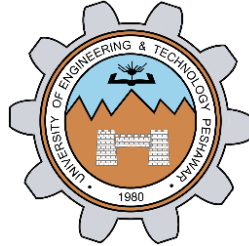


INTRODUCTION TO MATLAB

LAB # 01



Spring 2023

CSE301L Signals & Systems Lab

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

Engr. Sumayyea Salahuddin

Date:

March 3, 2023

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Lab Objective(s):

Objectives of this Lab are;

- Introduction to MATLAB
- MATLAB Environment
- MATLAB Help
- Variable arithmetic
- Built in Mathematical Functions
- Input and display
- Timing functions
- Introduction to M-files

MATLAB:

The objective of this lab is to introduce students to the MATLAB programming language and its capabilities. Students will learn about the different types of data that MATLAB can handle, such as matrices, arrays, and structures. They will also learn about the basic syntax of MATLAB, including how to write and run simple programs.

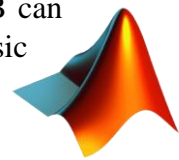


Figure 1:
MATLAB Logo

MATLAB Environment:

The MATLAB environment consists of several different components, including the Command Window, Workspace, and Editor. The Command Window is where students can enter commands and see their output. The Workspace shows all the variables that are currently defined in the program. The Editor is a window where students can write and save their MATLAB programs.

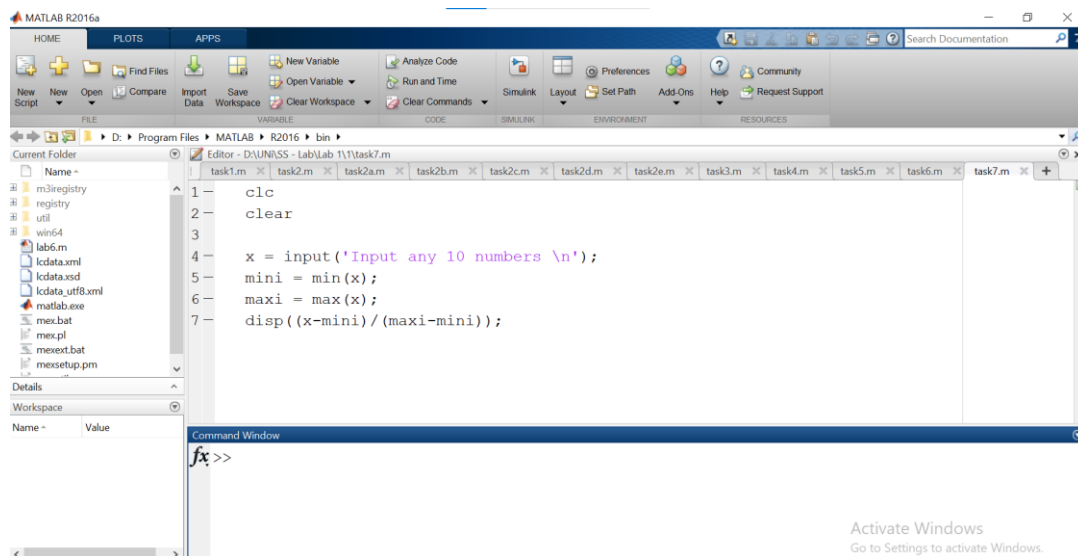
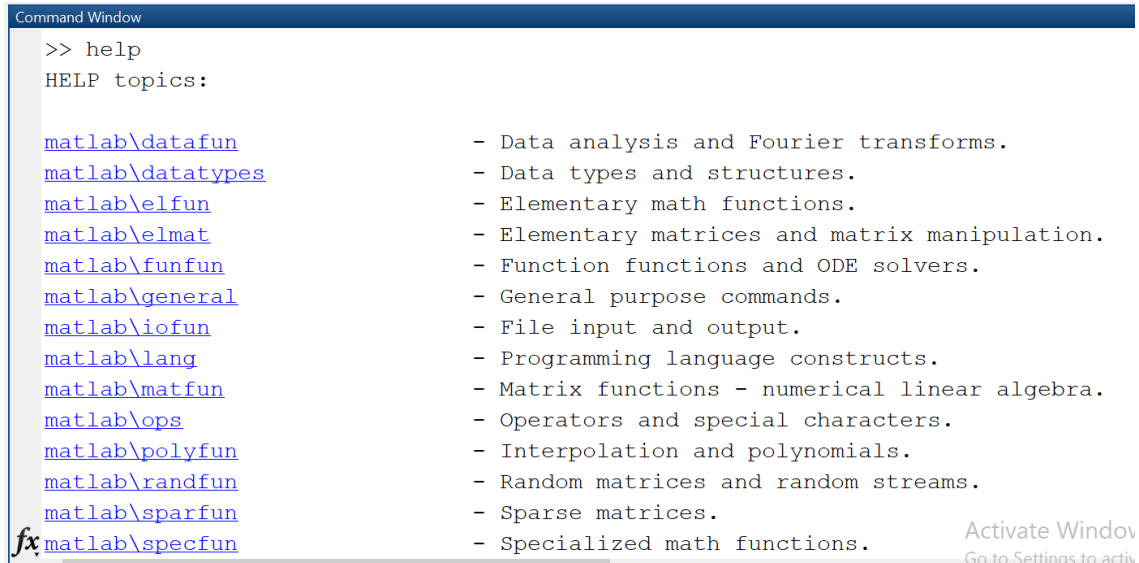


