ENGR 15100: Software Tools for Engineers
Spring 2015

Computer Assignment #5

Due: Tuesday, February 24, 2015, 9am CST

Departments of Engineering
School of Engineering, Mathematics, & Sciences
Purdue University Calumet

PURDUE
UNIVERSITY
CALUMET

1. OBJECTIVE

Continue working with obtaining user input and displaying formatted output. Additionally, become familiar with two-dimensional plotting.

2. PROCEDURE

Task I [35 points]

Create a MATLAB script file having the name `LASTNAME_LAB5_TASK1.m` and perform the following sequence of steps in the file. ***Unless otherwise specified, do not suppress the output to the MATLAB Command Window.***

(a) **[1 point]** Clear the MATLAB Workspace, clear the contents of the MATLAB Command Window, and close all MATLAB Figure Windows.

(b) **[4 points]** Using the built-in **`fprintf()`** function, display a message in the MATLAB Command Window similar to the message shown below. *Notice the tab at the beginning of the message and the blank line between the message and the prompt*.

```
	%%Lab #5 – Task #1 Solution of <YOUR FULL NAME> \\

>>
```

(c) **[2 points]** Using the built-in **`input()`** function, prompt the user to enter quantities for two *variables* named `dim` and `bounds`.

- Variable `dim` should be assigned a 2-element *row vector* representing the dimensions of a matrix `M` created in step (d). The first and second element of row vector `dim` should contain the values of the number of rows and number of columns of matrix `M`, respectively.
- Variable `bounds` is also a 2- element row vector representing the upper and lower bound of an open interval from which random real numbers are uniformly chosen. The first and second element of row vector `bounds` should contain the lower and upper bound values, respectively.

Prompt the user twice, once for variable `dim` and again for variable `bounds`. After completing this step, you should see an output in the MATLAB Command Window similar to the sample output shown below. Suppress the output to the MATLAB Command Window.

```
Enter the dimensions [rows, columns] of a matrix M: <enter row vector>
Enter bounds [lower, upper] of an open interval: <enter row vector>
```

When testing your script, enter `[5 6]` for variable `dim` and `[-20, 15]` for variable `bounds`.

(d) **[5 points]** Create a variable named `M` and assign to it a matrix having dimensions indicated by row vector **`dim`** whose elements are each randomly generated random real numbers chosen uniformly from the open interval indicated by row vector **bounds**. Use the built-in **`rand()`** function.

(e) **[10 points]** Using the built-in **`fprintf()`** function, display the scalar `M(2,2)` followed by scalar `M(2,3)` using a *single format string*.

- Format scalar `M(2,2)` as a real number and display its value within a 6-digit field width showing 2 digits after the decimal place.
- Format scalar `M(2,3)` using scientific notation and display its value within an 12-digit field width showing 4 digits after the decimal point.

After completing this step, you should see an output in the MATLAB Command Window similar to the sample output shown below.

```
Element M(2, 2) is  -2.86 and element M(2, 3) is   6.4140e+00
```

(f) **[13 points]** Using the built-in **`fprintf()`** function, display scalar `M(3,4)` followed by element `M(3,5)` using a *single format string*.

- Format scalar `M(3,4)` using the most compact numeric notation chosen among fixed-point and scientific notation. Left justify the output within a 7-digit field width showing 4 significant digits.
- Format scalar `M(3,5)` as a real number and display its value within a 9-digit field width showing only 1 digit after the decimal point. Pad empty spaces within the field width with zeroes and always show the sign of the scalar.

After completing this step, you should see an output in the MATLAB Command Window similar to the sample output shown below.

```
Element M(3, 4) is -14.67  and element M(3, 5) is -000017.3
```

Task II [30 points]

Create a MATLAB script file having the name `LASTNAME_LAB5_TASK2.m` and perform the following sequence of steps in the file. ***Unless otherwise specified, do not suppress the output to the MATLAB Command Window.***

(a) **[2 points]** Clear the MATLAB Workspace, clear the contents of the MATLAB Command Window, and close all MATLAB Figure Windows.

(b) **[4 points]** Create a *variable* named `t` and assign to it a row vector whose elements are equally spaced scalar values starting from -4 and ending with +4 in increments of `0.1`.

