Ex. No: 06 Introduction to MATLAB - Writing code using MATLAB programming

Date:

Aim:

1. To familiarize with the working environment of MATLAB.
2. To understand the basic commands.
3. To practice the operations with variables, Characters and Strings

Introduction:

1. MATLAB:

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

MATLAB is developed by MathWorks.

It allows **matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.**

It has numerous **built-in commands** and **math functions** that help you in mathematical calculations, generating plots, and performing numerical methods.

HISTORY:

MATLAB stands for **"MATRIX LABORATORY".**

"MATLAB was invented by mathematician and computer programmer Cleve Moler. He developed MATLAB's initial linear algebra programming .This was followed by Fortran code for linear equations in 1971."

The first early version of MATLAB was completed in the late 1970s. Early versions of MATLAB were simple matrix calculators with 71 pre-built functions.In the 1980s, Cleve Moler met John N. Little. They decided to reprogram MATLAB in C and market it for the IBM desktops that were replacing mainframe computers at the time.John Little and programmer Steve Bangert re-programmed MATLAB in C, created the MATLAB programming language, and developed features for toolboxes.MATLAB was first released as a commercial product in 1984 at the Automatic Control Conference in Las Vegas. MathWorks, Inc. was founded to develop the software and the MATLAB programming language was released.

Over time, MATLAB was re-written for early operating systems created by Digital Equipment Corporation, VAX, Sun Microsystems, and for Unix PCs. The first MATLAB compiler was developed by Stephen C. Johnson in the 1990s.In 2000, MathWorks added a Fortran-based library for linear algebra in MATLAB 6, replacing the software's original LINPACK and EISPACK subroutines that were in C. MATLAB's Parallel Computing Toolbox was released at the 2004 Supercomputing Conference and support for graphics processing units (GPUs) was added to it in 2010. Some especially large changes to the software were made with version 8 in 2012. The user interface was reworkedand Simulink's functionality was expanded.By 2016, MATLAB had introduced several technical and user interface improvements, including the MATLAB Live Editor notebook, and other features.

VERSIONS OF MATLAB:

Version	Release name	Number	Bundled JVM	Year	Release date	Notes
MATLAB 1.0				1984		
MATLAB 2				1986		

MATLAB 3				1987		First Matlab toolbox introduced; for ordinary differential equations
MATLAB 3.5				1990		Ran on DOS but needed a a 386 processor; needed coprocessor.
MATLAB 4				1992		Ran on Windows 3.1x and Macint
MATLAB 4.2c				1994		Ran on Windows 3.1x; needed coprocessor.
MATLAB 5.0	Volume 8			1996	December 1996	Unified releases across all platform
MATLAB 5.1	Volume 9			1997	May 1997	
MATLAB 5.1.1	R9.1					
MATLAB 5.2	R10			1998	March 1998	Last version working on classic M
MATLAB 5.2.1	R10.1					
MATLAB 5.3	R11			1999	January 1999	
MATLAB 5.3.1	R11.1				November 1999	
MATLAB 6.0	R12	12	1.1.8	2000	November 2000	First release with bundled Jav machine (JVM).

MATLAB 6.1	R12.1		1.3.0	2001	June 2001	Last release for Windows 95.
MATLAB 6.5	R13	13	1.3.1	2002	July 2002	
MATLAB 6.5.1	R13SP1			2003		
MATLAB 6.5.2	R13SP2					Last release for Windows 98, ME, IBM/AIX, Alpha/TRU SGI/IRIX.
MATLAB 7	R14	14	1.4.2	2004	June 2004	Introduced anonymous and functions re-introduced for Mac Mac OS X).
MATLAB 7.0.1	R14SP1				October 2004	
	R14SP1 +			2004	November 2004	Parallel Computing Toolbox introd
MATLAB 7.0.4	R14SP2		1.5.0	2005	March 7, 2005	Support added for memory-mappe
MATLAB 7.1	R14SP3	1.5.0			September 1, 2005	First 64-bit version available for XP 64-bit.
MATLAB 7.2	R2006a	15	1.5.0	2006	March 1, 2006	
MATLAB 7.3	R2006b	16	1.5.0		September 1, 2006	HDF5-based MAT-file support add

MATLAB 7.4	R2007a	17	1.5.0_07	2007	March 1, 2007	New bsxfun function added to element-by-element binary operat singleton expansion enabled.
MATLAB 7.5	R2007b	18	1.6.0		September 1, 2007	Last release for Windows and PowerPC Mac; License support for Windows Vista, new format for P-code.
MATLAB 7.6	R2008a	19	1.6.0	2008	March 1, 2008	Major enhancements to object programming abilities with a n definition syntax; ability to namespaces with packages.
MATLAB 7.7	R2008b	20	1.6.0_04		October 9, 2008	Last release for processors w/o New Map data structure, upgra random number generators.
MATLAB 7.8	R2009a	21	1.6.0_04	2009	March 6, 2009	First release for Microsoft 32 64-bit Windows 7; new e interface to .NET Framework.
MATLAB 7.9	R2009b	22	1.6.0_12		September 4, 2009	First release for Intel 64-bit Ma last for Solaris SPARC; new u the tilde operator (~) to arguments in function calls.
MATLAB 7.9.1	R2009b SP1		1.6.0_12	2010	April 1, 2010	Bug fixes.
MATLAB 7.10	R2010a	23	1.6.0_12		March 5, 2010	Last release for Intel 32-bit Ma
MATLAB 7.11	R2010b	24	1.6.0_17		September 3, 2010	Added support for enumeration features for running MATLAB NVIDIA CUDA-based GPUs.

MATLAB 7.11.1	R2010b SP1		1.6.0_17	2011	March 17, 2011	Bug fixes and updates.
MATLAB 7.11.2	R2010b SP2		1.6.0_17		April 5, 2012	Bug fixes.
MATLAB 7.12	R2011a	25	1.6.0_17		April 8, 2011	New rng function to control number generation.
MATLAB 7.13	R2011b	26	1.6.0_17		September 1, 2011	Added ability to access/change variables directly in MAT-files loading into memory, maximum local workers with Computing Toolbox from 8 to
MATLAB 7.14	R2012a	27	1.6.0_17	2012	March 1, 2012	Last version with 32-bi support.
MATLAB 8	R2012b	28	1.6.0_17		September 11, 2012	First with Toolstrip interface, MA Apps introduced; documentation system.
MATLAB 8.1	R2013a	29	1.6.0_17	2013	March 7, 2013	New unit testing framework.
MATLAB 8.2	R2013b	30	1.7.0_11		September 6, 2013	Built in Java Runtime En (JRE) updated to version 7; data type.
MATLAB 8.3	R2014a	31	1.7.0_11	2014	March 7, 2014	Simplified compiler set building MEX-files; USB support in core MATLAB; n local workers no longer limi with Parallel Computing Too

MATLAB 8.4	R2014b	32	1.7.0_11		October 3, 2014	New class-based graphics (a.k.a. HG2); tabbing fun GUI;improved user packaging and help files; ne for manipulations; Git-Subversio ation in IDE; big data with MapReduce (scalable to Hadoop); new py package using Python from MATLAB; new engine inte call MATLAB from Pytho new and functions: webread (RESTfu services with JS support), tcpclient (socket-ba connections), histcounts, hist nimatedline, and others.
MATLAB 8.5	R2015a	33	1.7.0_60	2015	March 5, 2015	
MATLAB 8.5	R2015aSP1		1.7.0_60		October 14, 2015	Last release suppo Windows XP and Win Vista.
MATLAB 8.6	R2015b	34	1.7.0_60		September 3, 2015	New MATLAB execu engine (a LXE); graph and digraph cl s to work with graphs networks; MinGW-w64 supported compiler

