Object-Oriented Programming

- First design and define the classes (of the objects)
  - Identify the properties (data) and actions (methods, i.e., functions) of each class
- Then create the objects (from the classes) that are then used, that interact with one another

Class Interval

- An interval has two properties:
  - left, right
- Actions—methods—of an interval include
  - Scale, i.e., expand
  - Shift
  - Add one interval to another
  - Check if one interval is in another
  - Check if one interval overlaps with another

To specify the properties and actions of an object is to define its class.
Executing an instance method

```matlab
classdef Interval < handle
    % An Interval has a left end and a right end
    properties
        left
        right
    end
    methods
        function Inter = Interval(lt, rt)
            % Constructor: construct an Interval
            Inter.left = lt;
            Inter.right = rt;
        end
        function scale(self, f)
            % Scale the interval by a factor f
            w = self.right - self.left;
            self.right = self.left + w*f;
        end
    end
end
```

```matlab
t = Interval(4,6);
```

```matlab
r = Interval(4,6);
r.scale(5)
disp(r.right) %What will it be?
```

```
4
14
```

Object is passed to a function by reference

```matlab
t = Interval(4,6);
r.scale(5)
disp(r.right) % updated value
```

```
177.54
```

Non-objects are passed to a function by value

```matlab
v = [2 4 1];
scale2(v,5)
disp(v) %NO CHANGE
```

```
2 4 10
```

Objects are passed to a function by reference

```matlab
v = [2 4 1];
scale2(v,5)
disp(v) %NO CHANGE
```

```
2 10 10
```

Syntax for calling an instance method:

```matlab
<reference>.<method>(<arguments for 2nd thru last parameters>)
```

```matlab
p = Interval(3,7);
r = Interval(4,6);

yesno = p.isIn(r);
% Explicitly call p's isIn method

yesno = isIn(p,r);
% Matlab chooses the % isIn method of one % of the parameters.
```

Method to find overlap between two Intervals

```matlab
function Inter = overlap(self, other)
    % Inter is overlapped Interval between self % and the other Interval. If no overlap then % self is empty Interval.
    ...
end
```
The overlap’s left (OLeft) is the rightmost of the two original lefts
The overlap’s right (ORight) is the leftmost of the two original rights

No overlap if OLeft > ORight

function Inter = overlap(self, other)
% Inter is overlapped Interval between self and the other Interval. If no overlap then % self is empty Interval.

Inter= Interval.empty();
left= max(self.left, other.left);
right= min(self.right, other.right);
if right-left > 0
    Inter= Interval(left, right);
end
end

% Example use of overlap function
A= Interval(3,7);
B= Interval(4,4+rand*5);
X= A.overlap(B);
if ~isempty(X)
    fprintf('(%f,%f)
', X.left,X.right)
end

Overloading built-in functions

- You can change the behavior of a built-in function for an object of a class by implementing a function of the same name in the class definition
- Called “overloading”
- A typical built-in function to overload is disp
  - Specify which properties to display, and how, when the argument to disp is an object
  - Matlab calls disp when there’s no semi-colon at the end of an assignment statement