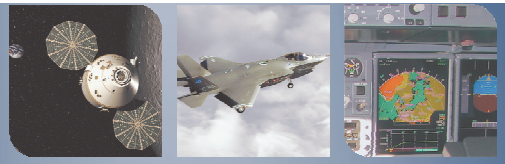
The MathWorks™ MATLAB®&SIMULINK®

Introduction to Object-Oriented Programming in MATLAB®

Jos Martin

Principle Software Engineer

jos.martin@mathworks.co.uk



MathWorks
Aerospace and Defence Conference '08

Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The *MATLAB* class system

What is a program?

Code

```
x = 12
while (x < 100)
    x = x+1
    if ( x == 23)
        disp('Hello')
    end
end
```

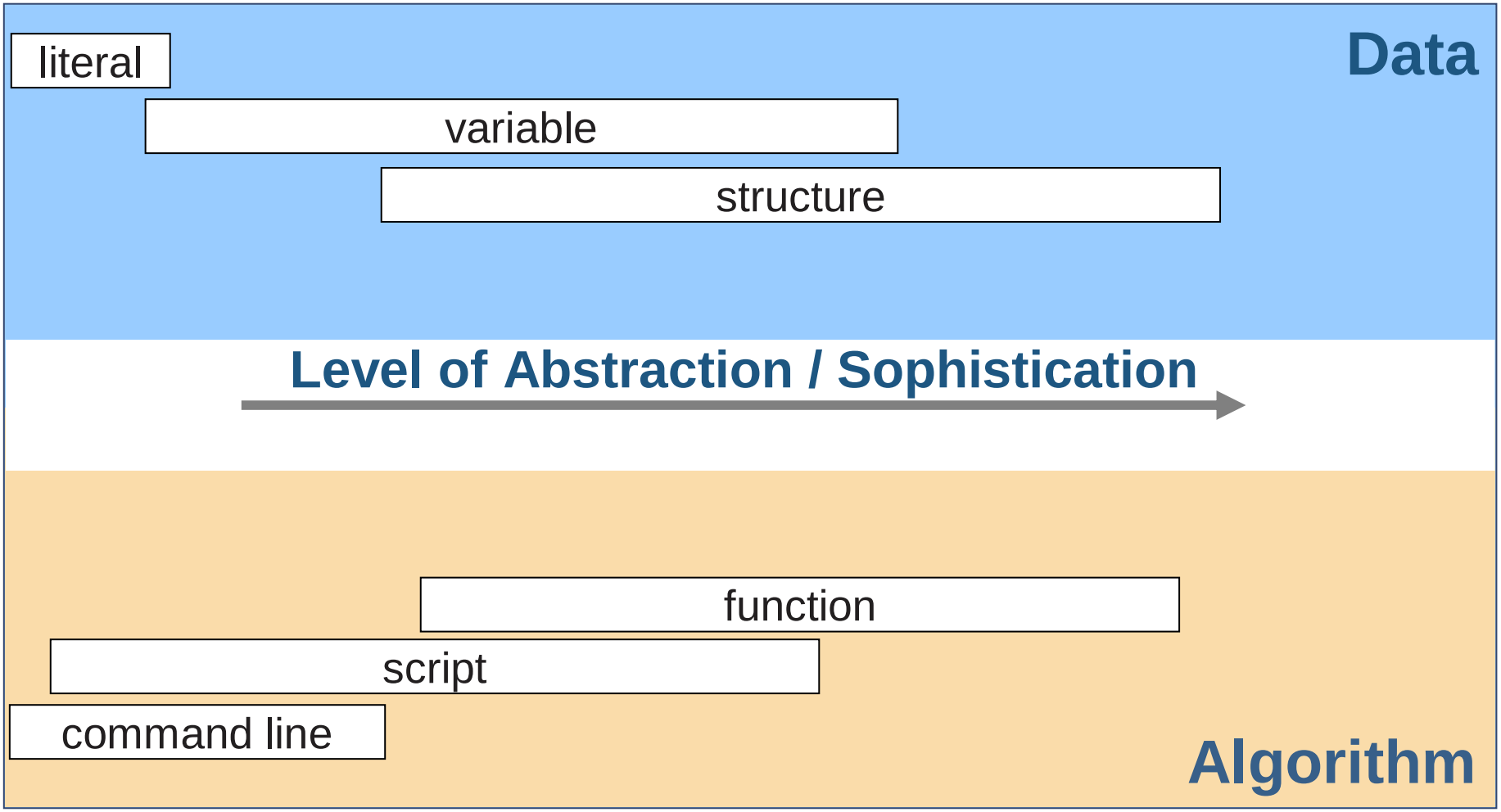
Data

```
x = 12
while (x < 100)
    x = x+1
    if ( x == 23)
        disp('Hello')
    end
end
```

Algorithm

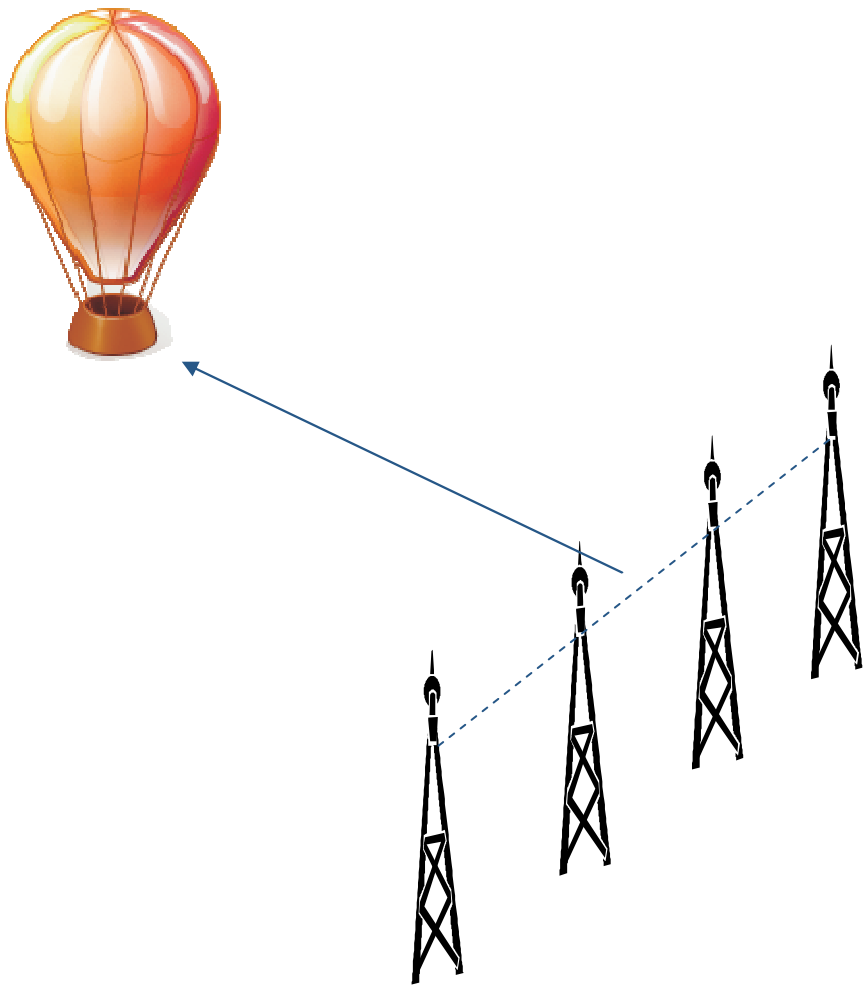
```
Assignment
Looping Test
    Increment
    Test to Act
        Take Action
    End
End
```

