

**From MATLAB Scripts to Complete Programs**

**Course Number:** MTLB-100
**Duration:** 1 day

**Overview**

This MATLAB training course teaches attendees how to create MATLAB programs composed of separate functional modules, advancing beyond simplistic scripts often used by MATLAB beginners. Students also learn programming design principles, as well as debugging and unit-testing for improved program reliability.

**Prerequisites**

Students should be comfortable using the MATLAB environment. No prior programming experience or familiarity is assumed for this course. However, familiarity with programming in any language 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**

* Transition from script-centric code to a program-centric one
* Apply good design principles to their code
* Understand various data-passing mechanisms between program components
* Use MATLAB timers for periodic or scheduled code
* Leverage MATLAB’s built-in debugger to detect and resolve bugs
* Use MATLAB’s built-in unit-testing framework to improve program reliability

**Outline**

* Introduction to MATLAB programming
	+ So what’s wrong with using scripts?
	+ Anatomy of a MATLAB function
	+ Input and output arguments
	+ MATLAB function types
	+ Using function handles
	+ Function call precedence
	+ MATLAB’s command/function duality
* Programmatic aspects
	+ Variables scope
	+ Flow control
	+ Passing data between program components
	+ Protecting source code IP using pcode
* Improving code reliability and robustness
	+ Coding conventions and best practices
	+ Using the MATLAB Debugger
	+ Code coverage with the MATLAB Profiler
	+ MATLAB’s unit-testing framework
	+ MATLAB’s Code Analyzer
	+ Folder-level reports: Code Compatibility, Contents, Coverage, Dependency
* Teamwork
	+ GIT/SVN integration
	+ MATLAB Projects
	+ Tradeoff considerations using shared network access
* Conclusion