

## Advanced MATLAB Application Development

SciEngineer's training courses are designed to help organizations and individuals close skills gaps, keep up-to-date with the industry-accepted best practices and achieve the greatest value from MathWorks® and COMSOL® Products.



### Advanced MATLAB Application Development

This two-day course provides a comprehensive introduction to the key principles and techniques of object-oriented application development using MATLAB®. The course is intended for MATLAB developers who will create medium- and large-scale applications for end users, as well as existing application developers looking for best practices and guidelines for structuring their code base. The course focuses on interactive application development, although the techniques are applicable to any large-scale application.

#### **Prerequisites**

MATLAB Programming Techniques and some experience of working with MATLAB graphics objects. Some existing knowledge of object-oriented programming in MATLAB or Object-Oriented Programming with MATLAB.

#### TOPICS

Day 1

- Organizing Application Data in a Model Class
- Managing Application Layout
- Visualizing Application Data with View Classes

#### Day 2

- Design Considerations for View and Controller Components
- Enabling Interactive User Control with Controller Classes
- Integrating Components with an Application Launcher



#### Organizing Application Data in a Model Class

## Managing Application Layout

<u>OBJECTIVE:</u> Create a model class to act as a data repository for an application and store the state of the system. Define and implement the core model responsibilities and actions.

OBJECTIVE: Create, access, and modify graphics objects within the MATLAB graphics hierarchy. Use predefined graphics callback functions. Use layout management objects to organize the appearance of the application's graphics objects and controls.

- Exploring models within the MVC framework
- Avoiding unnecessary data copies
- Storing application data
- Providing access to data for visualization
- Providing mechanisms for control operations on the data
- Communicating changes of state or data

• Navigating the MATLAB graphics hierarchy

- Creating, accessing, and modifying graphics and control objects
- Working with commonly used graphics objects
- Organizing application components using layout management objects
- Using predefined graphics callback functions

#### Visualizing Application Data with View Classes

<u>OBJECTIVE</u>: Define view classes for visualizing application data. Define and implement the core view responsibilities and actions.

- Views within the MVC framework
- The relationship of a view to a model
- The relationship of a view to its graphics objects
- Dynamic updates to views using listeners and callbacks
- The interaction between a model and its views

### **Design Considerations** for View and Controller Components

**OBJECTIVE:** Simplify the development of view and controller classes using inheritance. Provide end users with a convenient means of interacting with the application components. Ensure robust application behavior by managing the lifecycle of view and controller components.

#### • Unifying common properties and methods in a component superclass

- Unifying common construction code
- Using abstract methods to create a common interface
- Creating a user-friendly component API
- Managing the lifecycle of component objects

#### **Enabling Interactive User Control with Controller Classes**

**OBJECTIVE:** Create controller classes to provide user interaction with the application data. Define and implement the core controller responsibilities and actions.

- Controllers within the MVC framework
- The relationship of a controller to a model
- The relationship of a controller to its graphics objects
- Hybrid components that are both a view and a controller
- The interaction between a model and its controllers

#### **Integrating Components** with an Application Launcher

**OBJECTIVE:** Integrate all application components and create a class managing the initialization, dependencies, and running status of an application. Define and implement the core application launcher responsibilities and actions.

- Creating the main application figure and layout
- Creating and storing the application components
- Designing applications with menus and toolbars
- Packaging and updating apps



# Expand your knowledge