Progression of Programming Techniques



Example: Sensor Array

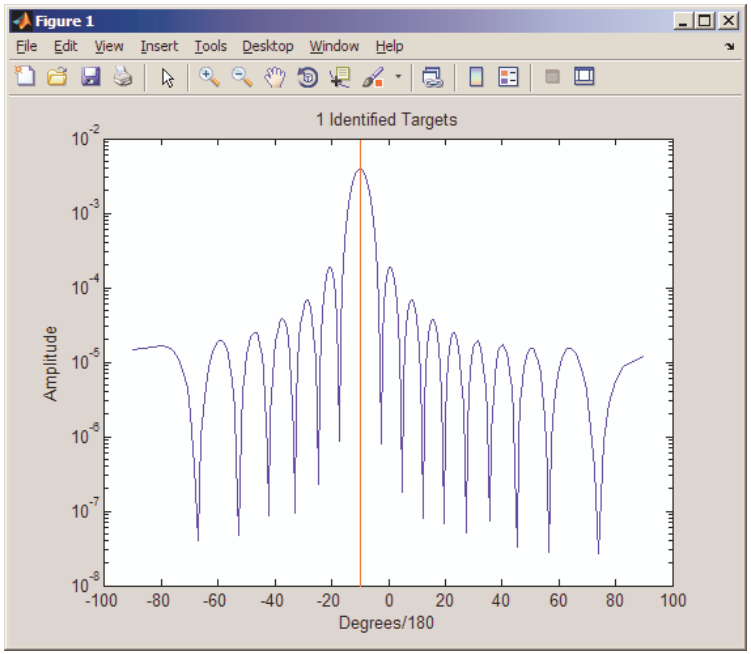
- Transmitting a signal from a weather balloon
- Locating the signal with a sensor array
- Computing the angle of arrival for the signal (AoA)



