

INTRODUCTION TO MATLAB

Lab Report # 01



CSE301 - L Signals & Systems Lab

Submitted by: **AWAIS SADDIQUI**

Registration No: **21PWCSE1993**

Class Section: **A**

Submitted to:

Dr. Durr-e-Nayab

Department of Computer System Engineering

UET Peshawar

301L: Signals & Systems Lab**LAB ASSESSMENT RUBRICS**

Marking Criteria	Exceeds expectation (5-4)	Meets expectation (3-2)	Does not meet expectation (1)	Score
1. Realization of Experiment	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 71-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-70% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	30%
2. Ability to apply required code utility or data structure	Able to apply required data type or data structure and produce correct results. Familiarize and selects proper functions for simulation of given problem using software tools like MATLAB.	Able to apply required data type or data structure but does not produce correct results. Need guidance to select proper functions for simulation of given problem using software tools like MATLAB.	Unable to identify required data type or data structure. Incapable of selecting proper functions for simulation of given problem using software tools like MATLAB.	20%
3. Documentation	Clearly and effectively documented including descriptions of all variables/functions. Specific purpose is noted for each function, control structure, input requirements and output results.	Basic documentation including descriptions of all variables/functions. Specific purpose is noted for each function and control structure.	No documentation included.	10%

4. Ability to run/debug	Executes Matlab codes without errors, excellent user	Executes Matlab codes without errors. User prompts are	Does not execute Matlab codes due to errors.	20%
--------------------------------	--	---	--	-----

	prompts, good use of symbols, spacing in output. Thorough and organized testing has been completed and output from test cases is included.	understandable, minimum use of symbols or spacing in output. Some testing has been completed.	User prompts are misleading or nonexistent. No testing has been completed.	
5. Results compilation	Show processed results effectively by conducting simple computations and plotting using collected data	Show processed results effectively by conducting simple computations and plotting using collected data with minor error	Unable to show processed results effectively by conducting simple computations and plotting using collected data with minor error	10%
6. Efficiency	Excellent use of CPU and Memory.	Good but not smart use of CPU and Memory.	Inefficient use of CPU and Memory.	10%
7. Lab Performance (Team work and Lab etiquettes)	Actively engages and cooperates with other group members in an effective manner. Respectfully and carefully observes safety rules and procedures	Cooperates with other group members in a reasonable manner. Observes safety rules and procedures with minor deviation.	Distracts or discourages other group members from conducting the experiment. Disregards safety rules and procedures.	10%

Instructor:

Name: _____

Signature: _____

WHAT IS MATLAB:

MATLAB is a commercial MATRIX LABORATORY package. MATLAB is a high-level programming language and computing environment used for numerical analysis, data visualization and programming. It is widely used in engineering, science and mathematics fields. MATLAB allows users to perform complex mathematical calculations and data analysis tasks with ease as well as develop and deploy algorithms and models for a variety of applications.

Lab objectives:

MATLAB will be used extensively in all the succeeding labs. The goal of this first lab is to gain familiarity with MATLAB and build some basic skills in the MATLAB language. Some specific topics covered in this lab are:

- Introduction to MATLAB.
- MATLAB Environment (command window, editor, workspace, working folder)
- MATLAB Help
- Variable arithmetic
- Built in Mathematical Functions
- Input and display
- Timing functions (clock, Tic Toc, etime).
- Introduction to M-files (scripts etc.)

-----TASK 01-----

- a.) 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`).

Code and Output:

```
Command Window
New to MATLAB? See resources for Getting Started.

>> x=26;
>> y=int8(x)

y =

    26

>> z=int16(x)

z =

    26

>> a=int32(x)

a =

    26

>> b=int64(x)

b =

    26

>> whos
  Name      Size      Bytes  Class  Attributes

  a         1x1         4  int32
  b         1x1         8  int64
  x         1x1         8  double
  y         1x1         1  int8
  z         1x1         2  int16

fx >>
```

-----TASK 01-----

b). Take your name in the command window e.g. name = 'Ali'. Convert it into 8-bit integer format using the int8 function.

Code and Output:

```
Command Window
New to MATLAB? See resources for Getting Started.

>> name='Awais'

name =

Awais


>> y=int8(name)

y =

    65    119     97    105    115

>> whos
  Name      Size      Bytes  Class  Attributes

  name      1x5         10   char
  y          1x5          5   int8
```

 >> |

-----TASK 01-----

c). Use the formatting commands present in MATLAB to convert the system Clock to whole numbers rather than floating points.

Code and Output:

```
Command Window
New to MATLAB? See resources for Getting Started.

>> formateCommand = clock

formateCommand =

    1.0e+03 *
    2.0230    0.0020    0.0280    0.0180    0.0050    0.0007

>> ceil(formateCommand)

ans =

    2023         2         28         18         5         1

>> floor(formateCommand)

ans =

    2023         2         28         18         5         0

>> fix(formateCommand)

ans =

    2023         2         28         18         5         0

fx >> |
```

-----TASK 02-----

Create an M-File to prove any five expressions from the following:

Use etime or tic toc functions to evaluate time taken for solving each of the five chosen expressions.

