CS1112 Exercise 2 Name: _____ NetID: _____

When you have completed the exercise, show this sheet and any associated programs to your discussion instructor, who will record that you have completed the work. If you do not finish this exercise during class, you have until *Sunday, 9/4, at 9pm* to get your exercise checked off during *consulting hours* or during *TAs' office hours*.

If you have any questions, **ask** your TA or a consultant immediately! They are in the lab to help you learn the material.

1 Testing formatting operators

Type the following lines of code into the Command Window in MATLAB (DO NOT COPY AND PASTE). Where you see `%____` , write the output produced by the code on the left.

```
val = input("Input a number: ");
fprintf('The value stored in val is: %f\n', val);    %___________________________________
fprintf('The value stored in val is: %d\n', val);    %___________________________________
fprintf('The value stored in val is: %e\n', val);    %___________________________________
fprintf('The value stored in val is: %g\n', val);    %___________________________________
fprintf('The value stored in val is: %c\n', val);    %___________________________________
```

If the last line prints something unexpected, why does it do that?

---

2 Temperature conversion **(modified from *Insight* M1.1.3)**

Download the file `convertCel2Fah.m` from the Exercises page and save it to your folder. Read and run the program. Let's change the presentation of the result by modifying the last statement in the program:

1. Change the *substitution sequence* from `%.3f` to `%.8f`. The substitution sequence is also called the *format sequence*. Save and run the program again and notice that the format of the number printed has changed. What does the substitution sequence `%.8f` specify? _____

2. Say, you want to use 10 character spaces for printing the entire value (including the decimal point) with 2 decimal places shown. You will then use the substitution sequence `%10.2f`. Again, make this change and observe the print format.

3 Minimum of a quadratic

Download and review `Eg1_2` in *Insight*.
http://www.cs.cornell.edu/courses/cs1112/ → Course info → Textbook extras → Code and Data → Eg1_2.m

[M1.2.5 from *Insight*] Modify `Eg1_2` to first check that L is less than or equal to R. If $R < L$, switch the values of R and L.

[M1.2.6 from *Insight*] Reorder the three branches of the conditional statement to first check whether the critical point is inside the interval.

4 Triangle

[Modified from M1.2.4 of *Insight*] The three interior angles of any triangle add up to 180°. Complete the program fragment below to print scalene, isoceles, or equilateral given three angles.

```
% Assume a, b, and c are positive integers representing angles in degrees that sum to 180

if (_____________________________________)
   disp('Scalene triangle')

elseif (_____________________________________)
   disp('Equilateral triangle')
else
   disp('Isoceles triangle')
end
```

5 Which quadrant?

Write two different programs to determine in which quadrant a user-input value of A degrees belongs. Assume that the user may enter any non-negative number. For example, 725° is the same, and should be treated, as 5°. (Hint: the function `rem` that you saw last week might be useful.) To avoid ambiguity, we use the following convention:

$$\text{Quadrant is } \begin{cases} 1 & \text{if } \ 0 \le A < 90 \\ 2 & \text{if } \ 90 \le A < 180 \\ 3 & \text{if } \ 180 \le A < 270 \\ 4 & \text{if } \ 270 \le A < 360 \end{cases}$$

Print which quadrant the user-input belongs. In the first script use four *separate* `if` statements (4 separate `if-end` constructs—no `else` or `elseif`) and call the program `angle1.m`. In the second script, use a *single* `if-elseif-...-else-end` construction for the evaluation and call it `angle2.m`.