						Windows;last version 32-bit support.
MATLAB 9.0	R2016a	35	1.7.0_60	2016	March 3, 2016	Released Live Scr interactive documents combine text, code, and ou (in the style of Lit programming); App Desi introduced: a new develop environment for building (with new kind of UI fig axes, and components); p execution of running prog using a Pause Button.
MATLAB 9.1	R2016b	36	1.7.0_60		September 15, 2016	Added ability to defin functions in scripts;a expansion of dimensions (p provided via explici to bsxfun); tall arrays data;new string type; new to encode/decode JSON MATLAB Engine API for J
MATLAB 9.2	R2017a	37	1.7.0_60	2017	March 9, 2017	Released MATLAB Onl cloud-based MATLAB desk accessed in a browser; double-quoted strings; new memoize funct for Memoization; expan object proper validation; mocking framew k for unit testing; MEX targ 64-bit by defa new heatmap function creating heatmap charts.

MATLAB 9.3	R2017b	38	1.8.0_121		September 21, 2017	Introduced a GPU Coder converts MATLAB cod CUDA code for Nvidia.
MATLAB 9.4	R2018a	39	1.8.0_144	2018	March 15, 2018	Improvements to the Liv introduction of the C+ interface; ability to custo completion; web applicat
MATLAB 9.5	R2018b	40	1.8.0_152		September 12, 2018	Added support for c providers, such as Am Web Services; Ne Network Toolbox repl with Deep Learning Toolb
MATLAB 9.6	R2019a	41	1.8.0_181	2019	March 20, 2019	Released MATLAB Pro added state ma programming with Stateflo
MATLAB 9.7	R2019b	42	1.8.0_202		September 11, 2019	Introduction of 'arg block for input val enabling of dot indexi function outputs; intro of Live Editor Tasks.
MATLAB 9.8	R2020a			2020	March 19, 2020	Removal of Mupad no improved support for CPUs (AVX2); default encoding for MATLA files; ability to stand-alone application Simulink.
MATLAB 9.9	R2020b				September 17, 2020	Improved support for CPUs (AVX2); online of Simulink.

MATLAB SOFTWARE:

Get a Trial-"Start a free 30-day trial of MathWorks software."

User Interface	**Description**
MATLAB	To get a trial, go to **Add-Ons** > **Get Add-Ons**. Click on a toolbox, and then click **Get Trial**. Follow all prompts.
MathWorks Account	1. Sign in to mathworks.com. If you are already signed in, select **My Account** from the drop-down menu under your account icon. 2. Under **My Software**, click **Get a trial**. Follow all prompts.

Link a License to Your Account

User Interface	**Description**
MATLAB	You cannot perform this task from within MATLAB.
MathWorks Account	1. Sign in to mathworks.com. If you are already signed in, select **My Account** from the drop-down menu under your account icon. 2. Under **My Software**, click **Link an additional license**. 3. Follow all prompts. If you are linking a network license, you may need to get the license number from your organization's license administrator.

Update Current Licenses

User Interface	**Description**

MATLAB	On the **Home** tab, in the **Resources** section, click **Help** > **Licensing** > **Update Current Licenses**. MATLAB displays a list of all your MathWorks licenses on this computer, with their status. When you select a license and click **Update Selected License**, MATLAB contacts MathWorks to retrieve the most current version of the License File for the license. The update process overwrites the current License File on your system. You need to restart MATLAB.
MathWorks Account	1. Sign in to mathworks.com. If you are already signed in, select **My Account** from the drop-down menu under your account icon. 2. Under **My Software**, click the license you want to update. If you have more licenses than fit on the screen, click **View additional Licenses or Trials** and then click the license you want. Clicking a license takes you to the License Center. 3. In the License Center, click the Install and Activate tab. 4. Click **Update License File**. 5. Click one of the icons under **Get License File**. You can download the license file or have it emailed to you. 6. Follow all prompts. MathWorks retrieves the most current version of the License File for the license. You need to restart MATLAB.

TOOLBOXES IN MATLAB:

When you are logged in using your MathWorks Account, you can use functions from the following toolboxes if you are licensed to use them:

- Statistics and Machine Learning Toolbox
- Curve Fitting Toolbox
- Control System Toolbox
- Signal Processing Toolbox

- Mapping Toolbox
- System Identification Toolbox
- Deep Learning Toolbox
- DSP System Toolbox
- Datafeed Toolbox
- Financial Toolbox
- Image Processing Toolbox
- Text Analytics Toolbox
- Predictive Maintenance Toolbox

“The most widely used toolboxes are

- **Image Procrssing Toolbox,**
- **Deep Learning Toolbox,**
- **Statistics and Machine Learning Toolbox.”**

MOBILE MATLAB:

Connect to the cloud:

Use your MathWorks Account to connect to MathWorks Cloud from MATLAB Mobile™. Linking a license that is current on MathWorks Software Maintenance Service to your MathWorks Account increases your storage quota and unlocks the ability to acquire data from device sensors.

With your MathWorks Account, you receive:

• Access MATLAB from the command-line

• View, run, edit and create files from the Editor

• Acquire data from device sensors

• Store your files and data on MATLAB Drive (you receive 250 MB of cloud storage)

Link a license that is current on MathWorks Software Maintenance Service to your MathWorks Account to unlock the following features:

• Access to other add-on products on your license

• 5 GB of cloud storage on MATLAB Drive

Features:

• Command-line access to MATLAB and add-on products

• 2D and 3D plots to visualize data

• Editor to view, run, edit and create MATLAB files

• Data acquisition from device sensors

• Image and video acquisition from the camera

• Cloud storage and synchronization with MATLAB Drive

• Custom keyboards to enter typical MATLAB syntax

Limitation:

The following features are not supported:

