

ENGR15100: SOFTWARE TOOLS FOR ENGINEERS**Laboratory 7**

PURPOSE: Learn about conditional statement.

For each problem, create a MATLAB script file and name it FIRSTNAME_LASTNAME_LAB7_problemX.m. Put ALL the commands for the required steps in your script file:

- Be sure to clear the display and the memory.
- Display your name.
- Separate and label different steps using comments.
- For each plot question, you can use pause and close functions to stop and check the plots, and then close the figure window.

You can use following template for each of the Problem.

```
%{
Class      : ENGR15100: Software Tools for Engineers
Instructor : Xiaoli Yang
Author     : [Student's Name]
Assignment : Lab [No.]
File Name  : Firstname_Lastname_LAB[No.]_Problem[No.].m
            (eg: Xiaoli_Yang_LAB1_Problem1.m)
Date       : [MM]/[DD]/[YY]
%}

%clear screen
clc

%clear workspace
clear

%display your name
disp('Your Full Name Here');
disp('Starting code: ');

%Start your source code here%

%End your code
disp('Completed');
```

SUBMITTING YOUR LAB:

Submit your lab by uploading .m file using the Brightspace Assignment feature no later than the date specified.

PROBLEM 1: The Quadratic Formula (40 points)

Write a program to determine the real roots of a general second-degree polynomial of the form $ax^2 + bx + c$. The roots of the quadratic equation $ax^2 + bx + c = 0$ may be found by using the quadratic formula provided below for your reference. Note, $D = b^2 - 4ac$ is known as the discriminant.

$$r_1, r_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{D}}{2a}$$

The roots (r_1, r_2) may be classified according to the value of the discriminant D as follows:

- (a) When D is positive, the equation $ax^2 + bx + c = 0$ has two real-valued and distinct roots.
- (b) When D is zero, the equation $ax^2 + bx + c = 0$ has two real-valued but repeated roots.
- (c) When D is negative, the equation $ax^2 + bx + c = 0$ has two complex-valued and distinct roots.

Your program should:

- (d) Prompt the user for the scalar values of a , b , and c .
- (e) Compute the value of the discriminant D . Suppress any Command Window output.
- (f) Display the number of real-valued roots according to value of D . Utilize a conditional if-end statement.
- (g) When the polynomial has real-valued roots, compute those root values. Suppress any Command Window output.
- (h) When the polynomial has real-valued roots, display those roots as real numbers showing a maximum of 3 digits after the decimal point for each root value.

Test your script with at least the following polynomials.

$$x^2 + 30x + 200$$

$$x^2 + 8x + 225$$

$$2x^2 + 24x + 7$$

Sample executions of the program are shown below. At the beginning of your program, make sure to clear all variables from the MATLAB Workspace as well as the contents of the MATLAB Command Window.

```
>> FIRSTNAME_LASTNAME_LAB7_problem1
Enter a value for coefficient a: -2
Enter a value for coefficient b: 7
Enter a value for coefficient c: 4
The polynomial has two real roots.
Root r1 = -0.500 and root r2 = 4.000

>> FIRSTNAME_LASTNAME_LAB7_problem1
Enter a value for coefficient a: 2
Enter a value for coefficient b: 8
Enter a value for coefficient c: 8
The polynomial has two repeated real roots.
Root r1 = -2.000 and root r2 = -2.000

>> FIRSTNAME_LASTNAME_LAB7_problem1
Enter a value for coefficient a: -5
Enter a value for coefficient b: 3
Enter a value for coefficient c: -4
```

PROBLEM 2: Classifying a Cartesian Data Point (30 points)

Write a program to prompt the user to enter a value for a variable x and a value for a variable y . The values for variables x and y correspond to a point $P = (x, y)$ located in the Cartesian plane (i.e. the Rectangular x - y plane).

Using a combination of conditional **if** statements and the built-in **fprintf** function, the program should display one of the following messages according to the value of x and y . Format the display of the coordinates as fixed-point numbers showing a maximum of 2 digits after the decimal point.

- Point (xVal, yVal) is located on the origin.
- Point (xVal, yVal) is located on the x -axis.
- Point (xVal, yVal) is located on the y -axis.
- Point (xVal, yVal) is located in quadrant I.
- Point (xVal, yVal) is located in quadrant II.
- Point (xVal, yVal) is located in quadrant III.
- Point (xVal, yVal) is located in quadrant IV.

Then, the program displays how far (i.e. distance) point (x, y) is from the origin. Format the display of the distance as a real number showing 3 digits after the decimal point. Test your script for all possible locations of (x, y) . Some sample executions of the script are shown below.

```
>> FIRSTNAME_LASTNAME_LAB7_problem2
Enter a value for the x coordinate: 3.2
Enter a value for the y coordinate: 1.9
Point (3.20, 1.90) is located in quadrant I.
Point (3.20, 1.90) is 3.722 units away from the origin.

>> FIRSTNAME_LASTNAME_LAB7_problem2
Enter a value for the x coordinate: -3.2
Enter a value for the y coordinate: 1.9
Point (-3.20, 1.90) is located in quadrant II.
Point (-3.20, 1.90) is 3.722 units away from the origin.

>> FIRSTNAME_LASTNAME_LAB7_problem2
Enter a value for the x coordinate: 0.0
Enter a value for the y coordinate: -3.8
Point (0.00, -3.80) is located on the y-axis
Point (0.00, -3.80) is 3.800 units away from the origin.
```

PROBLEM 3: Months of the Year (30 points)

Write a program that determines the name of a calendar month given an integer representing the month's number. The program should carry out the general steps outlined below and utilize **if-end** statements. Unless otherwise specified, suppress output to the Command Window.

- (a) At the beginning of your script, make sure to clear all variables defined in the MATLAB Workspace as well the contents of the MATLAB Command Window.
- (b) Prompt the user to enter an integer that represents one of the twelve months in the standard calendar year (1 = January, 12 = December). Assume the user always enters an integer.
- (c) In case the user enters an invalid month number, report back to the user that he/she must re-run the script because an invalid month number was entered. The program should terminate at this point and no “calculations” should be performed!
- (d) For valid month numbers, display to the MATLAB Command Window the name of the month corresponding to the integer entered by the user.

Test your script for all possible valid calendar month numbers and at least two invalid calendar month numbers. Some sample executions of the program are shown below.

```
>> FIRSTNAME_LASTNAME_LAB7_problem3
Enter an integer corresponding to a calendar month: 3
Calendar month 3 corresponds to March.

>> FIRSTNAME_LASTNAME_LAB7_problem3
Enter an integer corresponding to a calendar month: 7
Calendar month 7 corresponds to July.

>> FIRSTNAME_LASTNAME_LAB7_problem3
Enter an integer corresponding to a calendar month: -2
Month #-2 is an invalid month!
Please re-run the script to try again.
```