Figure 2: MATLAB Environment

MATLAB Help:

MATLAB has extensive built-in documentation and help features. This objective will teach students how to access and use MATLAB help to get more information about the syntax and usage of MATLAB functions.



```
Command Window
>> help
HELP topics:

matlab\datafun           - Data analysis and Fourier transforms.
matlab\datatypes         - Data types and structures.
matlab\elfun             - Elementary math functions.
matlab\elmat             - Elementary matrices and matrix manipulation.
matlab\funfun            - Function functions and ODE solvers.
matlab\general           - General purpose commands.
matlab\iofun             - File input and output.
matlab\lang              - Programming language constructs.
matlab\matfun            - Matrix functions - numerical linear algebra.
matlab\ops               - Operators and special characters.
matlab\polyfun           - Interpolation and polynomials.
matlab\randfun           - Random matrices and random streams.
matlab\sparfun           - Sparse matrices.
fx matlab\specfun        - Specialized math functions.
```

Figure 3: MATLAB Help

Variable arithmetic:

MATLAB allows for mathematical operations on variables. This objective will cover basic arithmetic operations such as addition, subtraction, multiplication, and division. Students will also learn about more advanced operations such as matrix multiplication and element-wise operations.

Built-in Mathematical Functions:

MATLAB has a wide range of built-in mathematical functions for various numerical operations. Examples of built-in mathematical functions include trigonometric functions, logarithmic functions, and exponential functions.

Input and Display:

MATLAB allows for user input and output through the Command Window. This objective will teach students how to prompt for input and display output. Students will learn how to use the "input" function to prompt for user input and the "disp" function to display output.

Timing Functions:

Timing functions are used to measure the execution time of a program or function. MATLAB has functions to measure time taken by a program or function to execute. This objective will teach students how to use these functions to measure execution time. Examples of timing functions include "tic" and "toc" for measuring the time taken to execute a program or function.

M-files:

M-files are MATLAB scripts or functions saved as plain text files. This objective will introduce students to creating and using M-files in MATLAB. Students will learn how to create and save M-files, and how to run them from the MATLAB command line. They will also learn how to write simple functions in MATLAB and call them from other scripts or functions.

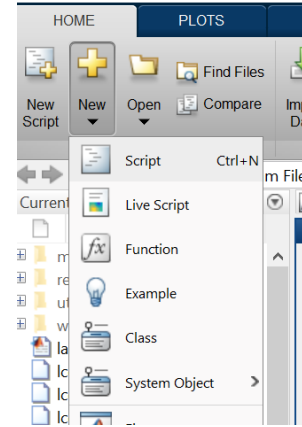


Figure 4: Creating an M-file

Task # 01:

- Matlab stores numeric data as double-precision floating-point by default. To store data as an 8- bit integer, `int8` (a conversion function) can be used. Type the sample code in MatLab command window:

```
>> x = 26
>> whos
>> y = int8(x)
>> whos
```

What difference do you see? State your findings. (Also try `uint16`, `uint32`, `uint64`)

- Take your name in the command window e.g. `name = 'Ali'`. Convert it into 8-bit integer format using the **`int8`** function.
- Use the formatting commands present in MatLab to convert the system Clock to whole numbers rather than floating points.

Code:

Part a: