# ENGR 15100: Software Tools for Engineers
## Spring 2015

## Computer Assignment #6

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

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

PURDUE
UNIVERSITY
CALUMET

# 1. OBJECTIVE

Become familiar with logical types, Boolean operations, and conditional statements.

# 2. PROCEDURE

## Task I: Relational/Logical Expressions [50 points]

Create a MATLAB script file having the name `LASTNAME_LAB6_TASK1.m` and perform the following sequence of steps in the file. Do not suppress any output from MATLAB.

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

(b) **[2 points]** Create a variable named `dayTemps_CHI` and assign to it a vector whose elements are the following daily temperatures (in ˚F) recorded for Chicago in the month of August.

```
[75 79 86 86 79 81 73 89 91 86 81 82 86 88 89 90 82 84 81 79 73 69
 73 79 82 72 66 71 69 66 66]
```

(c) **[2 points]** Create a variable named `dayTemps_SF` and assign to it a vector whose elements are the following daily temperatures (in ˚F) recorded for San Francisco in the month of August.

```
[69 68 70 73 72 71 69 76 85 87 74 84 76 68 79 75 68 68 73 72 79 68
 68 69 71 70 89 95 90 66 69]
```

(d) **[5 points]** Determine how many days San Francisco's temperature was above average.

(e) **[5 points]** Compute how many days Chicago's temperature was in the range `[62˚F, 78˚F]`.

(f) **[5 points]** Compute how many days San Francisco's temperature was cooler than 72˚F or warmer than 80˚F.

(g) **[5 points]** Compute how many days Chicago's temperature was not between 70˚F and 90˚F, inclusive.

(h) **[5 points]** Compute how many days San Francisco's temperature was not colder than 73˚F and not warmer than 89˚F.

(i) **[5 points]** Compute Chicago's temperatures that are warmer than 84˚F but cooler than 90˚F?

(j) **[5 points]** Compute San Francisco's temperatures that are warmer than 65˚F, cooler than 72˚F, but not 69˚F?

(k) **[5 points]** Compute on which day(s) San Francisco's temperature was *warmer or the same as* the temperature in Chicago?

(l) **[5 points]** Compute on which day(s) the temperature was *the same* in both cities.

## Task II: The Quadratic Formula [50 points]

In a script named **LASTNAME_LAB6_TASK2.m**, 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:

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

Your program should:

- **[10 points]** Prompt the user for the scalar values of $a$, $b$, and $c$.
- **[5 points]** Compute the value of the discriminant $D$. Suppress any Command Window output.
- **[15 points]** Display the number of real-valued roots according to value of $D$. Utilize a conditional **if-end** statement.
- **[15 points]** When the polynomial has real-valued roots, compute those root values. Suppress any Command Window output.
- **[5 points]** 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.

$x^2 + 30x + 200$ $x^2 + 8x + 25$ $2x^2 + 24x + 7$

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.

```
>> LASTNAME_LAB6_TASK2
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

>> LASTNAME_LAB6_TASK2
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

>> LASTNAME_LAB6_TASK2
Enter a value for coefficient a: -5
Enter a value for coefficient b: 3
Enter a value for coefficient c: -4
The polynomial has zero real roots.
```

## Task III Classifying a Cartesian Data Point [50 points]

In a script file named **LASTNAME_LAB6_TASK3.m**, write a MATLAB program that 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).

**[42 points]** 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.

- **[6 points]** Point **(xVal, yVal)** is located on the origin.
- **[6 points]** Point **(xVal, yVal)** is located on the $x$-axis.
- **[6 points]** Point **(xVal, yVal)** is located on the $y$-axis.
- **[6 points]** Point **(xVal, yVal)** is located in quadrant I.
- **[6 points]** Point **(xVal, yVal)** is located in quadrant II.
- **[6 points]** Point **(xVal, yVal)** is located in quadrant III.
- **[6 points]** Point **(xVal, yVal)** is located in quadrant IV.

**[8 points]** 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. Consider using the MATLAB **doc** to investigate the built-in **hypot()** function to assist you.

Test your script for all possible locations of $(x, y)$. Some sample executions of the script are shown below.

```
>> LASTNAME_LAB6_TASK3
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.

>> LASTNAME_LAB6_TASK3
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.

>> LASTNAME_LAB6_TASK3
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.
```

As always, at the beginning of the script, make sure to clear the MATLAB Workspace as well the contents of the MATLAB Command Window.

## Task IV: Submission Instructions

Upload the following to Blackboard Learn.

(a) **LASTNAME_LAB6_TASK1.m**.

(b) **LASTNAME_LAB6_TASK2.m**.

(c) **LASTNAME_LAB6_TASK3.m**