CISC106 Fall 2009 Lab07

- 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. Use the Unix copy command to copy your drawTriangle function from lab02 to your new lab directory (this is good practice, yes?). Write a function with a loop that asks the user for an input size, and calls drawTriangle on the size. The loop should stop when the user enters a negative size.

2. Write a function that prints n spaces. Use the function, along with your lineOfStar function from lab02, to write four new triangle drawing functions that behave as follows:

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

3. This problem is about using functions that handle I/O to call other functions that really do things.

   Place calls to your four triangle functions inside a wrapper function[1]. Your wrapper function will 1) prompt the user for a size, then 2) prompt the user for the type of triangle to print, then use a *switch*

[1]A wrapper is a function that wraps around another function to add utility. In this case we have a nice asterisk function that takes a parameter, and we write another function to handle I/O separately so we don't have to modify the asterisk function.

statement (see your text or the help fcn) 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).

4. When your wrapper is correctly prompting and selecting, add a loop so that it repeats until the user asks for a size of -1.

5. 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 emacs 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.

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.