(c) **[8 points]** Create a *variable* named `f` and assign to it a row vector whose elements are values obtained from evaluating the expression below for corresponding elements of row vector `t`.

$$f = t^3 - 3t^2 + 2t - 4$$

(d) **[1 point]** Using the built-in `figure()` function, open a new Figure Window named `Figure 2`.

(e) **[5 points]** Using the built-in `plot()` function, plot `f` vs. `t` using a dashed red line.

(f) **[7 points]** Format the plot generated by the previous step using the built-in `grid` command in combination with the built-in functions `title()`, `xlabel()`, and `ylabel()`.

- **[2 points]** Title the plot using the string `'f(t) = t^3 - 3t^2 + 2t - 4'`.
- **[2 points]** Label the graph's horizontal axis with the string `'t'`.
- **[2 points]** Label the graph's vertical axis with the string `'f(t)'`.
- **[1 point]** Activate the major access grid to aid in visualizing the plot

(g) **[1 point]** Using the `pause` command, pause the execution of your script.

(h) **[2 points]** Use the built-in `close()` function in combination with the `gcf` command to close the Figure Window currently in focus.

Now, execute your script. After inspecting your results and the corresponding graph, press `ANY KEY` to end your script's execution.

Task III [35 points]

Create a MATLAB script file having the name `LASTNAME_LAB5_TASK3.m` and perform the following sequence of steps in the file. ***Unless otherwise specified, do not suppress the output to the MATLAB Command Window.***

(a) **[2 points]** Clear the MATLAB Workspace, clear the contents of the MATLAB Command Window, and close all MATLAB Figure Windows.

(b) **[1 points]** Create a variable named `radius` and assign to it a row vector containing equally spaced values starting from `1` and ending in `15` in increments of `1`.

(c) **[2 points]** Create a variable named `circumference` and assign to it a row vector containing the circumference of circles having radii in the range `[1, 15]`.

(d) **[2 points]** Create a variable named `area` and assign to it a row vector containing the area of circles having radii in the range `[1, 15]`.

(e) **[2 points]** Create a `2 x 1` grid of graphs in a Figure Window named `Figure 1` and activate a graph (axis) in grid position `1`. Use built-in functions `figure()` and `subplot()`.

(f) **[6 points]** Using <u>only one</u> instance of the built-in `plot()` function, graph two data sets on the same graph in `Figure 1` according to the following guidelines.

- The first data set represents the circumference of various circles vs. their radii. The data points should be indicated by magenta plus markers that are not connected.
- The second data set represents the area those circles vs. their radii. The data points should be connected by a dashed cyan line.

(g) [6 points] Format the graph in `Figure 1` according to the following formatting guidelines using built in functions/commands **`xlabel()`**, **`ylabel()`**, **`title()`**, **`grid`**, **`legend()`**, and **`text()`**.

- [0.5 points] Turn on grid lines to more easily visualize the graph.
- [0.5 points] Label the graph's horizontal axis `'Radius (cm)'`.
- [0.5 points] Title the graph `'Circumference and Area of a Circle vs. Radius'`.
- [1.5 points] Uniquely identify each data set with the string `'Circumference vs. Radius'` and `'Area vs. Radius'`, respectively.
- [3 points] Annotate the area vs. radius data set at point $(7, 49\pi)$ with the annotation string `'Area of a Circle with Radius = 7 cm'`

(h) [1 point] Use the built-in function **`subplot()`** to activate a graph (axis) in position 2 of `Figure 1`.

(i) [7 points] Repeat step (f), but this time, plot the two data sets in grid position 2 of `Figure 1` using multiple instances of the built-in **`plot()`** function in combination with multiple instances of the built-in `hold` command.

(j) [3 points] Repeat step (g) to format the graph (axis) in grid position 2 of `Figure 1`.

(k) [1 point] Using the `pause` command, pause the execution of your script.

(l) [2 points] Use the built-in **`close()`** function in combination with the `gcf` command to close the Figure Window currently in focus.

Now, execute your script. After inspecting your results and each corresponding graph, press `ANY KEY` to end your script's execution.

Task IV

Upload the following files onto Blackboard Learn.

(a) Script file `LASTNAME_LAB5_TASK1.m`.

(b) Script file `LASTNAME_LAB5_TASK2.m`.

(c) Script file `LASTNAME_LAB5_TASK3.m`.