

CS1112 Discussion Exercise 11

1 Structure and structure array

Implement the following function for creating a *structure* for square data:

```
function Sqr= MakeSquare(x, y, L)
% Sqr is a square with
%   Sqr.x (x-coordinate of lower left corner) assigned x
%   Sqr.y (y-coordinate of lower left corner) assigned y
%   Sqr.length assigned L
```

Write a code fragment to create a *structure array* of length 5 where each component in the array is a structure containing square data. Make effective use of your function `MakeSquare`. Let all the x- and y-coordinates be random in (0,9) and let the k^{th} square have length k .

2 Revise methods in class Interval

During lecture we started developing the `Interval` class, which has properties `left` and `right` and methods `Interval` (the *constructor*, for constructing an `Interval` object) and `scale`. Add a method `getWidth` to return the width of an `Interval`. Once you have implemented `getWidth`, revise method `scale` to make use of `getWidth`.

```
classdef Interval < handle
% An Interval has a left endpoint and a right endpoint.
```

```
    properties
        left
        right
    end
```

```
    methods
        function Inter = Interval(lt, rt)
            % Constructor: construct an Interval object
            Inter.left= lt;
            Inter.right= rt;
        end
```

```
        function _____ = getWidth(self)
            % Return the width of the interval
            %%% Write your code below %%%
```

```
    end
```

```
    function scale(self, f)
        % Scale the interval by a factor f
        %%% Revise the code below %%%
```

```
        w= self.right - self.left;
```

```
        self.right= self.left + w*f;
```

```
    end
end %methods
end %classdef
```

3 Giving the Interval class more functionalities

Implement the methods specified below for class Interval.

```
classdef Interval < handle
% An Interval has a left endpoint and a right endpoint.

    properties
        left
        right
    end

    methods
        function Inter = Interval(lt, rt)
            % Constructor: construct an Interval object
            Inter.left= lt;
            Inter.right= rt;
        end

        function w = getWidth(self)
            % w is the width of the interval

        end

        function scale(self, f)
            % Scale the interval by a factor f
            w= self.getWidth();
            self.right= self.left + w*f;
        end

        function shift(self, s)
            % Shift the interval by s

        end

        function tf = isIn(self, other)
            % tf is true (1) if this Interval itself is in the other Interval

        end

        function Inter = add(self, other)
            % Inter is the resulting Interval formed by adding this Interval
            % itself to the other Interval

        end
    end %methods
end %classdef
```