color and legs. After you create the second turtle, show how to display the data in the first turtle and the second turtle.

   Add two more turtles for a total of four. Now write a function that will take the turtle array as a parameter and print each turtle[1]. Demonstrate creation of all turtles and the running of the function in the diary.

If your TA requires a paper copy, be sure that you have a printed copy of your function M-files, script M-files,image files, and diary files demonstrating your testing. All must be stapled together, with your name and lab section on the top page.

Be sure that you upload a copy of all the MATLAB function, script, imasge, and diary files to Sakai. Then, click submit ONLY ONCE to send these to your Sakai and your TA.

On the first page of every printed copy for this course, your name, section, and TA's name must appear.

[1]This function only requires one example.