## CISC106 Fall 2008 Lab09

- 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. Write a function that prints n spaces. Use the function, along with one of your asterisk functions from the previous lab, to write four new triangle drawing functions that behave as follows:

```
>> upRight(3)
  *
 * *
* * *
>> upLeft(4)
*
* *
* * *
* * * *
>> downRight(2)
* *
 *
>> downLeft(3)
* * *
* *
*
```

- 2. Place calls to your four triangle functions inside a wrapper function (see previous lab). Your wrapper will 1) prompt the user for a size, then 2) prompt the user for the type of triangle to print, then use a *switch* statement to call the appropriate function. You may ask your user to specify the kind of triangle with either a number or a word (number is easier).
- 3. When your wrapper is correctly prompting and selecting, add a loop so that it repeats until the user asks for a size of -1.

- 4. Demonstrate in a diary writing a matrix to a file and reading it back in, as follows:
  - (a) Create a matrix x.
  - (b) Save it using:

>> save x.txt x -ascii

- (c) Use pico to open the file x.txt. What do you see?
- (d) Clear the variable x in Matlab.
- (e) Attempt to examine x in Matlab.
- (f) Load the file:

>> x = load('x.txt')

- (g) Attempt to examine x in Matlab.
- (h) Stop your diary.
- 5. Write a Matlab function which takes an image array and a string as input arguments. It assigns in its output argument, array of frequency of occurrence of each intensity value (also called histogram). Based on whether the string is 'red', 'green', or 'blue', the output argument array contains the corresponding histogram of length 256 (as explained in the class). Use *switch* statement based on the string argument value to decide which channel's histogram to compute, and make sure to use 'otherwise' for invalid input.

Now, write a matlab script (wrapper) program which asks user to input image name, then ask which channel's histogram to compute (red, green, or blue). After calling the above function, plot the histogram for the user. Make sure to use 'imfinfo' to get the image dimensions (similar to class image programs).

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.