

CANDIDATE INFORMATION

MathWorks®

CERTIFIED MATLAB®
PROFESSIONAL

 MathWorks®**CERTIFIED MATLAB®
PROFESSIONAL**

CERTIFIED MATLAB PROFESSIONAL EXAM

Earning the Certified MATLAB Professional credential demonstrates that you have expanded your basic MATLAB skills to a level of mastery on par with the proficiency of the most advanced members of the MATLAB community.

PREREQUISITES

To properly prepare, we recommend taking the following MathWorks training courses:

- [MATLAB for Data Processing and Visualization](#)
- [MATLAB Programming Techniques](#)
- [Building Interactive Applications in MATLAB](#)

SAMPLE PROBLEMS

View [sample problems](#) representative of the format and difficulty level expected on the exam.

PREPARING FOR YOUR EXAM

OBJECTIVES TESTED

MathWorks training courses provide coverage across these objectives as well as exercises if additional learning and practice are necessary.

DATA PROCESSING AND VISUALIZATION

Importing Data

- Import a mixture of data types from arbitrarily formatted text files.
- Import only required columns of data from a text file.
- Automatically read data from multiple files in a folder.
- Merge data from multiple files.

Processing Data

- Process data with missing elements.
- Create and modify categorical arrays and use them to group and extract data.
- Aggregate and count groups of data.

Customizing Visualizations

- Determine property names and values associated with graphics objects.
- Locate and manipulate graphics objects.
- Customize plots by modifying properties of graphics objects.

Working with Irregular Data

- Import data from separate sections of a text file.
- Extract data from container variables.
- Interpolate irregularly spaced three-dimensional data.
- Visualize three-dimensional data in two and three dimensions.

PROGRAMMING TECHNIQUES

Utilizing Development Tools

- Ensure code provides desired results by using integrated MATLAB® code analysis and debugging tools.
- Measure code performance using tools like the MATLAB Profiler.
- Describe the concept of numerical accuracy.

Verifying Application Behavior

- Write code that provides tests for desired application behavior.
- Use a test suite to automatically verify application behavior.

MathWorks®

CERTIFIED MATLAB®
PROFESSIONAL

ADDITIONAL INFORMATION

For additional information regarding the MathWorks Certification program, visit mathworks.com/certification

Creating Robust Applications

- Call query functions to determine properties of variables.
- Define default values for function inputs.
- Implement a try-catch construct, along with MException objects, for handling error conditions.
- Write code to validate assumptions about function inputs and provide customized error messages for violations.

Structuring Code

- Select an appropriate type of MATLAB® function based on requirements for function visibility and workspace access.
- Create and call an anonymous function to change the interface to an existing function.
- Determine which function a program will call when multiple possibilities exist.

Structuring Data

- Select the most appropriate data type for an application by considering factors such as memory usage and efficiency of data extraction.
- Convert and concatenate data stored in cell arrays, structures, and structure arrays.
- Extract and manipulate subsets of data from various data set organizations.

Managing Data Efficiently

- Write code for preallocating various types of arrays.
- Use vectorization techniques to improve code performance.
- Explain memory usage when passing arrays to functions.

Creating a Toolbox

- Create documentation for a custom toolbox.
- Package code and documentation as a custom toolbox.

BUILDING INTERACTIVE APPLICATIONS

Graphics Objects

- State the layers in the graphical object hierarchy in MATLAB
- Create a MATLAB graphics object
- Determine graphics object properties and acceptable values
- Obtain a variable that refers to a graphics object
- Modify properties of a graphics object

Components of an Interface

- State the order of execution of the application code throughout the lifetime of the application
- Add a graphical control, such as a pushbutton, to a MATLAB figure window
- Specify behavior of a graphical control by associating it with MATLAB code

Programming Considerations for Interactive Applications

- Organize graphics objects to facilitate passing them into callbacks
- Write a function for use as a callback
- Pass user-defined data into callback functions
- Organize interface creation code and callbacks into a single MATLAB file

MathWorks®

CERTIFIED MATLAB®
PROFESSIONAL

Creating Interfaces Using GUIDE

- Modify behavior of graphical objects created by GUIDE
- Use GUIDE to lay out user interface objects
- Assign unique names to graphical objects using the Tag property
- Modify layout and properties of graphical objects created by GUIDE

