• Previous lecture:
  – Objects are passed by reference to functions
  – Details on class definition (constructor, instance method)

• Today's lecture:
  – More practice with instance methods
  – Overriding methods
  – Array of objects
  – Methods that handle variable numbers of arguments

• Announcements:
  – Prelim 2 tonight 7:30pm
  – CALS Auditorium in Kennedy Hall (Rm 116)
  – Lab exercise problem 2 to be submitted on CMS by Monday 11/14, at 11pm.

Object is passed to a function by reference

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

Objects are passed to functions by reference. Changes to an object’s property values made through the local reference (self) stays in the object even after the local reference is deleted when the function ends.

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.
```

Syntax for calling an instance method:

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

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

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

A class file has the name of the class and begins with keyword classdef:

```matlab
classdef interval < handle

properties
    left
    right
end

definition 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
```

Compare two intervals

1. `redRight < blueRight`
2. `redRight < blueRight`
3. `redRight < blueRight`
4. `blueRight < redRight`
5. `blueRight < redRight`
6. `blueRight < redRight`
Overriding 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 "overriding" (called "overloading" in Matlab documentation)
- A typical built-in function to override is `disp`
  - Specify which properties to display, and how, when the argument to `disp` is (a reference to) an object
  - Matlab calls `disp` when there's no semi-colon at the end of an assignment statement

MATLAB allows an array to be appended

```
v= [3 1 5 9]
v(7)= 4
```
- What happens to `v(5)` and `v(6)`?

```
3 1 5 9 0 0 4
```
- MATLAB assigns some “default value” to the skipped over components for simple, cell, and struct arrays.
- For arrays of objects, you must implement the constructor to handle such a situation

MATLAB allows an array to be appended

```
A = Interval(3,7);
A(2)= Interval(4,6); % Array of length 2
A(3)= Interval(1,9); % Array of length 3
A(5)= Interval(2,5); % Array of length 5
```
- Error!
  - Interval constructor we have so far requires two parameters:
  - User specified two arguments as required for A(5), but…
  - Matlab has to assign A(4) "on its own" by calling the constructor, but no arguments get passed → Error!

MATLAB allows an array to be appended

```
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
A= Interval(3,7);
B= Interval(4,4+rand*5);
X= A.overlap(B);
if ~isempty(X)
    fprintf('(%.2f,%.2f)
', X.left,X.right)
end
```
- Built-in function to create an empty array of the specified class

An "array of objects" is really an ...

- array of references to objects

```
A= Interval(3,7);
A(2)= Interval(4,6);
A(3)= Interval(1,9);
```
- Built-in function to create an empty array of the specified class
Constructor that handles variable number of args

- When used inside a function, \texttt{nargin} returns the number of arguments that were passed.
- If \texttt{nargin} \neq 2, constructor ends without executing the assignment statements. Then \texttt{Inter.left} and \texttt{Inter.right} get any default values defined under properties. In this case the default property values are \texttt{[]} (type \texttt{double}).

```matlab
classdef Interval < handle
properties
    left
    right
end
methods
    function Inter = Interval(lt, rt)
        if nargin==2
            Inter.left= lt;
            Inter.right= rt;
        end
    end
end
```

If a class defines an object that may be used in an array…

- Constructor must be able handle a call that does not specify any arguments
  - Use built-in command \texttt{nargin}, which returns the number of function input arguments passed.

- The overridden \texttt{disp} method, if implemented, should check for an input argument that is an array and handle that case explicitly. Details will be discussed next lecture.

---

A function to create an array of \texttt{Intervals}

```matlab
function inters = intervalArray(n)
    % Generate n random Intervals. The left and right ends of each interval is in (0,1)
    for k = 1:n
        randVals= rand(1,2);
    end
end
```

An independent function, not an instance method. See \texttt{intervalArray.m}

A function to find the widest \texttt{Interval} in an array

```matlab
function inter = widestInterval(A)
    % inter is the widest Interval (by width) in A, an array of Intervals
end
```

An independent function, not an instance method. See \texttt{widestInterval.m}