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

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

• Announcements:
  – Discussion this week in computer lab
  – Prelim 2 tonight 7:30pm
    • Kennedy Hall Auditorium (Rm 116)
  – Lab exercise problem 4 to be submitted on CMS by Sunday 5/1, at 11pm. You may work in a pair, both partners must be enrolled in the same dis section.

**classdef syntax summary**

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

```matlab
classdef classname < handle
  Properties
  % An Interval has a left end and a right end
  Properties
  Instance methods
  Constructor returns a reference to the class object
  Methods:
  scale(self, f)
  % Scale the interval by a factor f
  
end
end
```

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

```matlab
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

An “array of objects” is really an ...

array of references to objects

```matlab
A = Interval(3,7);  % Array of length 1
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:
  ```matlab
  function Inter = Interval(lt, rt)
  % Constructor: construct an interval of length the
  % Inter.left = lt; 
  % Inter.right = rt;
  end
  ```
• 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!
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
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 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
```

A function to find the widest Interval in an array

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

A weather object can make use of Intervals...

- Define a class \texttt{LocalWeather} to store the weather data of a city, including monthly high and low temperatures and precipitation
  - Temperature: low and high \rightarrow an \texttt{Interval}
    - For a year \rightarrow length 12 array of \texttt{Interval}
  - Precipitation: a scalar value
    - For a year \rightarrow length 12 numeric vector
  - Include the city name: a string

```matlab
classdef LocalWeather < handle
    properties
        city
        temps  % array of Intervals
        precip % numeric vector
    end
    methods
        ...
    end
end
```

Weather data file

```plaintext
//Ithaca
//Monthly temperature and precipitation
//Lows (cols 4-8), Highs (col 12-16), precip (cols 20-24)
//Units:  English
15 31 2.08
17 34 2.06
23 42 2.64
34 56 3.29
44 67 3.19
53 76 3.99
58 80 3.83
56 79 3.63
48 71 3.69
NaN 59 NaN
32 48 3.16
22 36 2.40
```

See \texttt{ithacaWeather.txt}, \texttt{LocalWeather.m}
classdef LocalWeather < handle
    properties
        city = '';  temps = Interval.empty();  precip = 0;
    end
    methods
        function lw = LocalWeather(fname)
            fid = fopen(fname, 'r');
            s = fgetl(fid);
            lw.city = s(3:length(s));
            for k = 1:3
                s = fgetl(fid);
            end
            for k = 1:12
                s = fgetl(fid);
                lw.temps(k) = Interval(str2double(s(4:8)), ...
                    str2double(s(12:16)));
                lw.precip(k) = str2double(s(20:24));
            end
            fclose(fid);
        end
    end %methods
end %classdef

Function to show data of a month of LocalWeather

function showMonthData(self, m)
    % Show data for month m, 1<=m<=12.

end

Observations about our class Interval

- We can use it (create Interval objects) anywhere
  - Within the Interval class, e.g., in method overlap
  - “on the fly” in the Command Window
  - In other function/script files – not class definition files
  - In another class definition
- Designing a class well means that it can be used in many different applications and situations