

MATLAB Programming Techniques

Prerequisites

MATLAB Fundamentals

Day 1 of 2	
Structuring Data	<p>Objective: Explore choices for storing data within a MATLAB application.</p> <ul style="list-style-type: none">Data typesConcatenation and conversionCell and structure arraysExtracting data
Managing Data Efficiently	<p>Objective: Use common techniques for improving performance when storing, accessing, and processing data.</p> <ul style="list-style-type: none">PreallocationVectorizationMemory managementObjects and classes
Utilizing Development Tools	<p>Objective: Use integrated tools to manage code, diagnose problems, debug applications, and measure performance.</p> <ul style="list-style-type: none">Generating reports on multiple filesFinding potential problems in codeDebugging codeAssessing code performanceProjects
Day 2 of 2	
Creating Robust Applications	<p>Objective: Create robust applications that withstand unexpected input and produce meaningful errors. Use built-in MATLAB functions and programming constructs, and employ standard techniques for handling error conditions.</p> <ul style="list-style-type: none">Creating flexible function interfacesChecking for warning and error conditionsWorking with the try-catch construct and MException objects

Structuring Code	<p>Objective: Modularize code into readable and maintainable functions. Explore the different function types available in the MATLAB language and the tradeoffs associated with each type. See how function type affects the performance and reliability of the code.</p> <ul style="list-style-type: none">Private functionsLocal functionsNested functionsFunction handlesPrecedence rulesComparison of function types
Verifying Application Behavior	<p>Objective: Create tests to verify that code meets requirements.</p> <ul style="list-style-type: none">Writing test scriptsRunning test scriptsUsing test functions