Programming Applications Using the GUIDE Template

- Use the handles structure created by GUIDE to manipulate graphics objects within a callback
- Write callbacks that can communicate with each other by adding local data to the application

MathWorks®

CERTIFIED MATLAB®
PROFESSIONAL

TEST FORMAT

The MathWorks Certified MATLAB Professional (MCMP) exam consists of two sections: 25 multiple-choice questions and 8 performance-based problems. MATLAB access is not permitted during the multiple-choice section of the exam. The performance-based problem section requires code segments to be written in MATLAB. MATLAB and the documentation will be available during this portion of the exam, though no other resources, online or otherwise, are permitted. To earn the MCMP credential, submissions for both sections of the exam must meet or exceed the passing criteria for the exam instance.

WRITING MATLAB CODE

The performance-based problems require code submissions written in MATLAB. Submissions must meet all the requirements outlined in the problem statement as well as the basic expectations outlined in the next section.

While there are always opportunities to improve upon submissions by adding additional error checking, comments, or code for edge cases, these additions need to be balanced with the time constraint of the exam. Consider moving on to other problems if spending more than 15-20 minutes on a problem. There will be no bonus points for solutions that go above and beyond the requirements. Additionally, there are no bonus points for “clever tricks” or obscure syntax. Code submissions should clearly communicate the solution to other MATLAB programmers.

Comments in the MATLAB code are welcome and appreciated to help explain the intent of the code. However, given the time constraints of the exam, comments are not required.

MathWorks®

 CERTIFIED MATLAB®
 PROFESSIONAL

EXPECTATIONS FOR SUBMISSIONS

Each submission must meet minimum criteria to receive credit. The scoring process also evaluates requirements set forth in the problems statement. The table below outlines the minimum criteria:

Category	Criteria
Meets Requirements	<p>Solutions must not:</p> <ul style="list-style-type: none"> • Make system calls using system command, ! operator, or any other method of accessing a system command prompt. • Use MEX-files or Simulink blocks. • Make calls through external interfaces to any other programming environments such as Java, Python, .NET, or ActiveX. • Make calls to undocumented functionality, or anything that does not contain explicit instructions in the documentation for use. • Exception: Calls to any documented, pre-existing MATLAB functions that may make use of any of the functionality outlined above are allowed.
Correct Answer/ Stability	<p>Solutions must not:</p> <ul style="list-style-type: none"> • Produce run-time errors as a result of default execution as outlined in the problem statement. • Produce warnings that indicate final results are incorrect, incorrect functions are being called, or the correct functions are being called incorrectly. • Exception: Errors are acceptable when a problem statement explicitly requires an error for a given set of inputs or conditions.
Implementation	<p>Solutions must not:</p> <ul style="list-style-type: none"> • Use functions which indirectly change the workspace such as assignin, evalin, eval, and feval. • Write new functions or code that replicate existing MATLAB functionality (see table). • Contain Code Analyzer warnings if there is an automatic fix or a fix with instructions provided. • Violate any of the stated Vectorization Rules (see table). • Use variable names that collide with common MATLAB functions (see list of common MATLAB functions). • Contain code that grows the size of an array incrementally in a loop when the final array size is known. • Exception: Automatically generated code may contain Code Analyzer messages. These messages do not need to be addressed.

MathWorks®

 CERTIFIED MATLAB®
 PROFESSIONAL

VECTORIZATION RULES

Unless otherwise noted in a problem statement, the vectorization rules outlined in the table below serve as the minimum criteria for all submissions

Rule	Accepted Application	Example Violation
Use element-wise operators to perform mathematical, relational, or logical operations on corresponding elements of arrays.	<pre>x = rand(1, 10); y = rand(1, 10); z = x .* y;</pre>	<pre>x = rand(1, 10); y = rand(1, 10); for i = 1:10 z(i) = x(i) * y(i); end</pre>
Pass entire arrays to functions that accept them instead of passing smaller subsets individually in a loop.	<pre>x = 1:10; y = sin(x);</pre>	<pre>x = 1:10; for i = 1:10 y(i) = sin(x(i)); end</pre>
Call functions that return entire arrays in a single function call rather than building an array incrementally.	<pre>x = rand(1, 10)</pre>	<pre>for i = 1:10 x(i) = rand(); end</pre>
Use vectors for extracting multiple elements of an array when indexing.	<pre>x = rand(5); y1 = x(:, 4);</pre>	<pre>x = rand(5); for i = 1:5 y1(i) = x(i, 4); end</pre>
Use logical indexing for the extraction of elements of an array based on a condition.	<pre>x = randn(1, 30); y = x(x > 0);</pre>	<pre>x = randn(1, 30); for i = 1:30 if x(i) > 0 y = [y x(i)]; end end</pre>

MathWorks®

 CERTIFIED MATLAB®
PROFESSIONAL

MATLAB FUNCTIONALITY TO KNOW

Familiarity with the MATLAB operators, keywords, and functions in the table below is assumed knowledge for the MCMP exam. Submissions for exam problems must not recreate any of this functionality when the appropriate function already exists to address the need. Care should also be taken not to choose variable names that take precedence over these function names. Submissions for exam problems may use any other documented functions not appearing in the table, as long as it is not part of an add-on product (toolbox). Additionally, exam problems may introduce other functions as part of the problem statement.

Mathematical Operators	+ - * /	\ ^ . * . /	.\ .^
Data Types	cell struct categorical	table datetime	duration calendarDuration
Mathematical Functions	sin cos tan asin acos atan abs	exp log log10 log2 nthroot round sqrt	polyfit polyval pi ceil floor mod
Array Creation Functions	ones zeros rand randi randn	true false eye linspace logspace	: (colon operator) meshgrid
Statistical Functions	sum prod cumsum cumprod mean	median min max diff	std var cov fft
Array Dimensions	length	numel	size

MathWorks®

 CERTIFIED MATLAB®
PROFESSIONAL

MATLAB FUNCTIONALITY TO KNOW (CON'T)

Set Operations	union intersect unique	sort sortrows ismember	setdiff setxor
Grouping and Binning	accumarray discretize	findgroups splitapply	histcounts
String Operations	strcmp strcmp strsplit	strjoin strtrim strfind	lower upper deblank
Dates and Time	datenum datevec	datestr now	clock between
Plotting Functions	plot plotyy loglog semilogx semilogy scatter contour surf pie	bar histogram image imagesc subplot xlabel ylabel title legend	text axis ylim xlim grid hold colormap colorbar datetick
Graphics and UI Components	get set findobj findall gcf gca	uicontrol uitable uipanel uimenu uitoolbar guidata	figure axes uigetfile uiputfile msgbox errordlg close
Logical and Relational Operators	> < >=	<= == ~=	~ &
Logical Functions	any all nnz find isequal	isnan isinf isempty ismissing isnumeric	isvector isfinite isbetween isa is* (where * is a datatype)

MathWorks®

 CERTIFIED MATLAB®
PROFESSIONAL

MATLAB FUNCTIONALITY TO KNOW (CON'T)

File I/O	load save fopen fclose feof fscanf	fprintf disp textscan datastore fgetl imread	imwrite xlsread xlswrite readtable dlmread dlmwrite
Conversion Functions	num2str str2double cell2mat mat2cell num2cell	cellstr table2array array2table table2cell cell2table	cell2struct struct2cell char logical
Programming Keywords	break case catch classdef continue else	elseif end for function if	otherwise return switch try while
Vectorization	repmat reshape bsxfun	arrayfun cellfun structfun	varfun rowfun
Help and Troubleshooting	doc help whos which	ver tic toc timeit	clc clear error warning

MathWorks®

CERTIFIED MATLAB®
PROFESSIONAL

HOW TO PREPARE

Review the Tested Objectives for the exam. MathWorks Training courses provide coverage across these objectives as well as exercises if additional learning and practice are necessary.

Review the “MATLAB functionality to Know” table. Study the documentation pages for any functions that are unfamiliar.

Work through exercises. MathWorks training courses provide additional exercises for practice.

Read the Release Notes for the latest releases of MATLAB to stay up-to-date on new functionality, as well as changes to older functionality.

Pay attention to Code Analyzer messages in the MATLAB Editor. Be able to fix anything flagged in the code as potentially incorrect or inefficient. A list of the Code Analyzer Messages for the current release can be found under the “Code Analyzer” section of the MATLAB preferences.

Try the exam practice problems. These problems offer an idea of the difficulty and format to expect on the exam.