• Viewing, editing or evaluating live scripts with the Live Editor

• Using MATLAB Apps, such as Curve Fitting

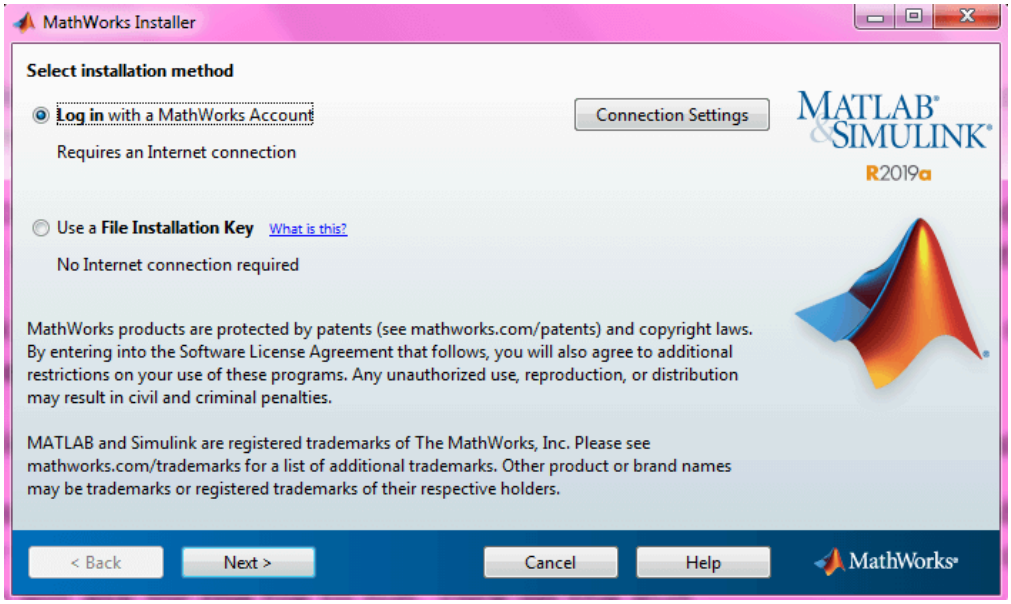
• Creating apps with App Designer

• Interacting with 3D figures

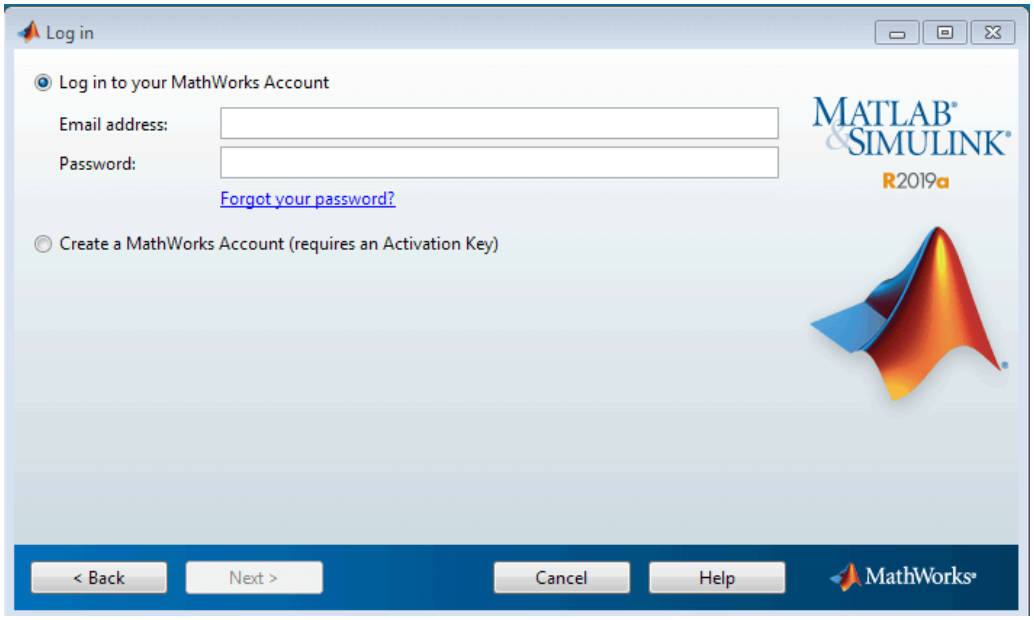
• Opening or creating models using the Simulink graphical environment

MATLAB INSTALLATION:

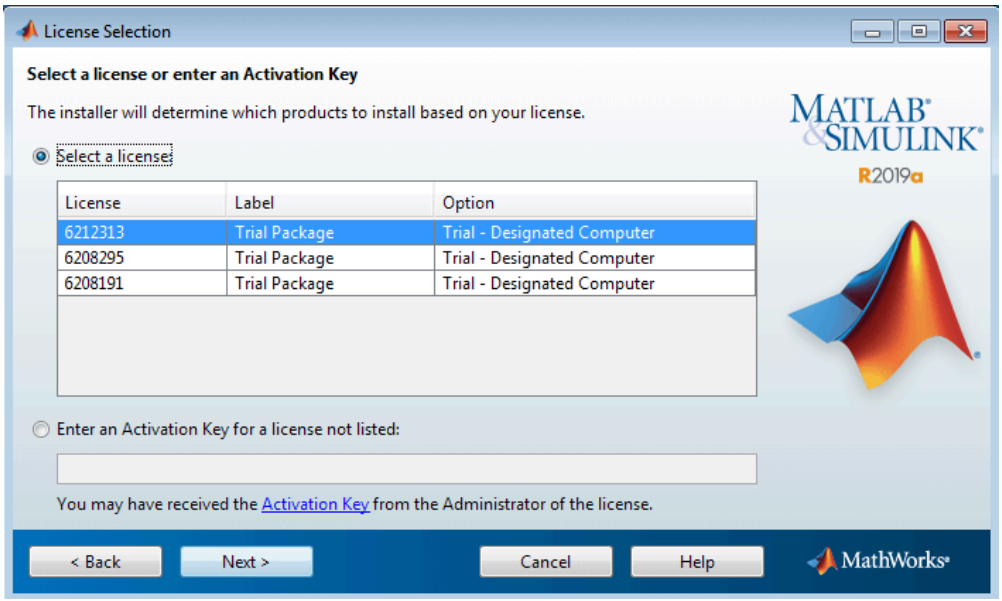
Step 1: Double click on the MATLAB icon (the binary file which we downloaded earlier). After clicking the icon, a pop-up will ask for the installer to run, click on the **Run**. A MathWorks Installer window will pop-up on the screen.



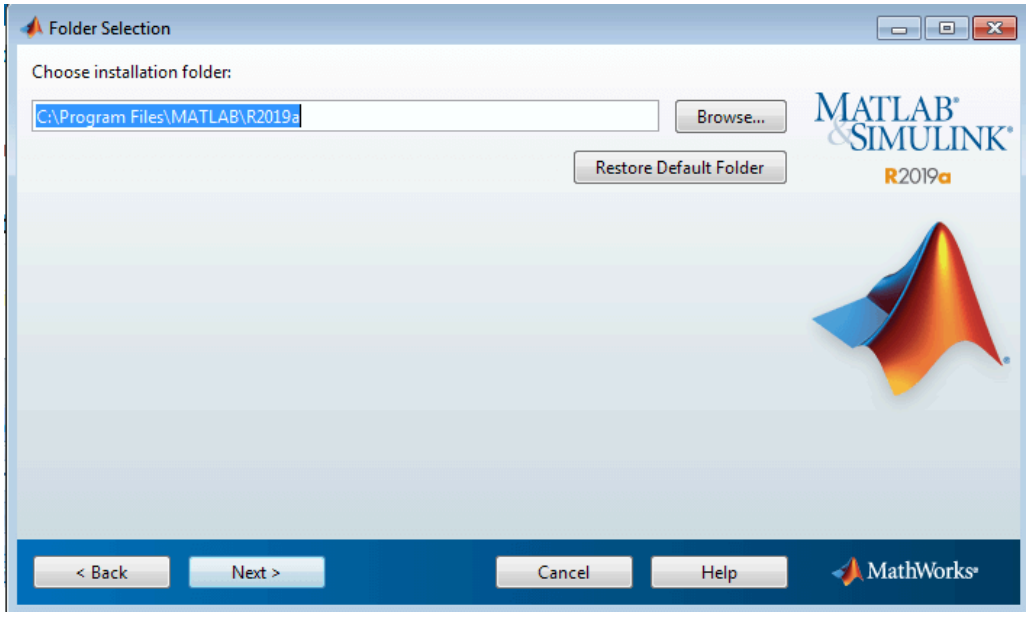
- By default, the first option, i.e., Log in with a MathWorks Account, is selected, we will proceed with this option. And do remember to check your internet connection for proper installation of the MATLAB environment. So click on Next in the bottom of the window.
- Accept the license terms by selecting yes in the next page and again click on the Next button.
- A new page appears, by default, the first option is selected, Log in to your MathWorks Account. Enter here your email id and password that we created during the creation of our account with MathWorks. Refer to the image below and click on next.



