CS1112 Exercise 12 Name: _____ NetID: _____

You have until Tuesday, 11/15, at 9pm to submit the completed class definition in problem 1 on MATLAB Grader. Problem 2 is extra practice for prelim 2 and does not need to be submitted.

1 Class `Interval`

1.1 Download the file `Interval.m` from the *Exercises* page. Let's play with some `Interval` objects in the Command Window:

```
format compact          % Set Command Window to single spacing
a = Interval(3,7)       % See in Workspace pane that the class of a is Interval.  Read
                        %   Interval.m to see how properties were declared.
disp(a.left)            % Access the left property using dot notation; should be 3
disp(a.right - a.left)  % Should be 4, the interval's width
a.shift(10)             % Call a's shift method to shift interval a to the right by 10
                        %   units.  Method shift doesn't return a value (see method
                        %   definition in Interval.m), so you do not see anything displayed
                        %   in Command Window
disp(a)                 % Display interval a now: _________________________
b = Interval(9,15);
g = a.isIn(b)           % Is interval a in interval b? ____________
                        %   Read method isIn.  Ask if you have any questions.
h = b.isIn(a)           % Is interval b in interval a? ____________
```

Observations: To access an *instance variable* (property), the syntax is ReferenceName.VariableName . To access an *instance method* (method defined inside a classdef for each object), the syntax is ReferenceName.MethodName(args for 2nd through last parameters) .

1.2 Implement method `getWidth` in class `Interval` as specified. Then try the following in the Command Window:

```
clear all           % In older versions of MATLAB, need to clear objects made using the old
                    %   class definition before using an updated class definition.  Not
                    %   necessary in the latest version of MATLAB.
a = Interval(3,7);
w = a.getWidth()    % w should be 4
```

1.3 Read method `scale` in class `Interval`. Revise `scale` to make effective use of method `getWidth`. Next try the following in the Command Window:

```
a.scale(2)          % Note that method scale does not return anything
disp(a)             % a should be (3,11)
```

1.4 Implement method `add` in class `Interval` as specified. In interval arithmetic, the sum's left end is the sum of the two original left ends; the sum's right end is the sum of the two original right ends. Next try it out in the Command Window:

```
b = Interval(0,2);
c = a.add(b)        % c should be (3,13)
```

Do you understand everything so far? If not, ask for help!

1.5 Above, MATLAB's built-in `disp` function was used to display the properties of an object. We can *override* the built-in method to display what *we* want to see for an object of class `Interval`! To do so, we simply implement a `disp` method inside the classdef of `Interval`. This was done but commented out. *Un*comment the `disp` method in class `Interval` now, save the file, and type the following code in the Command Window:

```
x = Interval(3,7) % What is displayed? _____________________________
                  % Since you didn't use a semicolon to end the assignment statement,
                  %   Matlab called the disp method to display x.  Since x is of type
                  %   Interval and class Interval has its own disp method, that specific
                  %   disp method was used instead of the built-in disp.
```

Copy the contents of your completed file `Interval.m` into the code box for Problem 1 in MATLAB Grader. Test (and correct if necessary) your class definition.

2 Images and uint8 arithmetic

In class you have worked with images in the RGB colorspace, where channel 1 is R (red), channel 2 is G (green), and channel 3 is B (blue). But images can be represented in other colorspaces. Consider the YCoCg colorspace, where channels Y, Co, and Cg are mathematically related to R, G, and B via the following equations:

$$Y = \frac{1}{4}(2G + R + B)$$
$$Co = \frac{1}{2}(R - B) + 128$$
$$Cg = \frac{1}{4}(2G - R - B) + 128$$

(This assumes the R, G, and B are in the range 0-255 and will yield Y, Co, and Cg in that same range.)

Implement the following function (remember that variables of type `uint8` will only ever store values 0 to 255).

```
function ycocg = rgb2ycocg(rgb)
% Transform image from RGB colorspace to YCoCg colorspace.
% rgb: 3D uint8 array such that rgb(i,j,k) is the value of channel
%      k (R, G, or B) for the pixel at row i and column j.
% ycocg: 3D uint8 array such that ycocg(i,j,k) is the value of
%        channel k (Y, Co, or Cg) for the pixel at row i and column j after
%        being transformed from RGB to YCoCg.
```