

**Object-Oriented MATLAB Programming**

**Course Number:** MTLB-106  
**Duration:** 1 day

**Overview**

This Object-Oriented MATLAB Programming training course teaches students how to create high-quality, maintainable MATLAB programs using the modern object-oriented programming paradigm, replacing the “spaghetti code” too commonly found in beginner-level MATLAB scripts. Students are led from object-oriented design basics to advanced features such as access attributes, encapsulation, property value validations, getter and setter methods, event notification and handling, backward compatibility across MATLAB releases, and run-time performance considerations.

**Note:** This course can be condensed to a half-day with fewer examples and less hands-on practice. We strongly recommend the full-day version if possible.

**Prerequisites**

Attendees should have taken Accelebrate's [From MATLAB Scripts to Complete Programs course](file:////training/matlab-scripts-programs) or have equivalent knowledge. Students should already be comfortable using the MATLAB environment and have some basic programming experience. No prior object-oriented experience is assumed or required for this course. However, familiarity with object-oriented constructs in other programming environments (e.g., C#, C++, Java, or Python) would be helpful.

**Materials**

All MATLAB training students will receive comprehensive courseware.

**Software Needed on Each Student PC**

* Any Windows, Linux, or macOS operating system
* A recent version of MATLAB

**Objectives**

* Master the fundamentals of object-oriented coding
* Write MATLAB classes
* Customize MATLAB class blocks using attributes
* Understand the differences between, and uses of, handle and value classes
* Write reusable, maintainable MATLAB classes
* Use class inheritance to override objects’ functionality and behavior
* Use class constructs to notify run-time events to subscribed callback listeners
* Use property validators and accessor methods
* Explore MATLAB's hidden object-oriented features
* Understand MATLAB release compatibility and run-time performance tradeoffs

**Outline**

* Introduction to MATLAB OOP
  + Comparing paradigms: OOP vs. procedural programming
  + Importance of OOP for development and maintainability
  + OOP use in MATLAB code
  + Benefits and drawbacks of using MATLAB OOP
  + MATLAB OOP’s historic evolution and future outlook
* Programming Object-Oriented MATLAB
  + Components of MATLAB OOP
  + The format of a MATLAB class
  + Offline & run-time inspection of class components
  + Constructor and destructor methods
  + Handle vs. value classes
  + Class packages, folders, files, and attributes
  + Handling accessibility attributes
  + Bypassing property restrictions (carefully!)
  + Dependent (on-the-fly) properties
  + Specifying and validating property types and values
  + Setter and getter methods for property values
* Advanced Object-Oriented MATLAB
  + Copying objects (deep/shallow copy)
  + Class inheritance
  + Overloading class components
  + Using static classes
  + Object pooling
  + The singleton design pattern
  + Notifying and listening to events
  + Using enumerated values
  + MATLAB releases compatibility considerations
  + Run-time performance aspects and tradeoffs
* Conclusion