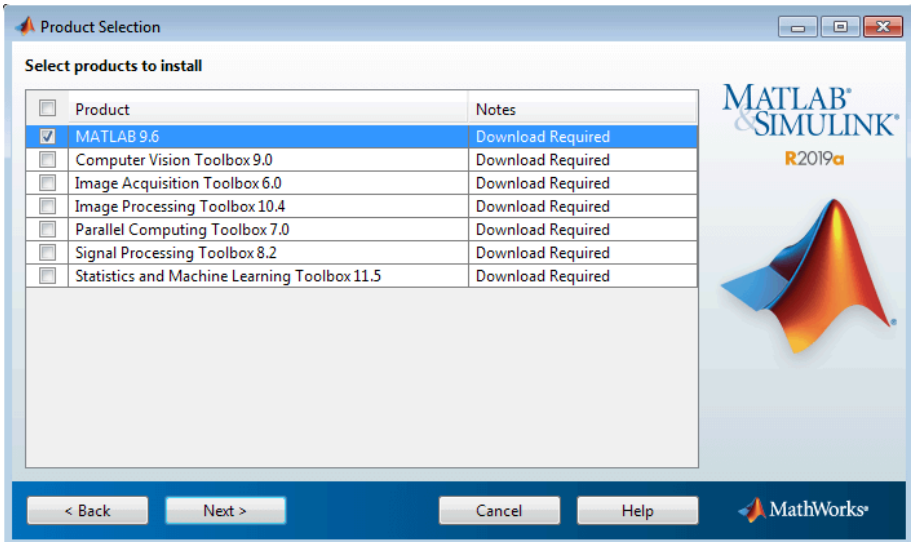
Step 2: A license Selection window will appear, a preselected license id will be highlighted with a blue background. Here you have to select your license id; this is the id which we have saved during STEP 9 of downloading of the installer (we urged to note down that id during that time) and again click on Next.



- A new Folder Selection window appears, no need to change the folder location for installation of MATLAB, click on Next.

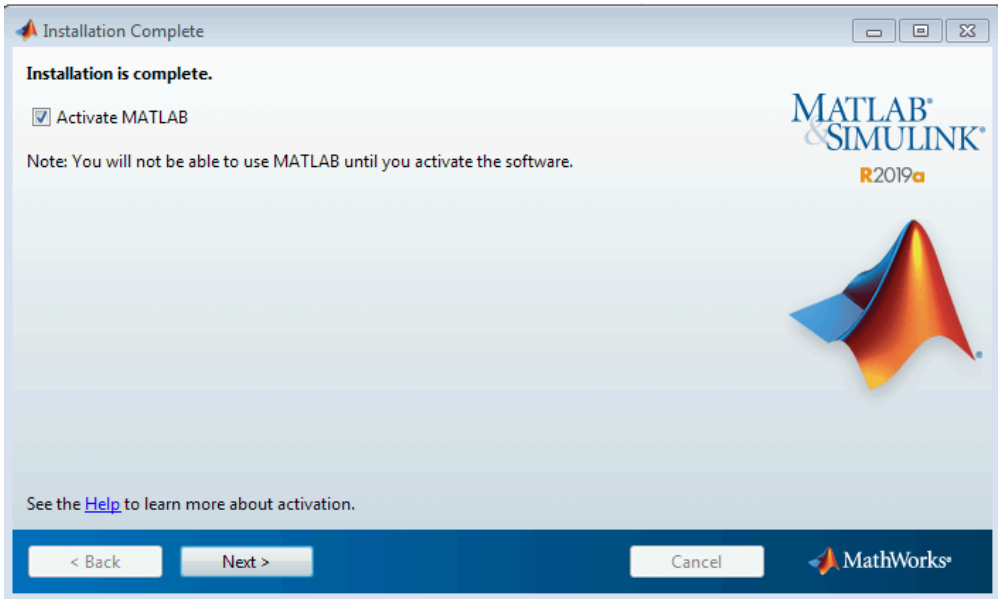


Step 3: Next is Product Selection window, the first product is MATLAB 9.6, this is mandatory to select because it is the MATLAB environment, and from other products, you can choose as many of your choices and click on Next.



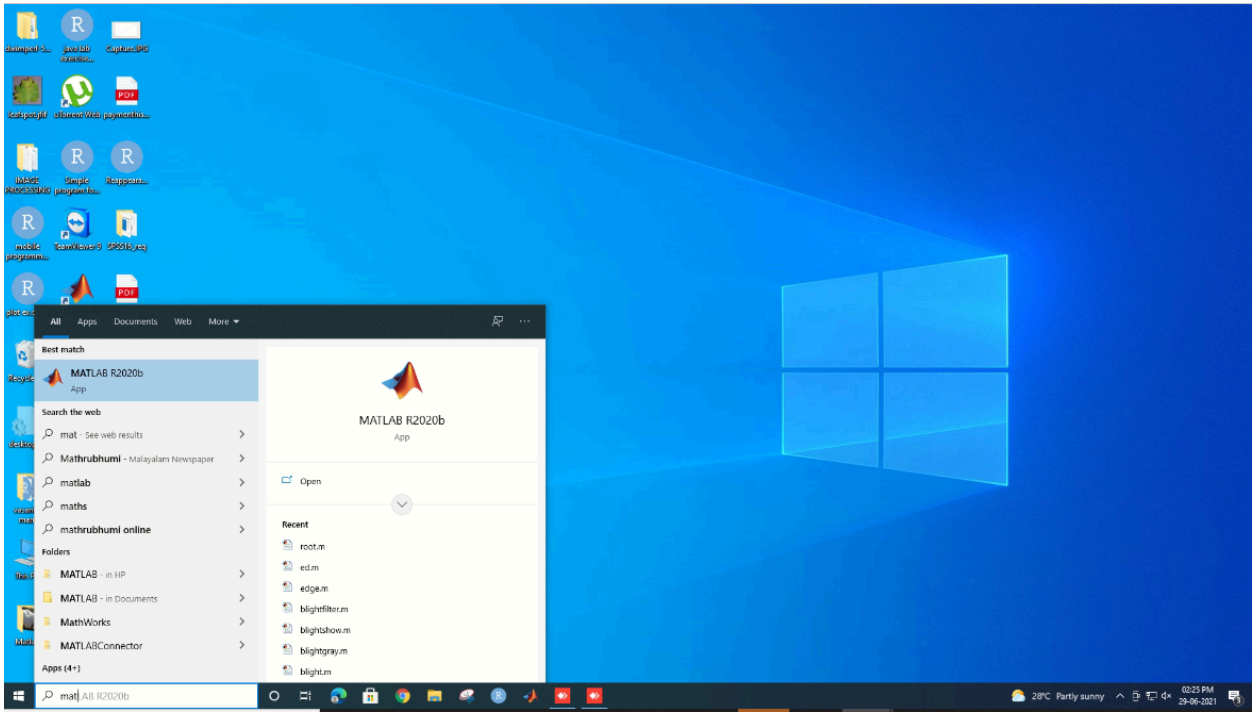
- Next is the Installation Options window, select options as per your choice. Any time you feel something to change, you can go back to the previous step by clicking on the Back button.
- Next is the Confirmation window, here you no need to do anything, confirm what you are going to download in the process of the installation of MATLAB, its other Add-on products, and what is the size of the downloads; and click on Install.
- By clicking on Install, downloading of all the products will be started. It's a massive download, so you have to wait for some time to complete the download.

Step 4: After downloading of all products and completion of the installation, a window appears that says to Activate the MATLAB, no need to do anything, click on the Next button.

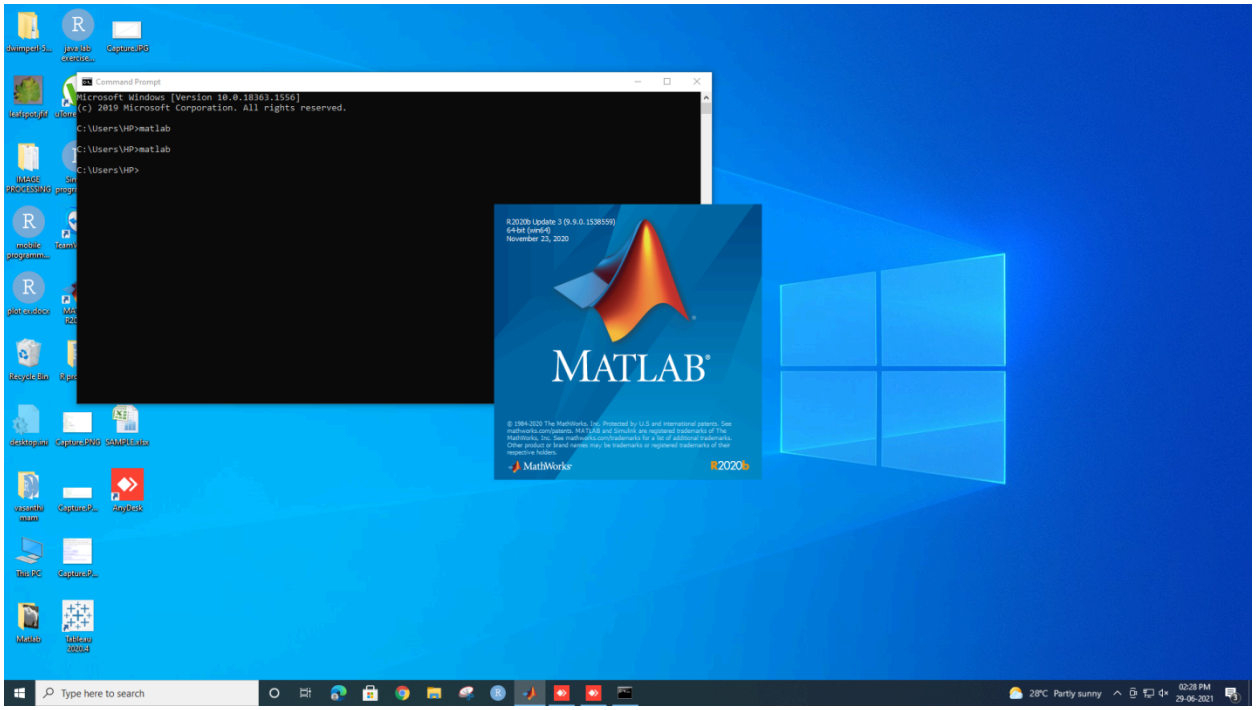


- After clicking on Next, a new window appears that says about what is the meaning of activation. Proceed by clicking on Next.
- Again a new window appears displaying your email id and your products' license id, proceed by clicking on Confirm button.
- Congratulations, you have completed the installation process and successfully installed the MATLAB and its other products. Now click on the Finish button.

Step 5: A MATLAB shortcut will be created on the desktop as per our choice during the installation process. Now we can work with MATLAB by clicking on the icon placed there on the desktop.



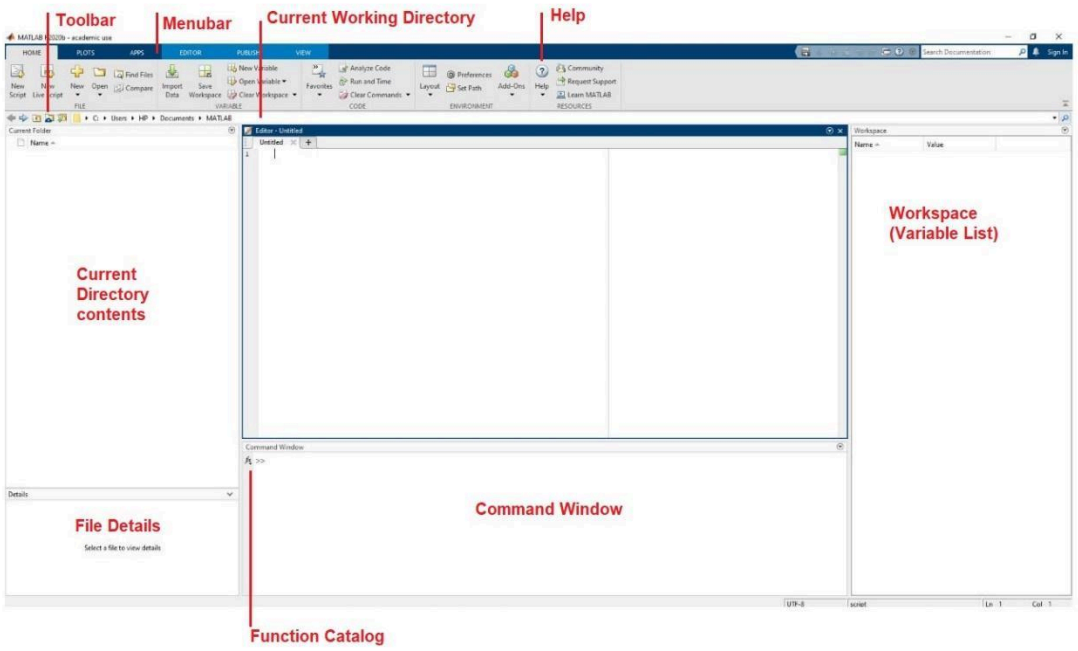
Step 6: By command prompt:

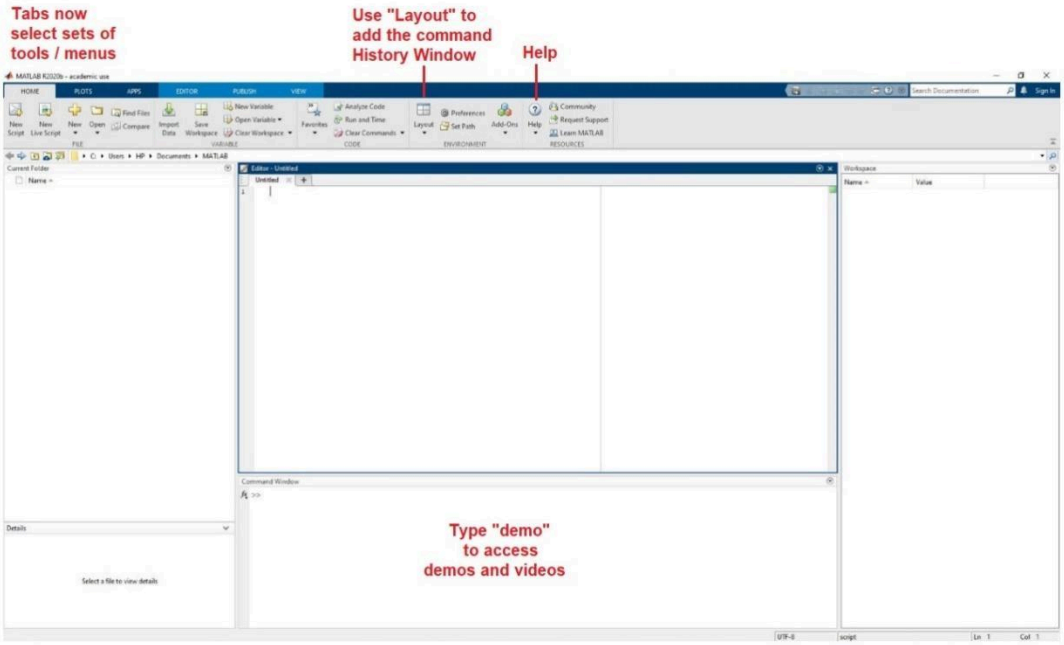


MATLAB DESKTOP:

MATLAB provides a Desktop GUI based environment. The Home tab of the environment contains three panels there:

1. **Current folder:** It is located on the left side; here, you can access your files.
2. **Command Window:** It is located on the right side; it is the command prompt, and here you can enter commands to operate the functions, to assign variables, and for calculations.
3. **Workspace:** It is located on the left side right below the Current Folder; here, all variables that you create are stored, and data from other files can also be imported here.





WORKING WITH MATLAB ENVIORNMENT:

Matlab is an interactive mathematical program that allows mathematical (statistical etc) calculations as well visualization of this data. One of the main tasks you will perform in engineering is problem solving. To solve many of these problems, you will find Matlab to be a

powerful tool to use.Matlab has hundreds of built-in functions and can be used to solve problems ranging from the very simple to the sophisticated and complex. Whether you want to do some simple numerical or statistical calculations, some complex statistics, solve simultaneous equations, make a graph, or run and entire simulation program, Matlab can be an effective tool.

Since MATLAB is such a large and versatile application, the information in these modules will be split into small segments in order to help you gain familiarity. As you progress in your studies, you will probably have an opportunity and need to use various aspects of MATLAB's capabilities.

Opening a Matlab Session

When you first open Matlab, you need to specify the location on the computer where you want to save your work. The "path" is a list of locations i.e. subdirectories, that the Matlab environment uses during its operation. This path needs to be the location of your m-files, diaries, work saving area, and other information

You can add to the path by going to the file menu and using the set path option or using the path browser.

1. Select the File menu.
2. Select the 'Set Path' menu item. Wait for dialog box to appear.
3. Click the 'Add Folder' button and navigate to and highlight your destination folder and click OK. You should see the path to the folder in the list
4. Click the ok button.

Saving your work in Matlab:

There are two options for saving your work in Matlab.

- One is the diary and

- the other is the m-file.

Diaries:

A diary is an ASCII file, which contains everything that appears in the command window. Once you have a diary that catalogs everything that you did in a given Matlab session you can refer to it, see what you have completed for your assignment for example, and use appropriate sections in your lab report. Here is how you create a diary.

>>diary *filename*

where *filename* is the name of your choice for the diary using less than 8 characters. Next you need to type:

>>diary on

This will turn the diary on and now everything that appears in the command window will be recorded. When you are finished you need to type:

>>diary off

This command will turn off your diary. Now you can open your diary in Word, or Notepad, and see all of your work. You can edit the diary by removing all of the unwanted code and errors thus preparing a report or a record of your work.

M-files

An m-file is a program or script that you write, save in a file, and then run in Matlab. To make an m-file, open up a new m-file from the file menu. This will open a special Matlab debugger and program writing environment. You can also write your m-file in a notepad. Whatever you do, remember to save your work on your machine or vuspace.

For example

```
%simplegraph
  y=[012345];
  x=[013579];
  plot(x,y)
     title ('practice graph')
```

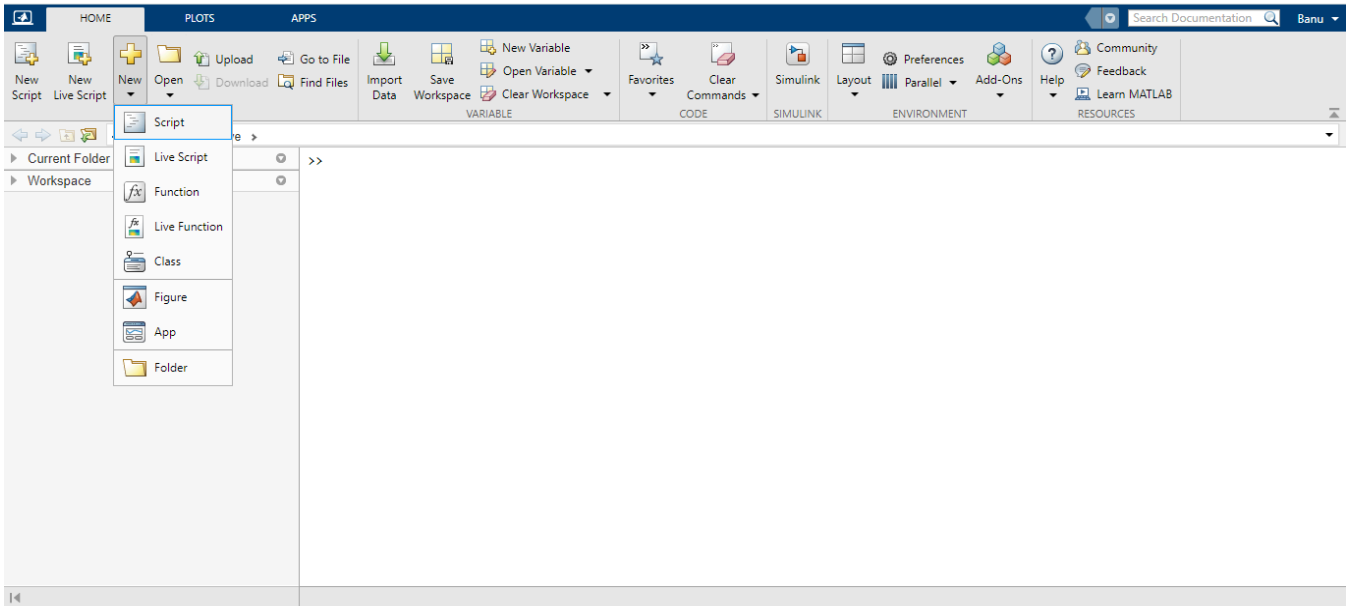
This code plots x and y with the title 'practice graph'.

Now, you need to save your program as an m-file. To do this, select save from the file menu. You can name the program with an extension .m;

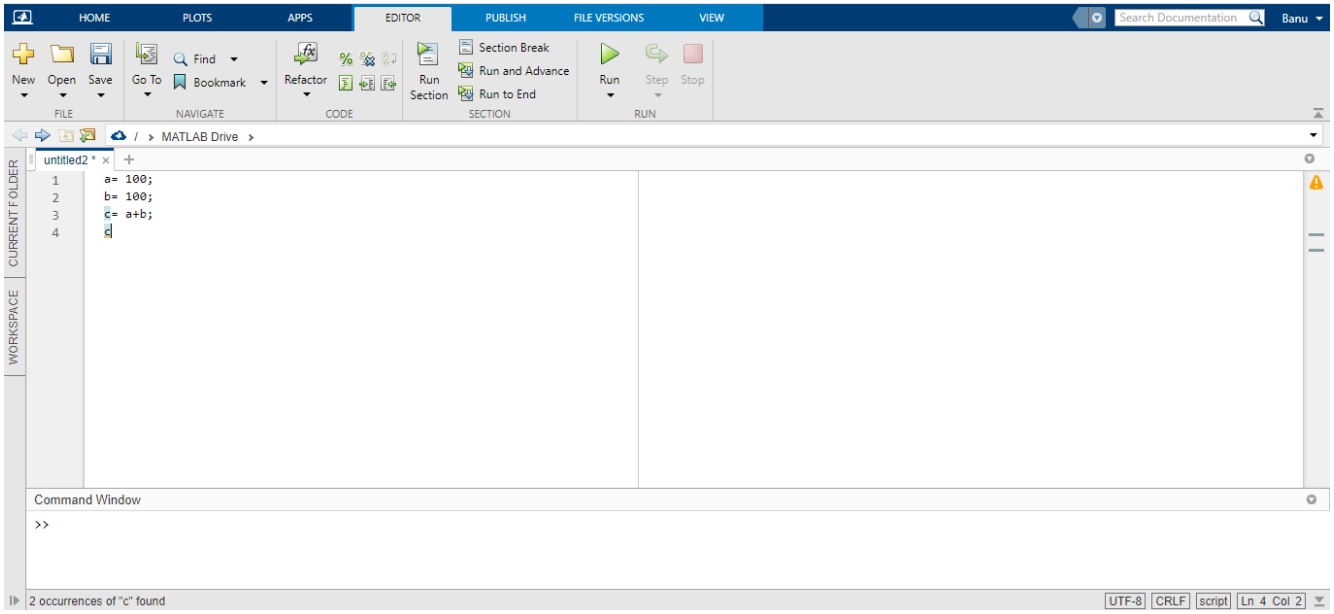
simple.m

Now, the next time you open Matlab, set the path to the same drive and subdirectory where you have the m-file stored. Now all you need to do to execute the program you just wrote is to type the name of the program in the command window. 'simple' without the extension (.m).You need to be sure that you give your m-file a name that is not used as a command code by Matlab (like plot, title, average, etc). You must also make sure that the filename does not have any spaces. If your code appears to be correct but the m-file will not run, try a new filename.Another way to run your m-file is to select the command 'run script' from the file menu. This command will allow you to type in the path and name of your m-file.

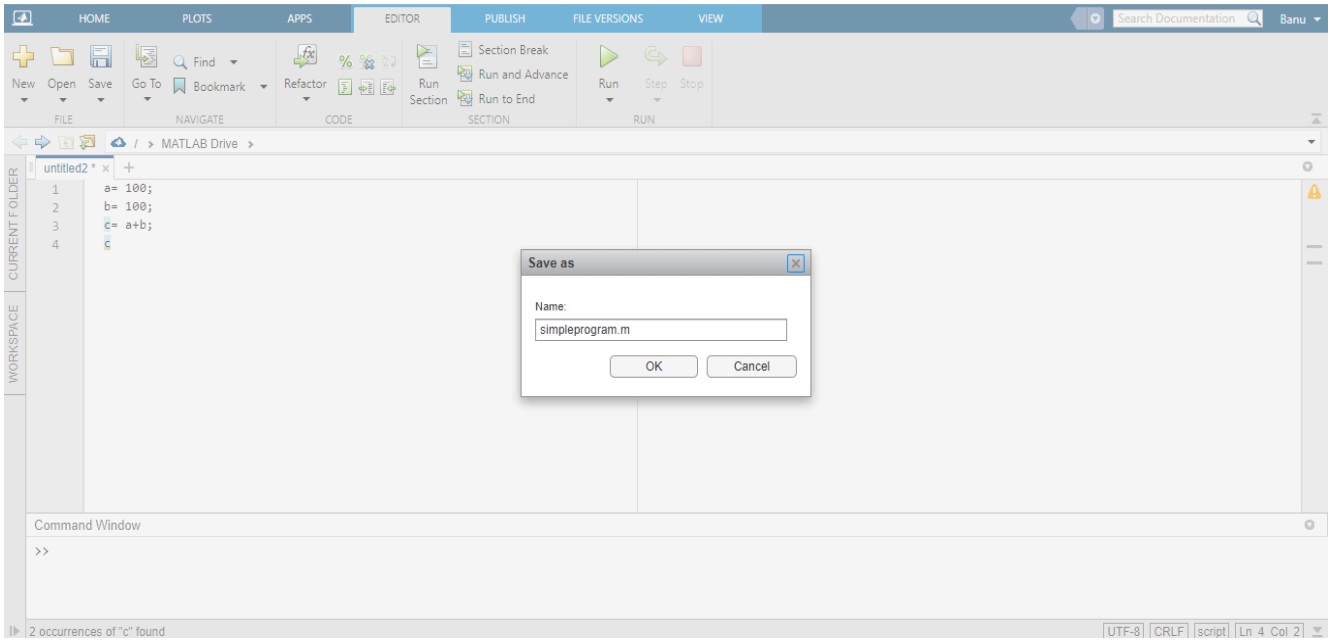
1. Creating New Script:



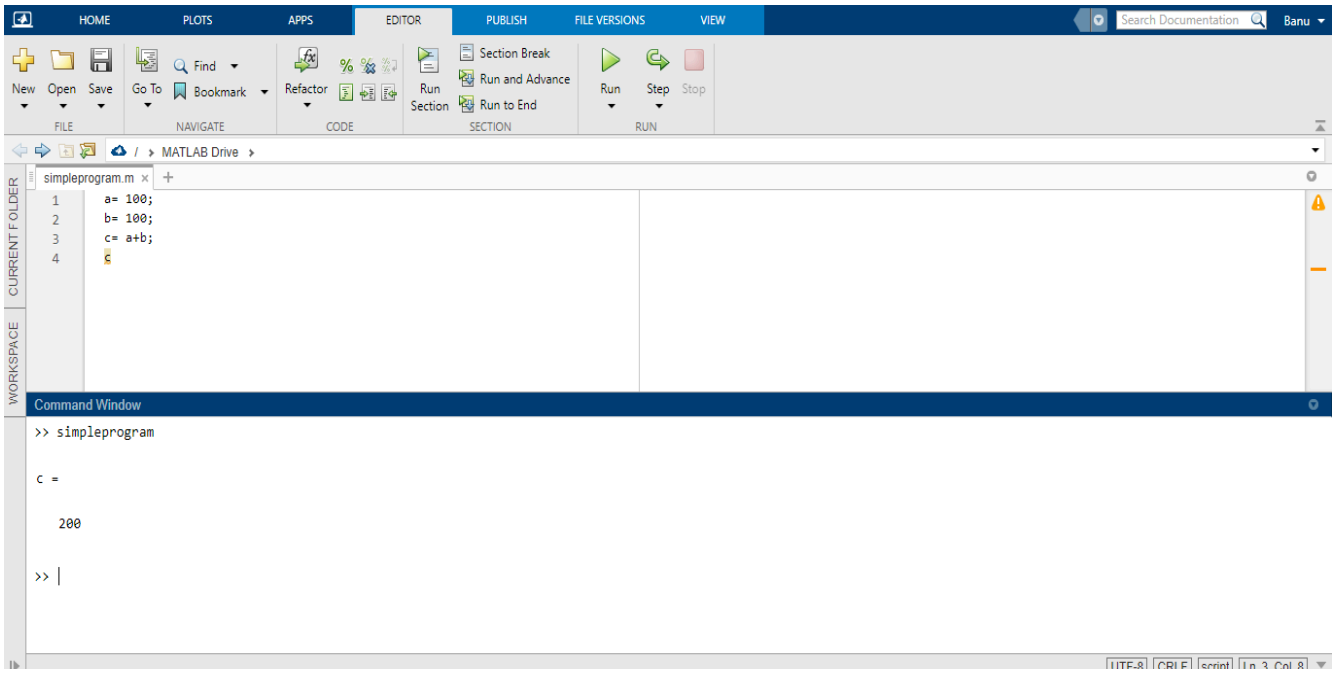
2. Editor window(to write program):



3. Save the program:



4. Output Window:



BASIC COMMANDS:

Matlab Syntax	**Functionality**
>>	Matlab is waiting for input
>> exit	Exit Matlab
>>clc	Clears the command window
>> clear	Clears all variables in the Matlab workspace
>> clear a	Clears only the variable a
>> ctrl *c*	Interrupts and stops a Matlab program or command
>> who	Lists all the variables currently in use
>>whos	Lists all the current variables and their sizes
>> %comment	Everything after % is a comment or explanation
>> k = 2	Gives k a value of 2
>> cats = 5	Gives cats a value of 5
>> k*cats	Multiplies k by cats
>> k	Check to see what is the value of k
>> a = [1 2 3]	Defines a 3-D vector with name of "a"
>> b = [2 1 2]	Defines another vector with the name "b"
>> c = a + b	Creates c by vector addition of a & b
>> d = a – b	Creates d by vector subtraction of b from a
>> e = k*a	Creates e by multiplying the vector "a" by scalar "k"
>> f = a + b;	The semicolon suppresses the printing of the output on the screen
>> h = a*b'	Multiplies vector a & vector b using the dot product
>>z=a.*b	element by element multiplication of vectors 'a' and 'b'

Commands for Managing a Session:

Command	Purpose
Clc	Clears command window.
Clear	Removes variables from memory.
Exist	Checks for existence of file or variable.
global	Declares variables to be global.
help	Searches for a help topic.
lookfor	Searches help entries for a keyword.
quit	Stops MATLAB.
who	Lists current variables.
whos	Lists current variables (long display).

Commands for Working with the System

Command	Purpose
cd	Changes current directory.
date	Displays current date.

delete	Deletes a file.
diary	Switches on/off diary file recording.
dir	Lists all files in current directory.
load	Loads workspace variables from a file.
path	Displays search path.
pwd	Displays current directory.
save	Saves workspace variables in a file.
type	Displays contents of a file.
what	Lists all MATLAB files in the current directory.
wklread	Reads .wk1 spreadsheet file.

Input and Output Commands:

Command	**Purpose**
disp	Displays contents of an array or string.
fscanf	Read formatted data from a file.
format	Controls screen-display format.
fprintf	Performs formatted writes to screen or file.

input	Displays prompts and waits for input.
;	Suppresses screen printing.

The **fscanf** and **fprintf** commands behave like scanf and printf functions. They support the following format codes −

Format Code	**Purpose**
%s	Format as a string.
%d	Format as an integer.
%f	Format as a floating point value.
%e	Format as a floating point value in scientific notation.
%g	Format in the most compact form: %f or %e.
\n	Insert a new line in the output string.
\t	Insert a tab in the output string.

The format function has the following forms used for numeric display –

Format Function	**Display up to**
format short	Four decimal digits (default).
format long	16 decimal digits.

format short e	Five digits plus exponent.
format long e	16 digits plus exponents.
format bank	Two decimal digits.
format +	Positive, negative, or zero.
format rat	Rational approximation.
format compact	Suppresses some line feeds.
format loose	Resets to less compact display mode.

Vector, Matrix and Array Commands:

The following table shows various commands used for working with arrays, matrices and vectors −

Command	**Purpose**
cat	Concatenates arrays.
find	Finds indices of nonzero elements.
length	Computes number of elements.
linspace	Creates regularly spaced vector.
logspace	Creates logarithmically spaced vector.
max	Returns largest element.

min	Returns smallest element.
prod	Product of each column.
reshape	Changes size.
size	Computes array size.
sort	Sorts each column.
sum	Sums each column.
eye	Creates an identity matrix.
ones	Creates an array of ones.
zeros	Creates an array of zeros.
cross	Computes matrix cross products.
dot	Computes matrix dot products.
det	Computes determinant of an array.
inv	Computes inverse of a matrix.
pinv	Computes pseudoinverse of a matrix.
rank	Computes rank of a matrix.
rref	Computes reduced row echelon form.
cell	Creates cell array.

celldisp	Displays cell array.
cellplot	Displays graphical representation of cell array.
num2cell	Converts numeric array to cell array.
deal	Matches input and output lists.
iscell	Identifies cell array.

Plotting Commands:

Command	**Purpose**
axis	Sets axis limits.
fplot	Intelligent plotting of functions.
grid	Displays gridlines.
plot	Generates xy plot.
print	Prints plot or saves plot to a file.
title	Puts text at top of plot.
xlabel	Adds text label to x-axis.
ylabel	Adds text label to y-axis.
axes	Creates axes objects.

close	Closes the current plot.
close all	Closes all plots.
figure	Opens a new figure window.
gtext	Enables label placement by mouse.
hold	Freezes current plot.
legend	Legend placement by mouse.
refresh	Redraws current figure window.
set	Specifies properties of objects such as axes.
subplot	Creates plots in subwindows.
text	Places string in figure.
bar	Creates bar chart.
loglog	Creates log-log plot.
polar	Creates polar plot.
semilogx	Creates semilog plot. (logarithmic abscissa).
semilogy	Creates semilog plot. (logarithmic ordinate).
stairs	Creates stairs plot.
stem	Creates stem plot.

Si

```
%simple calculation
y=[0 1 2 3 4 5];
x=[0 1 3 5 7 9];
a = x + y;
b = 5 * x;
plot (a,b)
```

OPERATION WITH VARIABLES,CHARACTERS,STRINGS:

String Arrays

string	String array
strings	Create string array with no characters
join	Combine strings
plus	Add numbers, append strings

Simple example:

```
str = "A horse! A horse! My kingdom for a horse!"

str =
"A horse! A horse! My kingdom for a horse!"

Remove the exclamation point.

str = erase(str,"!")

str =
"A horse A horse My kingdom for a horse"

Convert all letters in str to lowercase characters.

str = lower(str)

str =
"a horse a horse my kingdom for a horse"
```

Character Arrays

char	Character array
cellstr	Convert to cell array of character vectors
blanks	Create character array of blanks
newline	Create newline character

Simple example:

Convert a numeric array to a character array.

```
A = [77 65 84 76 65 66];
C = char(A)
C = 'MATLAB'
```

V. Programs to practice operations with variables, Characters and Strings

1. Write a program to use variable:

NoOfStudents = 6000;

TeachingStaff = 150;

NonTeachingStaff = 20;

Total = NoOfStudents + TeachingStaff + NonTeachingStaff;

disp(Total);

Output:

simpleprogram.m

```
1   NoOfStudents = 6000;
2   TeachingStaff = 150;
3   NonTeachingStaff = 20;
4
5   Total = NoOfStudents + TeachingStaff + NonTeachingStaff;
6   disp(Total);
```

```
>> simpleprogram
        6170
```

2. Write a program to Concatenate Two Character Vectors.

```
>> s1 = 'Good';
s2 = 'morning';
s = strcat(s1,s2)

s =

     'Goodmorning'
```

3. Write a program to Concatenate Two String Arrays.

```
>> str1 = ["Agri ","culture "];
str2 = ["Horti","culture"];
str = strcat(str1,str2)

str =

  1×2 string array

    "Agri Horti"    "culture culture"
```