

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

#### Processing Data

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

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

### MATLAB PROGRAMMING TECHNIQUES

#### Structuring Data

- Extract and manipulate subsets of data from various data set organizations.
- 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 various data types.

#### Managing Data Efficiently

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

#### Utilizing Development Tools

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

**MathWorks®****CERTIFIED MATLAB®  
PROFESSIONAL**

## ADDITIONAL INFORMATION

For additional information regarding the MathWorks Certification Program, visit [mathworks.com/certification](https://mathworks.com/certification).

### Creating Robust Applications

- Call query functions to determine properties of variables.
- Implement a try-catch construct, along with MException objects, for handling error conditions.
- Define default values for function inputs.
- 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.
- Determine which function a program will call when multiple possibilities exist.
- Create and call an anonymous function handle to change the interface to an existing function.

### Verifying Application Behavior

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

### Creating a Toolbox

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

## BUILDING INTERACTIVE APPLICATIONS IN MATLAB

### Using the App Designer Environment

- Package and deploy apps to share with others.
- Use the App Designer interface to lay out components of an app.
- Make components interactive by adding callbacks to them.

### Creating and Updating Plots

- Create apps with graphical elements that can be modified by user interaction.
- Create apps with continuously adjustable components.
- Add extra functions and callbacks to apps.

### Creating Responsive Components

- Store and use application data within an app.
- Create apps with components that adapt appropriately to user actions.

### Managing Multiple Windows and Applications

- Use predefined dialog boxes in app callbacks.
- Catch unexpected errors and communicate them to the user.
- Build apps that can communicate with each other.

MathWorks®

CERTIFIED MATLAB®  
PROFESSIONAL

## TEST FORMAT

The MathWorks Certified MATLAB Professional (MCMP) exam consists of two sections: 25 multiple-choice questions and eight 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 problem statement. The table below outlines the minimum criteria.

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

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 &gt; 0);</pre>	<pre>x = randn(1, 30); for i = 1:30     if x(i) &gt; 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 re-create 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 they are not part of an add-on product (toolbox). Additionally, exam problems may introduce other functions as part of the problem statement.

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

**MathWorks®**
**CERTIFIED MATLAB®  
PROFESSIONAL**
**MATLAB FUNCTIONALITY TO KNOW (Cont.)**

<b>Set Operations</b>	union intersect unique ismember	sort sortrows setdiff setxor	join innerjoin outerjoin
<b>Grouping and Binning</b>	accumarray discretize	findgroups splitapply	histcounts groupsummary
<b>String Operations</b>	split join contains	replace extract plus	lower upper deblank
<b>Dates and Time</b>	datetime datestr between	calyears calmonths calweeks caldays	minutes seconds hours years
<b>Plotting Functions</b>	plot plotyy loglog semilogx semilogy scatter contour surf pie bar	histogram image imagesc pcolor tiledlayout subplot xlabel ylabel title legend	text axis ylim xlim grid hold colormap colorbar patch
<b>Graphics and UI Components</b>	gcf gca figure axes close delete	uifigure uiaxes uistack uigetfile uiputfile uigetdir	inputdlg listdlg uiopen uisave uialert uiconfirm
<b>Logical and Relational Operators</b>	> < >=	<= == ~=	~ &   && 



MathWorks®

 CERTIFIED MATLAB®  
PROFESSIONAL

**MATLAB FUNCTIONALITY TO KNOW (Cont.)**

<b>Logical Functions</b>	any all nnz find isequal	isnan isinf isempty ismissing isnumeric strcmp	isvector isfinite isbetween isa is* (where * is a datatype)
<b>File I/O</b>	load save fopen fclose feof fscanf fprintf disp	textscan datastore read readaall preview fgetl	imread imwrite readtable writetable readmatrix writematrix
<b>Conversion Functions</b>	num2str str2double cell2mat mat2cell num2cell	cellstr table2array array2table table2cell cell2table	cell2struct struct2cell char logical string categorical
<b>Programming Keywords</b>	break case catch classdef continue else	elseif end for function if	otherwise return switch try while
<b>Function Interfaces</b>	arguments nargin nargout	varargin varargout	validateattributes
<b>Vectorization</b>	repmat reshape bsxfun	arrayfun cellfun structfun	varfun rowfun
<b>Help and Troubleshooting</b>	doc help whos which	ver tic toc timeit	clc clear error warning

MathWorks®

CERTIFIED MATLAB®  
PROFESSIONAL

## HOW TO PREPARE

The best preparation is hands-on experience with a wide variety of applications in MATLAB. The professional-level exam validates proficiency that comes with a few years of experience analyzing and visualizing data, optimizing performance, and creating graphical applications. In addition to this experience, the following activities are recommended to be ready for the exam:

- Review the tested objectives for the exam. MathWorks Training courses provide coverage across these objectives and exercises if additional learning and practice are necessary.
- Review the “MATLAB Functionality to Know” table. Study the documentation pages for any unfamiliar functions.
- Read the release notes for the latest releases of MATLAB to stay up to date on new functionality and 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 checks performed by the Code Analyzer within the current release can be found under the “Code Analyzer” section of the MATLAB preferences.
- Try the exam sample problems. These problems offer an impression of the difficulty and format to expect on the exam.