CS/INFO 1305 Programming Exercise 1
Due Monday, July 25, at 9pm

1 The Earth, an oblate spheroid (a *what*?)

→ A sphere flattened at the poles. Another example is the m&m candy.

The surface area of a sphere is $4\pi r^2$ where r is the radius. Different variations to this formula can be used to estimate the surface area of an oblate spheroid like the Earth (or the amount of magic coating on an m&m!):

1. $4\pi r_1 r_2$
2. $4\pi((r_1 + r_2)/2)^2$

where r_1 is the equatorial radius and r_2 is the polar radius. Write a program to solicit the two radii of an oblate spheroid, calculate and display the area estimates based on the above formulae, and display the difference between the two estimates. Save the file as `oblateArea.m`.

2 Function evaluation

Write three different programs (scripts) to determine in which quadrant a user-input value of A degrees belong. 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` might be useful.) To avoid ambiguity, we use the following convention:

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

Print the result (quadrant number). In the first script use four *separate* `if` statements (4 separate `if-end` constructs) 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`. In the third script, use *nesting* without using the `elseif` clause and call it `angle3.m`. Pay close attention to the differences among the three programs.

3 Golden rectangle

The *golden ratio* $\phi = (1 + \sqrt{5})/2$ is one of the most interesting numbers in all of mathematics. For example, the ancient Greeks regarded an L-by-W rectangle with $L/W = \phi$ as the most aesthetically appealing rectangle.

Write a script `goldenRect` that randomly generates two values in the range of 1 to 9 to represent the side lengths of a rectangle. Assign the larger value to L and the other value to W (it's OK if the randomly generated values are the same). If the ratio L/W is within $\phi \pm 0.2$, draw the rectangle in red. If the ratio is really "unappealing," i.e., greater than 3, do not draw the rectangle. Otherwise draw the rectangle in yellow. Use the given function `DrawRect` (see Lab 1). Use the following statements to set up your figure window before drawing the rectangle (similar to `drawDemo` in Lab 1):

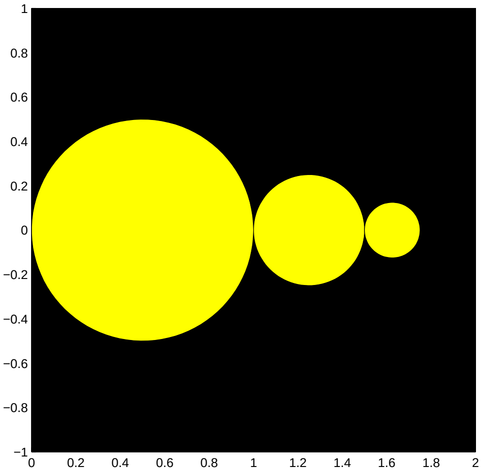
```
close all       % close all previous figure windows
figure          % open a figure window
axis equal off  % use equal scaling in the x- and y-axes; hide the axes
```

How to generate a random value? The statement `v= rand` assigns to variable `v` a random number in the range of 0 to 1. So how do you get a random number within a different range? First, the statement `v= rand` gets you a real number in the range of 0 to 1. Next, scale (think multiply) and shift (think add) the value `v` to get the range you want.

4 Apology to Zeno

Zeno of Elea (ca. 490 BC ca. 430 BC) proposed a series of paradoxes that confounded philosophers for a long time. The dichotomy paradox can be stated as follows: Suppose I want to get to the door. Before I can get to the door, I must first get halfway there. Before I can get halfway there, I must go a quarter of the way. Before I can go a quarter of the way, I must go an eighth of the way, and so on. This means I can never get to the door because 1, 1/2, 1/4, 1/8, ... is an infinite series—there is an infinite number of tasks to accomplish.

If I draw a disk of diameter 1, followed by an adjacent disk of diameter 1/2, followed by an adjacent disk of diameter 1/4, and so on, eventually I will not be able to see the disks anymore because of screen resolution. Write a script to explore this idea—find out how many *visible* disks can be drawn by experimentation (trial and error). Submit a script called `zenoDisks` that draws only the sequence of visible disks—no extra (useless) calls to function `DrawDisk`. Below is a figure showing only the first three disks in the sequence.

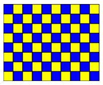


5 Stars, disks and rectangles

Write a script `myPoster` that uses the given functions `DrawRect`, `DrawDisk`, and `DrawStar`, to create a figure of your choice! The only requirement is that the script must involve the (meaningful) use of a `for`-loop. It is not necessary to use all three draw functions but you should use at least one. Have fun!

Below are challenge questions for fun, not for credit. Enjoy!

Challenge 1: Write a script that draws `m`-by-`n` unit squares in a checker pattern. Solicit `m` and `n` from the user and assume they are positive integers.



Challenge 2: Watch a brief video on the `while`-loop:

`http://fit.cit.cornell.edu/fit108v2/?cmd=pass&l=2&c=3&v=53`

You may need to log in using your Cornell NetID and password (look for "Cornell community single sign-on" on the right side of the screen"). Then write a script `stepPyramidSkeleton.m` to draw a step pyramid. The base rectangle is L-by-H where $H \leq L$ are user-entered values. Each step has the same height H. The next rectangle up is 2/3 the length of the rectangle below, and so forth. The top step must have a length no less than H. Use function `DrawRect` and use a `while`-loop.