

# MATLAB to C with MATLAB Coder

## Training Objectives

This two-day course focuses on generating C code from MATLAB® code using MATLAB Coder™. The focus is on developing MATLAB code that is ready for code generation, generating C code that meets optimization requirements, and integrating generated code into parent projects and external modules. This course is intended for intermediate to advanced MATLAB users.

## Prerequisites

*MATLAB® Fundamentals* and knowledge of C programming language

## Products

- MATLAB Coder™

## Course Outline

### Day 1 of 2

#### Code Generation with MATLAB Coder (1.5 hrs)

*Objective:* Become familiar with MATLAB Coder and its applications.

- MATLAB Coder overview
- Workflow for generating C code from MATLAB code
- Generating C code
- Verifying generated code
- Navigating generated code

#### Preparing MATLAB Code for Code Generation (2.0 hrs)

*Objective:* Use MATLAB Coder coding standards to write MATLAB code that is ready for code generation.

- Translating MATLAB code into C code
- Calling unsupported MATLAB functions
- Preparing existing MATLAB code
- Code preparation workflows

#### Working with Fixed-Size Data (2.0 hrs)

*Objective:* Generate C code from MATLAB code that has fixed-size or constant inputs.

- Data characteristics overview
- Specifying fixed-size, top-level inputs
- Specifying constant top-level inputs

## Working with Variable-Size Data (2.5 hrs)

*Objective:* Generate C code from MATLAB code that has variable-size inputs or local data.

- Specifying variable-size, top-level inputs
- Specifying variable-size local data
- Reusing variables

## Day 2 of 2

## Working with Global Data, Structures, and Cell Arrays (2.0 hrs)

*Objective:* Generate C code from MATLAB code that contains persistent data, global variables, input structures, or cell arrays.

- Persistent variables
- Global variables
- Working with structures
- Cell arrays in generated code
- Passing arguments by reference

## Integrating with External Code (2.0 hrs)

*Objective:* Integrate generated C code from MATLAB Coder with external C code.

- Code integration overview
- Entry points to generated code
- Integrating external C code using MATLAB Coder interface
- Integrating external C code using an external IDE
- Calling external C functions
- Code verification and profiling
- Source code debugging

## Optimizing Generated Code (2.0 hrs)

*Objective:* Use various options and techniques to optimize generated code.

- Code optimization with loop unrolling and null initialization
- Function inlining and file partitioning
- Configuration objects
- Removing unnecessary code
- Naming conventions in generated code
- Converting a project to a script

## Appendix A: Fixed Point Design (2.0 hrs)

*Summary:*

- Fixed-Point Designer™ overview
- Automated workflow for fixed-point conversion