Procedural Programming

- Easy to learn
- Minimal planning

- No formal relationship between data and functions
- Every detail is exposed



Data and Actions to Implement

Wavelength

Number

Spacing

Location

Reading

Frequency

Determine peaks

Compute FFT

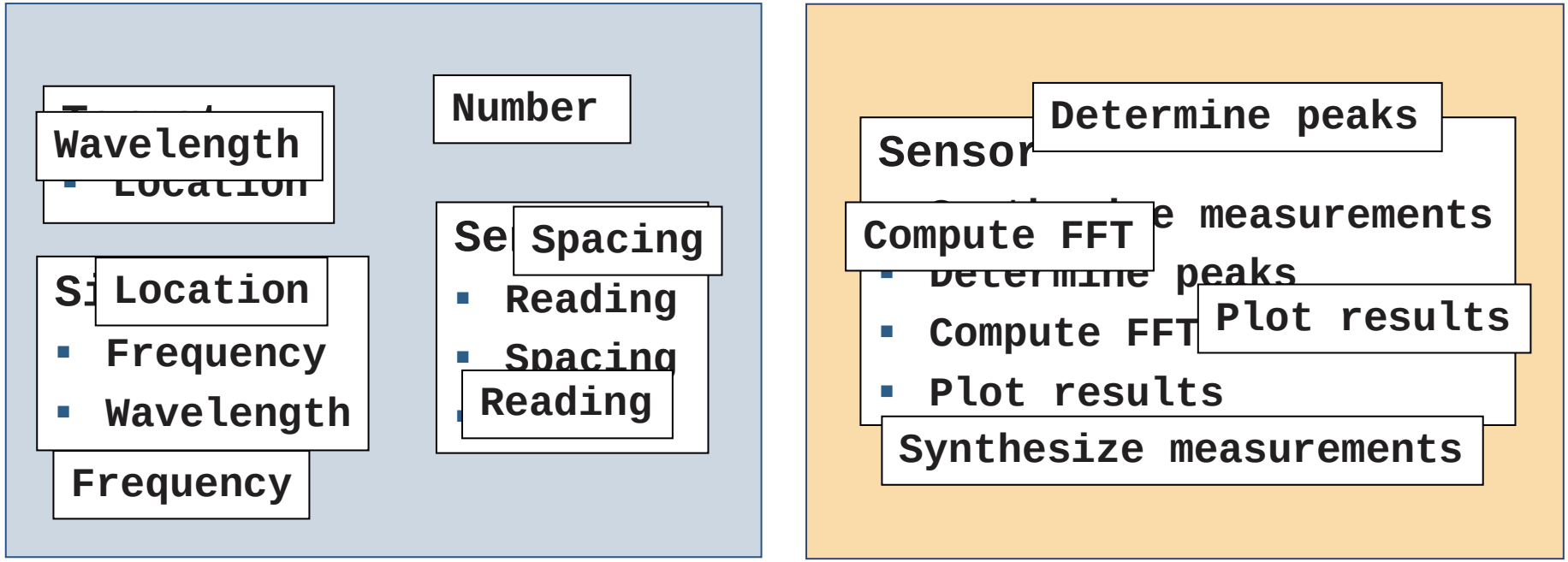
Plot results

Synthesize measurements

Data

Actions

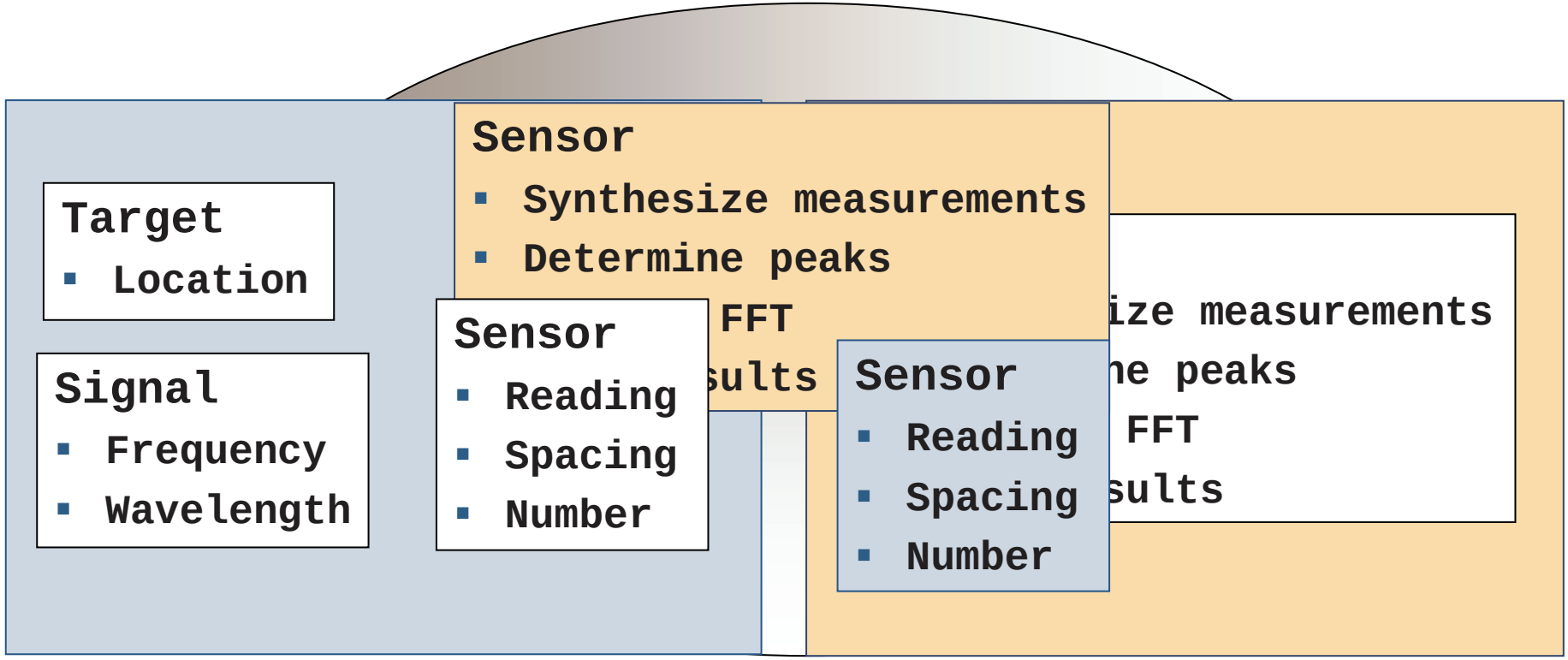
Related Data and Actions



Data

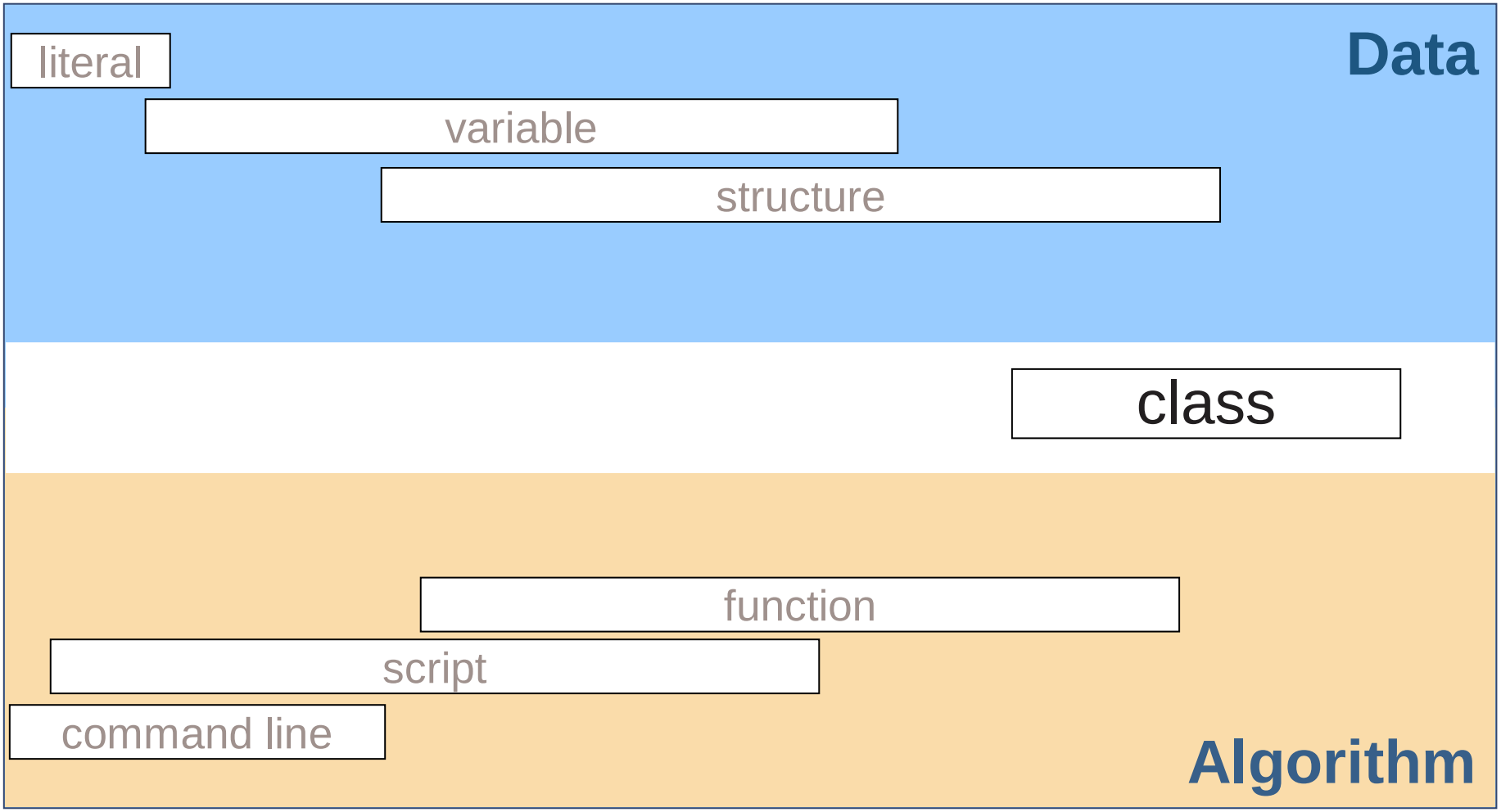
Actions

Grouping Related Items



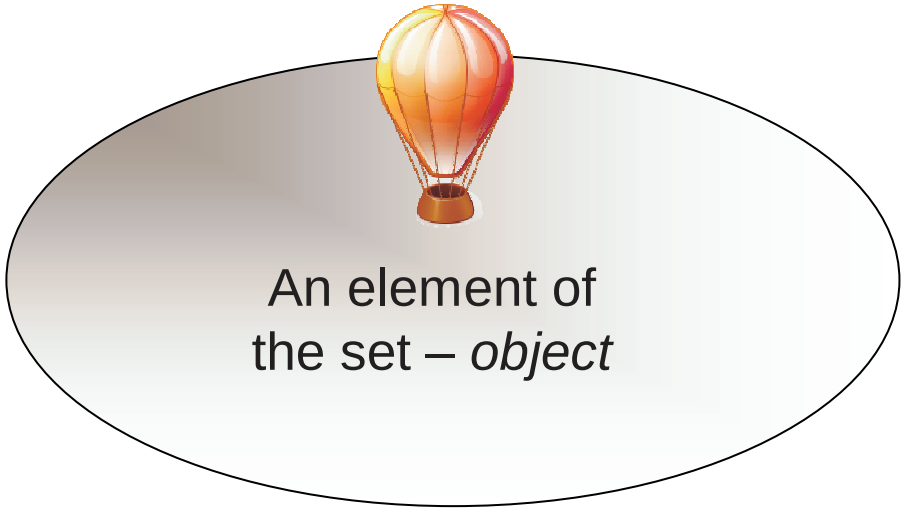
Data Class Actions

Progression of Programming Techniques



Object-Oriented Terminology

- Class
  - Blueprint of an idea
  - *Properties* (data)
  - *Methods* (algorithms)
- Object
  - Specific example of a *class*
  - *Instance*



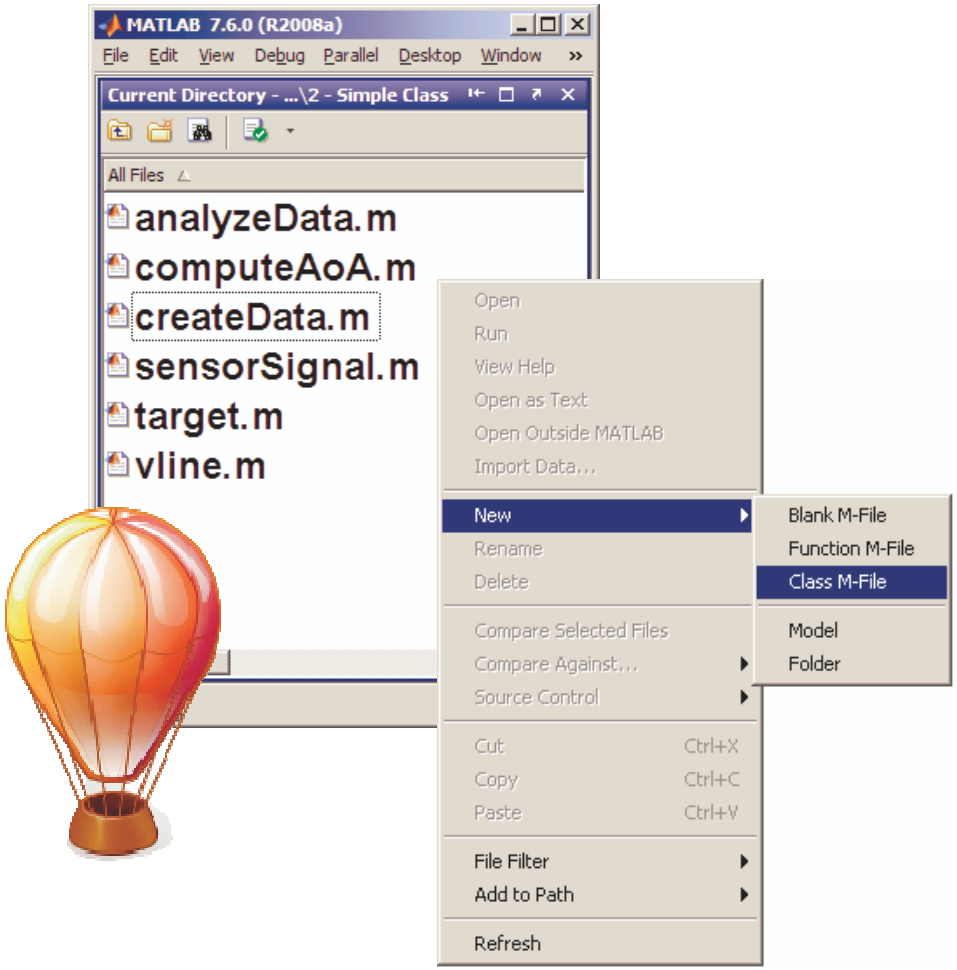
Defined set – *class*

Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The *MATLAB* class system

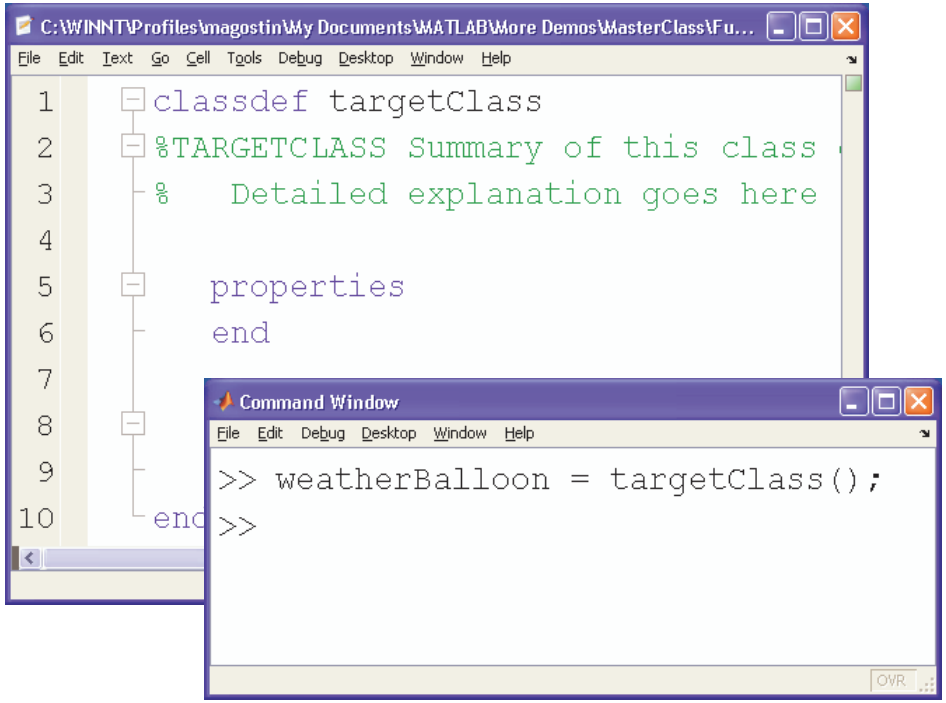
Demonstration: Building a Simple Class

- Define a target *class*
- Create the weather balloon *object*
- Use the *object* in place of the structure



Objects

- Easy to create
- Manage their own data
- Interchangeable with a structure
  - No other code changes required
  - *Properties* behave similar to field names
  - Can't add fields arbitrarily

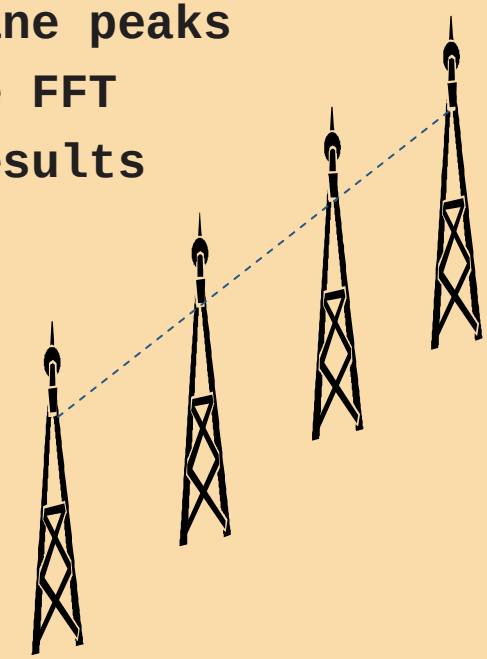


Demonstration: Adding Methods to a Class

- Start from a sensor *class* with existing *properties*
- Add a *method* to compute angle of arrival (AoA)
- Integrate a sensor *object* into the existing code

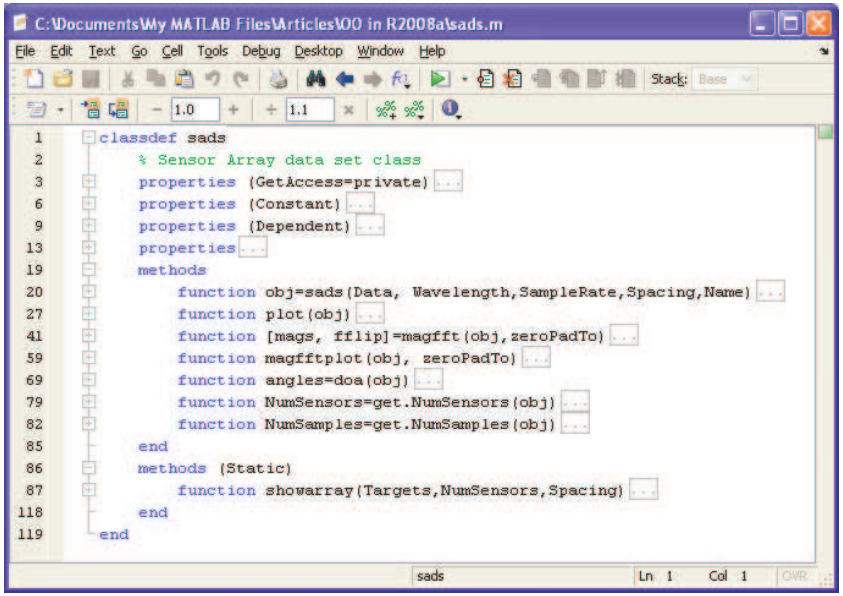
Sensor

- Synthesize measurements
- Determine peaks
- Compute FFT
- Plot results



Objects with Methods

- Have immediate access to their own data (*properties*)
- Allow you to overload existing functions
- Allow you to perform custom actions at creation and deletion



Goals

- Object-oriented programming
- Basic syntax in MATLAB®
- The *MATLAB* class system

The MATLAB Class System

- Designed to ‘feel’ like MATLAB
  - Incorporates matrix indexing

```
>> x = 2*obj.data(1:end);
```

  - Inherent overloading

```
varargout = obj.function(varargin)
```

- Works like an object-oriented language
  - Encapsulation, inheritance, polymorphism, etc.

Taking Methods and Properties Further

- Control access
- Create constants
- Make values interdependent
- Execute methods when properties change

External Methods
- Plot results
- Compute AoA

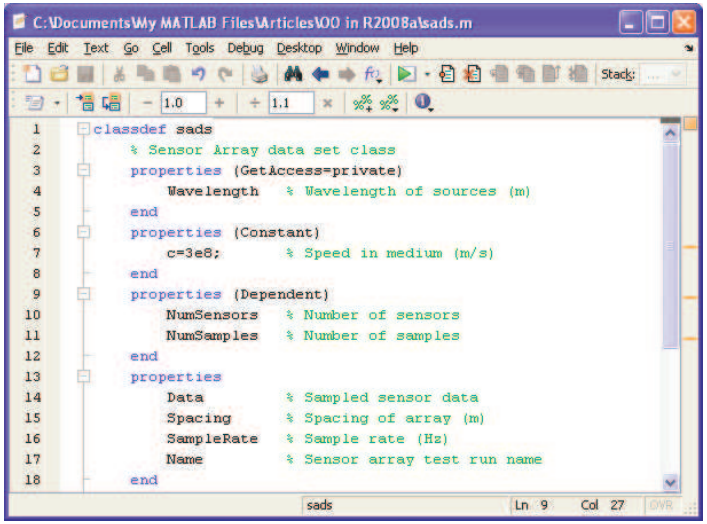
Internal Methods
- Synthesize measurements
- Determine peaks
- Compute FFT

External Data
- Reading
- Spacing
- Number

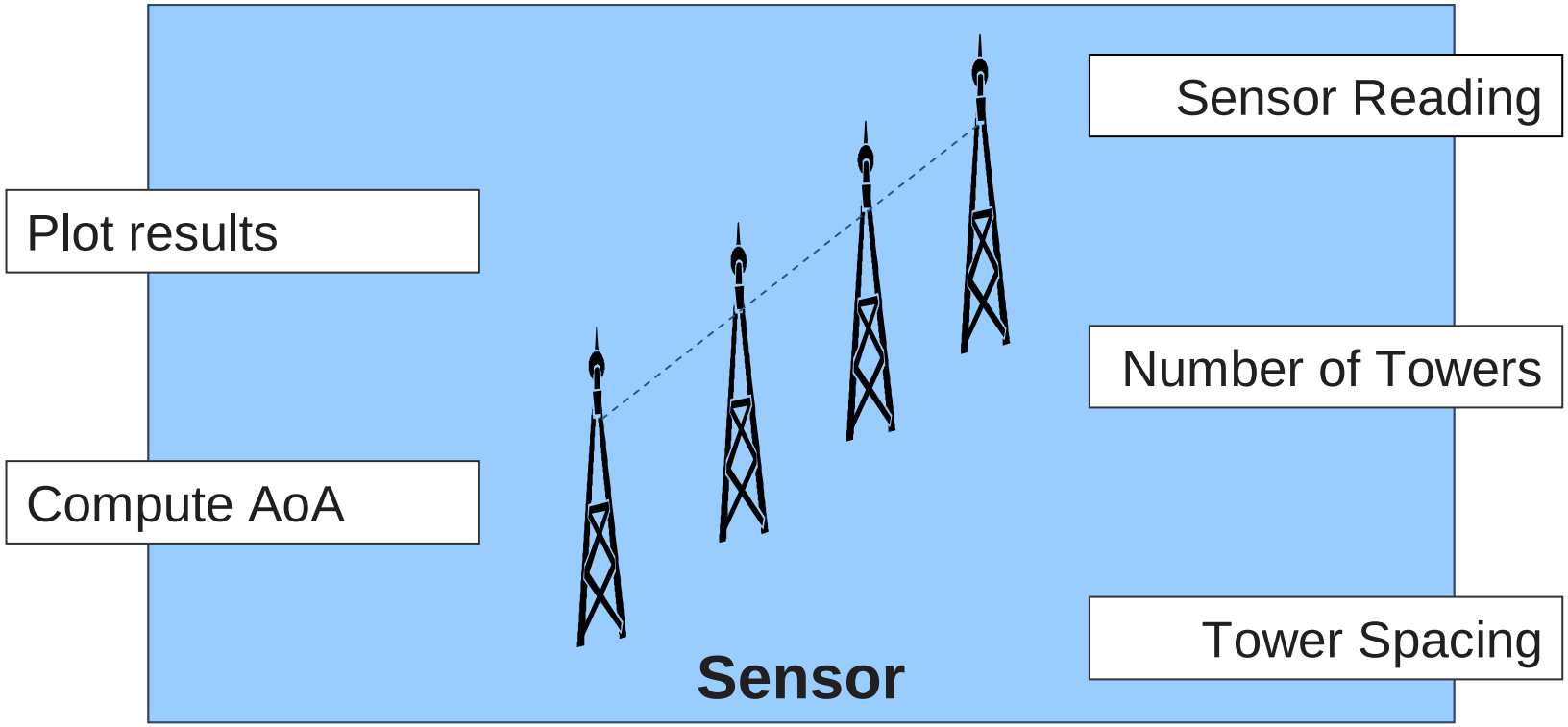
Internal Data
- Speed of light
- Noise ratio
- etc.

Demonstration: Applying Attributes

- Control access
  - `Access = public`
  - `Access = protected`
- Restrict modification
  - `Constant`
  - `Dependent`

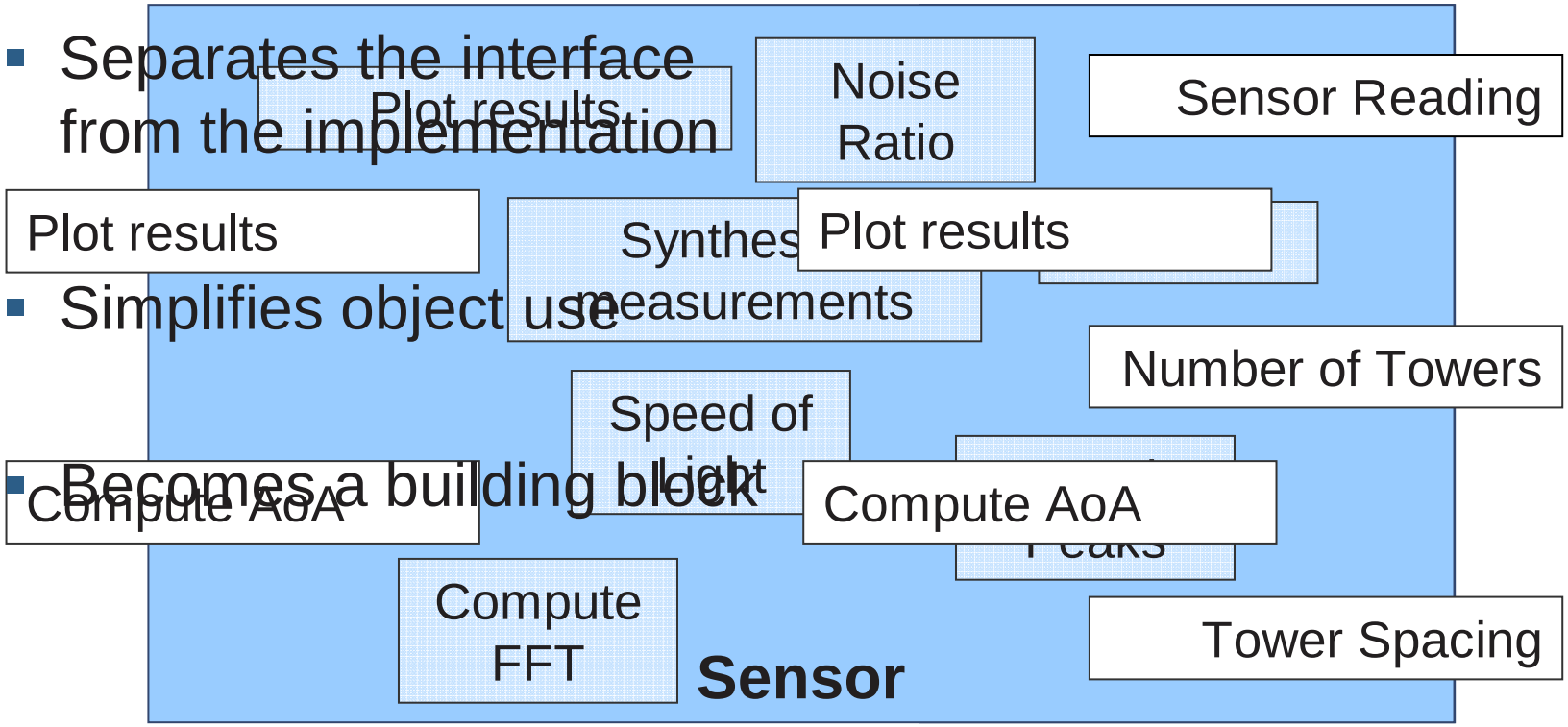


Encapsulation

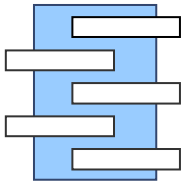


Encapsulation

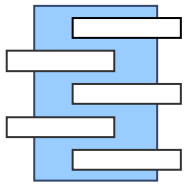
- Separates the interface from the implementation
- Simplifies object use
- Becomes a building block



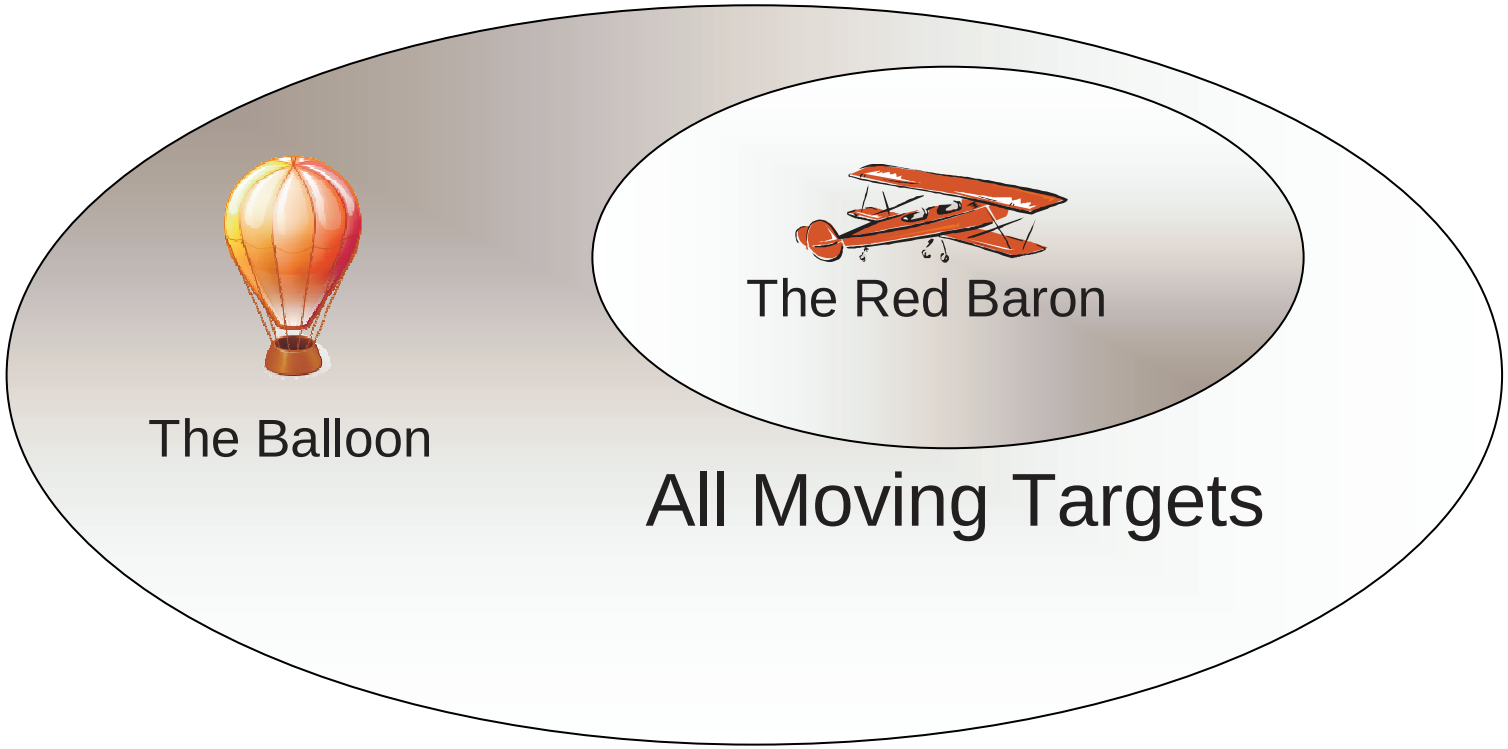
Using an Object as a Building Block



```
Assignment
Looping Test
    Increment
    Test to Act
        Take Action
    End
End
```



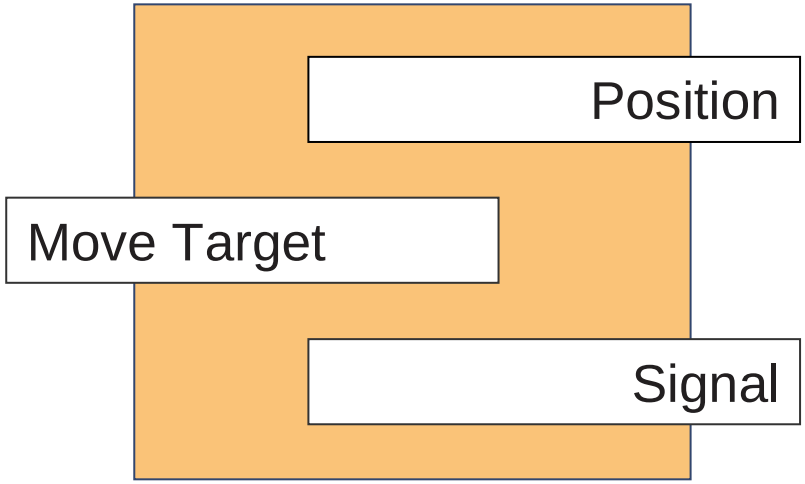
Using a Class as a Building Block



All Targets

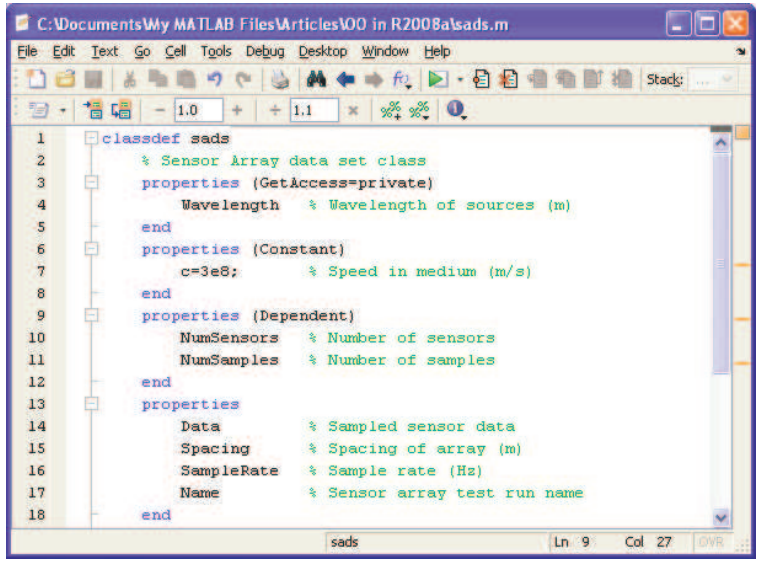
Demonstration: Creating a Moving Target

- Define a new *class* moving target
- *Inherit* from the existing *class* target
- Add a *method*
- Use the moving target



Inheritance

- *Subclass* substitutes for the *superclass*
- Allows re-envisioning and re-implementing the *superclass*
- Builds on proven code
- Allows inheriting from the base MATLAB classes



How does '=' work in MATLAB?
Round 1

```
>> a = 10000;
>> b = a;
>> b = 20000;
>> disp(a)
```

a) 10,000

b) 20,000

c) Something else

d) No idea

How does '=' work in MATLAB?
Round 2

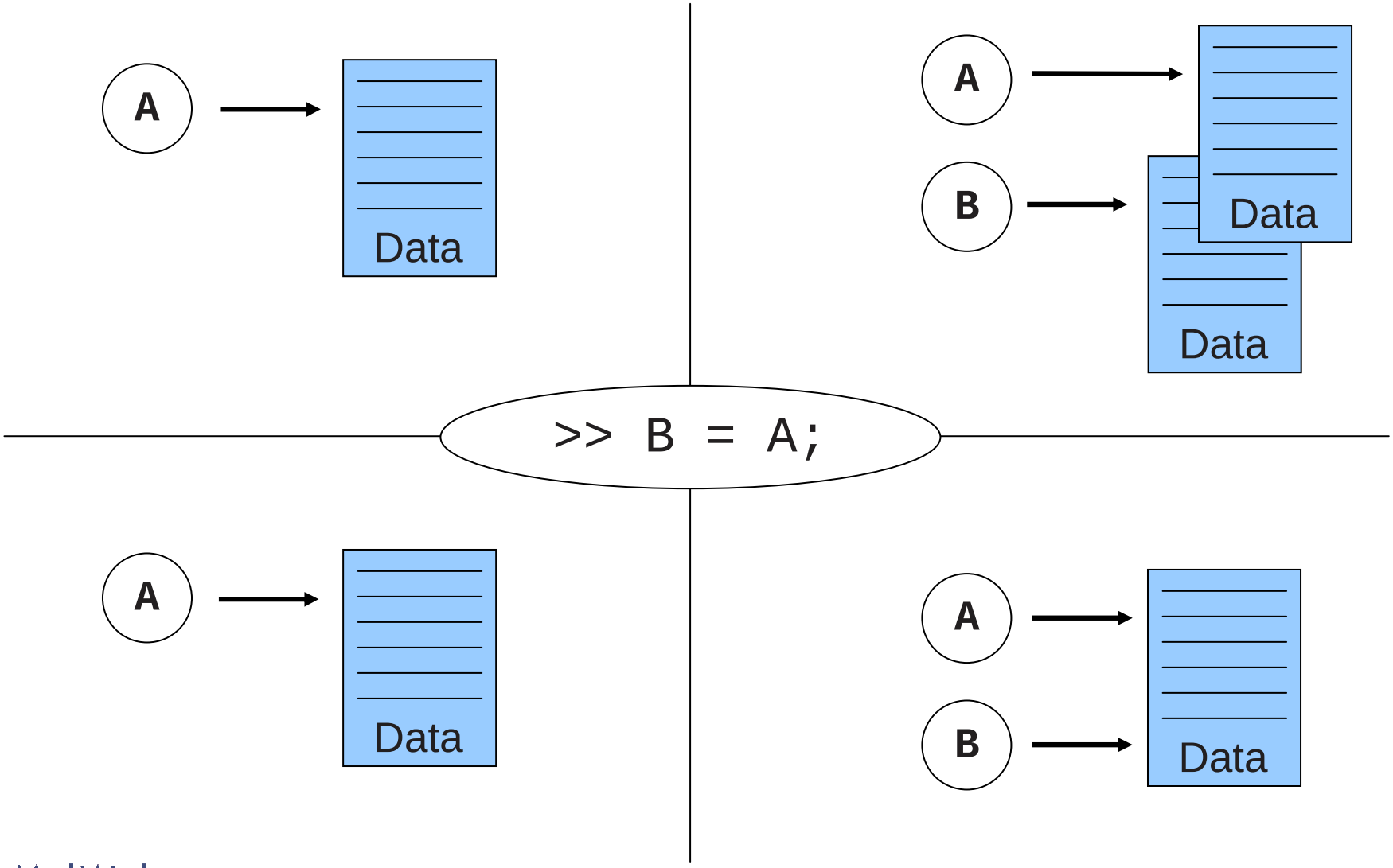
```
>> a = analoginput('winsound'); addchannel(a,1);
>> a.SampleRate = 10000;
>> b = a;
>> b.SampleRate = 20000;
>> disp(a.SampleRate)
```

a) 10,000

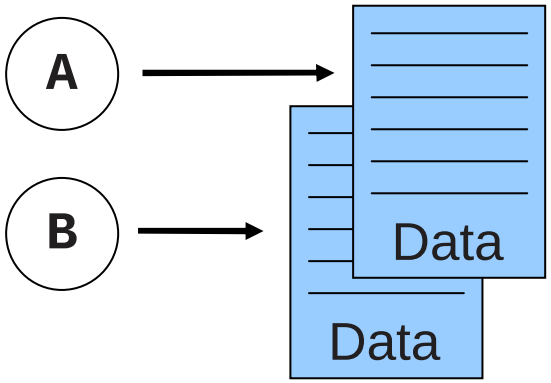
b) 20,000

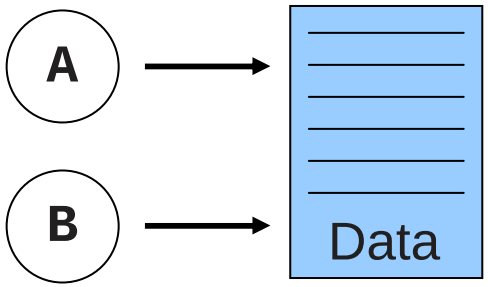
c) Something else

d) No idea


A
Data
A
B
Data
Data
>> B = A;
A
Data
A
B
Data

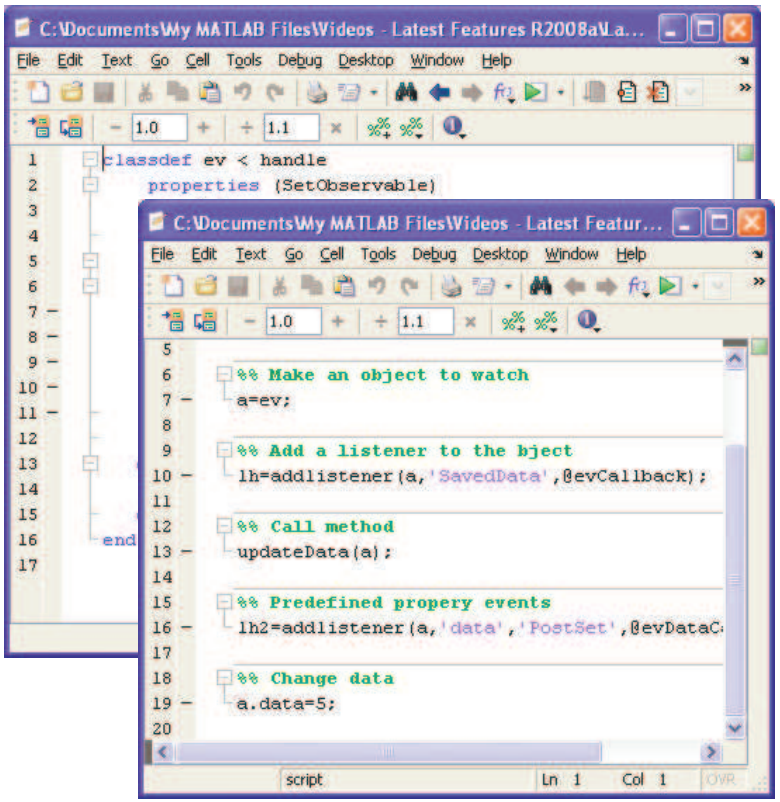
Value Class	Handle Class
MATLAB default	Use: `< handle`
'=' *copies* data	'=' *references* data
data in workspace	*handle* in workspace





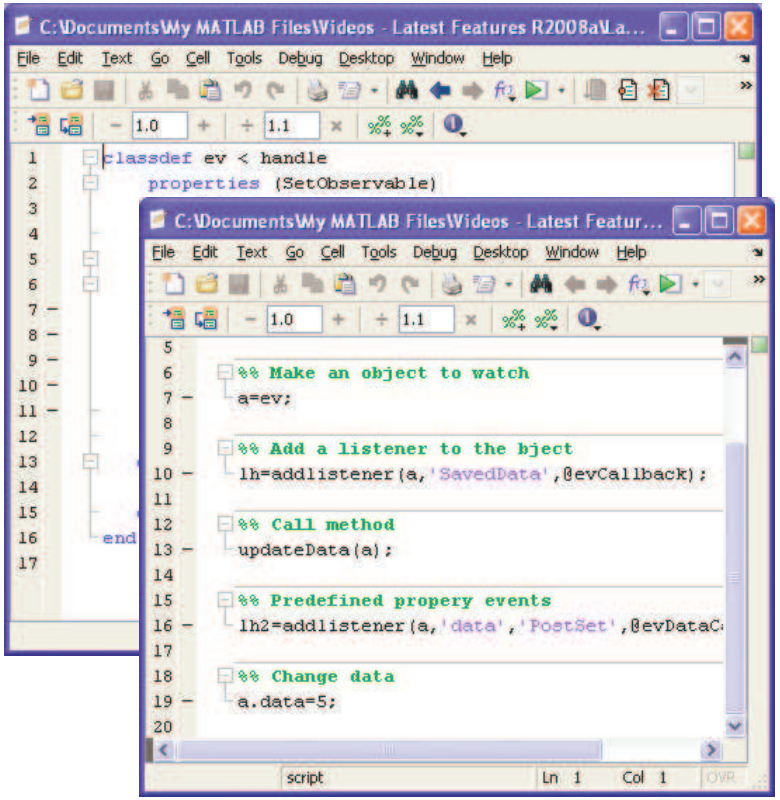
Optional Demonstration: Using Events

- Events
  - Created in a handle object
  - `events` block in `classdef`
  - `notify(…)` triggers event
- Listeners
  - Triggers call back function
  - `addlistener(…)`
  - Useable anywhere



Events and Listeners

- Uses technology related to
  - `preSet`
  - `postSet`
  - `preGet`
  - `postGet`
- Gives the ability to trigger action



- Anything can listen to an observable object

The MATLAB Class System

- Class definition file describes object behavior
- Objects can substitute for structures
- Apply attributes for a clean interface
- Build on existing classes with inheritance

Extends the matrix-based language to objects

Additional Resources

MATLAB® - The Language of Technical Computing - Mozilla Firefox

http://www.mathworks.com/products/matlab/

The MathWorks™
Accelerating the pace of engineering and science

Products & Services | Industries | Academia

MATLAB Overview
- Description
- Function List
- Demos and Webinars
- System Requirements
- Latest Features

Support & Training
- Product Support
- Documentation
- Downloads & Trials
- Training

Other Resources
- Technical Literature
- User Stories
- Related Books
- News and Events

MATLAB® - The Language of Technical C

MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran.

- Introduction and Key Features
- Developing Algorithms and Applications
- Analyzing and Accessing Data
- Visualizing Data
- Performing Numeric Computation
- Publishing Results and Deploying Applications
- NEW Object-Oriented Programming in MATLAB

FREE Interactive MATLAB CD & Tech Kit

MATLAB Add-On Products | MATLAB Appl

MATLAB is the foundation for Simulink® and all ot extended with add-on products for:

Math and Optimization

Done

MATLAB® - Object-Oriented Programming in MATLAB® - Mozilla Firefox

http://www.mathworks.com/products/matlab/object_oriente

Home | Select Country | Contact Us

The MathWorks™
Accelerating the pace of engineering and science

Products & Services | Industries | Academia | Support | User Commun

MATLAB Overview
- Description
- Function List
- Demos and Webinars
- System Requirements
- Latest Features

Support & Training
- Product Support
- Documentation
- Downloads & Trials
- Training

Other Resources
- Technical Literature
- User Stories
- Related Books
- News and Events

MATLAB® 7.6

Object-Oriented Programming in MATLAB®

The object-oriented programming capabilities of the MATLAB® language enable you to develop complex technical computing applications faster than with other languages, such as C++, C#, and Java™.

Using new capabilities in R2008a, you can define classes and apply standard object-oriented design patterns in MATLAB that let you benefit from code reuse, inheritance, encapsulation, and reference behavior without engaging in the low-level housekeeping tasks required by other languages.

Watch video on developing classes in MATLAB® (11 min)

Key Features
- Class definition files, enabling definition of properties, methods, and events
- Handle classes with reference behavior, aiding the creation of data structures such as linked lists
- Events and listeners, allowing the monitoring of object property changes and actions
- JIT-Accelerator support, providing significantly improved object performance
- Development environment support for the creation and use of classes

Learn About Object-Oriented Programming in MATLAB
- MATLAB Classes and Object-Oriented Programming (documentation)
- Sample code comparisons (MATLAB Central)
  - MATLAB and C++
  - MATLAB, C++, Java, Python®, and Ruby

Done

Questions and Answers