Spring 2018

LAB 4 - FUNCTIONS

Objectives: To learn how to use MATLAB functions, in particular:

- How a function is constructed
- How to receive input and deliver output

A MATLAB function is different from a MATLAB script in that it can receive input and deliver output. The structure of a function is as follows (you do not need to type this...it is for informational purposes only):

```
function [out1,out2,...,outN] = functionName(in1,in2,...,inM)
% This section defines the "help" information.
% Include all usage information here.
% Anything in this block will be displayed when you type
% "help functionName" in the command window.
% You can also include "help functionName" in your script
% to show the user how information is entered into functions
% without them asking.

out1 = f(in1,in2,...,inM); % supress output in functions
out2 = g(in1,in2,...,inM);
...
outN = h(in1,in2,...,inM);

y = 7;
% variables defined here stay here. when we run this in
% our script or command window y is undefined

end % not necessary to end a function unless you have subfunctions below
```

1. **Boot It Up**

   (a) Start up MATLAB
   - There is no need to start a diary file today.

2. **Functions**

   (a) Write a function named *rightTriangle.m* that allows the user to input 2 sides of a right triangle and delivers the following information back to the user:

   i. the value of the hypotenuse

   ii. both interior angles, θ_1 (angle opposite $side_1$) and θ_2 (angle opposite $side_2$). Recall,

$$\begin{aligned} \sin(\theta_1) &= \frac{s_1}{h} \\ \theta_1 &= \sin^{-1}\left(\frac{s_1}{h}\right) \end{aligned}$$

   You can use $asin(s_1/h)$ and $asin(s_2/h)$ to find θ_1 and θ_2, respectively.

   iii. $\cos\theta_1$, $\cos\theta_2$, $\sin\theta_1$, $\sin\theta_2$, $\tan\theta_1$, $\tan\theta_2$

   An example of usage is as follows:

```
>> rightTriangle(sqrt(3),1)

    hypotenuse = 2.00

    theta1 is the angle between 1.00 and 2
    theta2 is the angle between 1.73 and 2
```

```
theta1 = 1.05 radians
sin(theta1) = 0.87
cos(theta1) = 0.50
tan(theta1) = 1.73

theta2 = 0.52 radians
sin(theta2) = 0.50
cos(theta2) = 0.87
tan(theta2) = 0.58
```

Be sure to include a 'help' block that gives the user usage instructions. In the command window enter 'help rightTriangle' to view the usage information, then run your function from the command window using the following values:

i. side1 = 3, side2 = 4
ii. side1 = 38, side2 = 93
iii. side1 = 5, side2 = 5

(b) Write 6 functions that convert temperature values between Celcius, Fahrenheit, and Kelvin (0° C = 273.15 K). Name your functions c2f.m, c2k.m, f2c.m, f2k.m, k2c.m and k2f.m. Your output should be formatted as such:

```
>> k = c2k(0)

  k =

      273.15
```

Be sure to include a 'help' block that gives the user usage instructions. In the command window enter 'help c2f' to view the usage information, then run your function from the command window to convert the following values:

i. 33° C to F and K
ii. −55° F to C and K
iii. 2320 K to C and F

Be sure to keep these files...we will likely use them in future labs. You may need the following equations:

$$C = \frac{5}{9}(F - 32)$$
$$F = \frac{9}{5}C + 32$$

(c) The distance between any two points (x_1, y_1) and (x_2, y_2) is given by:

$$distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The area of a triangle is:

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

where a, b and c are the lengths of the sides of the triangle, and s is equal to half the sum of the lengths of the three sides of the triangle: $s = \frac{1}{2}(a + b + c)$. Write a script, *triangle.m*, that will prompt the user to enter the coordinates of three points that that determine a triangle (i.e., the x and y coordinates of each point). The script will then calculate and print the area of the triangle, along with a plot of the triangle. You will need a distance function, *triangleDist.m*, that returns the distance between 2 points. This function will need 4 inputs to perform the calculation (x_1, y_1, x_2, y_2). An area function, *triangleArea.m*,

will need to accept 3 side lengths Your script will call the distance function 3 times and the area function once. Use the *patch* function to plot your triangle.

$$\begin{aligned} X &= [x_1; x_2; x_3] \\ Y &= [y_1; y_2; y_3] \end{aligned}$$

(d) Once the TA has checked your files (rightTriangle.m, c2f.m, c2k.m, f2c.m, f2k.m, k2c.m, k2f.m, triangle.m, triangleDist.m and triangleArea.m) you are free to go. Do not forget to upload these files to Canvas! Log off of your machine before you leave the lab.