MATLAB Code:

```
Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab1.m
Lab1.m  x  +
1 - clear all
2 - clc
3 - tic
4 - a=60;
5 - b=40;
6 - disp('sin(a+b)=sin(a)cos(b)+cos(a)sin(b)');
7 - LHS=sin(a+b);
8 - disp(LHS)
9 - RHS=sin(a)*cos(b)+cos(a)*sin(b);
10 - disp(RHS);
11 - disp('sin(a-b)=sin(a)cos(b)-cos(a)sin(b)');
12 - LHS=sin(a-b);
13 - disp(LHS)
14 - RHS=sin(a)*cos(b)-cos(a)*sin(b);
15 - disp(RHS);
16 - disp('cos(a+b)=cos(a)cos(b)-sin(a)sin(b)');
17 - LHS=cos(a+b);
18 - disp(LHS);
19 - RHS=cos(a)*cos(b)-sin(a)*sin(b);
20 - disp(RHS)
21 - disp('tan(a+b)=tan(a)+tan(b)/(1-tan(a)tan(b))');
22 - LHS=tan(a+b);
23 - disp(LHS);
24 - RHS=tan(a)+tan(b)/(1-tan(a)*tan(b));
25 - disp(RHS);
26 - disp('tan(a-b)=tan(a)-tan(b)/(1+tan(a)tan(b))');
27 - LHS=tan(a-b);
28 - disp(LHS);
29 - RHS=tan(a)-tan(b)/(1+tan(a)*tan(b));
30 - disp(RHS);
31 - toc
```


Task-2 Output:

```
Command Window
New to MATLAB? See resources for Getting Started.

sin(a+b)=sin(a)cos(b)+cos(a)sin(b)
-0.5064

-0.5064

sin(a-b)=sin(a)cos(b)-cos(a)sin(b)
0.9129

0.9129

cos(a+b)=cos(a)cos(b)-sin(a)sin(b)
0.8623

0.8623

tan(a+b)=tan(a)+tan(b)/1-tan(a)tan(b)
-0.5872

-0.5029

tan(a-b)=tan(a)-tan(b)/1+tan(a)tan(b)
2.2372

2.0590

Elapsed time is 0.001005 seconds.
fx >>
```

-----TASK 03-----

Write a CGPA Calculator program using M-File: Design a transcript for your second-semester result i.e. take grade points and credit hours of each subject as input from user and store in variables. Take the product of each subject grade points with its credit hours and divide by total 13 credit hours in order to evaluate CGPA. Show the results in the form of well-designed transcript using disp and input commands. Use the following table to display equivalent grades for each grade point:

Code:

```
Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab_01\Lab1Task3.m*
Lab1Task3.m* x +
1 - clc
2 - %Transcript
3 - disp('Grade          Grade Point');
4 - disp('=====');
5 - disp('A          4.00');
6 - disp('A-         3.67');
7 - disp('B+         3.33');
8 - disp('B          3.00');
9 - disp('B-         2.67');
10 - disp('C+         2.33');
11 - disp('C          2.00');
12 - disp('C-         1.67');
13 - disp('D+         1.33');
14 - disp('D          1.00');
15 - disp('F          0');
16 - disp('=====');
17 - %Subject-1
18 - Grade_1=input('Circuit & System-1 Grade = ');
19 - cHour_1=input('Credit Hour = ');
20 - GPA_1=Grade_1*cHour_1;
21 - %Subject-2
22 - Grade_2=input('Circuit & System-1 Lab Grade = ');
23 - cHour_2=input('Credit Hour = ');
24 - GPA_2=Grade_2*cHour_2;
25 - %Subject-3
26 - Grade_3=input('Computer Programming Grade = ');
27 - cHour_3=input('Credit Hour = ');
28 - GPA_3=Grade_3*cHour_3;
29 - %Subject-4
30 - Grade_4=input('Computer Programming Lab Grade = ');
31 - cHour_4=input('Credit Hour = ');
32 - GPA_4=Grade_4*cHour_4;
33 - %Subject-5
34 - Grade_5=input('DE Grade = ');
35 - cHour_5=input('Credit Hour = ');
36 - GPA_5=Grade_5*cHour_5;
37 - %Subject-6
38 - Grade_6=input('EDG Grade = ');
39 - cHour_6=input('Credit Hour = ');
40 - GPA_6=Grade_6*cHour_6;
41 - %Subject-7
42 - Grade_7=input('EDG Lab Grade = ');
43 - cHour_7=input('Credit Hour = ');
44 - GPA_7=Grade_7*cHour_7;
45 - %Subject-8
46 - Grade_8=input('CPS Lab Grade = ');
47 - cHour_8=input('Credit Hour = ');
48 - GPA_8=Grade_8*cHour_8;
49 -
50 - totalCreditHours=cHour_1+cHour_2+cHour_3+cHour_4+cHour_5+cHour_6+cHour_7+cHour_8;
51 - Total_GPA=GPA_1+GPA_2+GPA_3+GPA_4+GPA_5+GPA_6+GPA_7+GPA_8;
52 - CGPA=Total_GPA/totalCreditHours;
53 - disp('SGPA');
54 - disp(CGPA);
```

Task-3 Output:

Current Folder

Workspace

Name	Value
CGPA	2.8950
cHour_1	3
cHour_2	1
cHour_3	3
cHour_4	1
cHour_5	3
cHour_6	2
cHour_7	1
cHour_8	2
GPA_1	9.9900
GPA_2	3.6700
GPA_3	6.9900
GPA_4	3.6700
GPA_5	9
GPA_6	5.3400
GPA_7	3
GPA_8	4.6600
Grade_1	3.3300
Grade_2	3.6700
Grade_3	2.3300
Grade_4	3.6700
Grade_5	3
Grade_6	2.6700
Grade_7	3
Grade_8	2.3300
Total_GPA	46.3200
totalCreditHours	16

Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab_01\Lab1Task3.m

Command Window

New to MATLAB? See resources for [Getting Started](#).

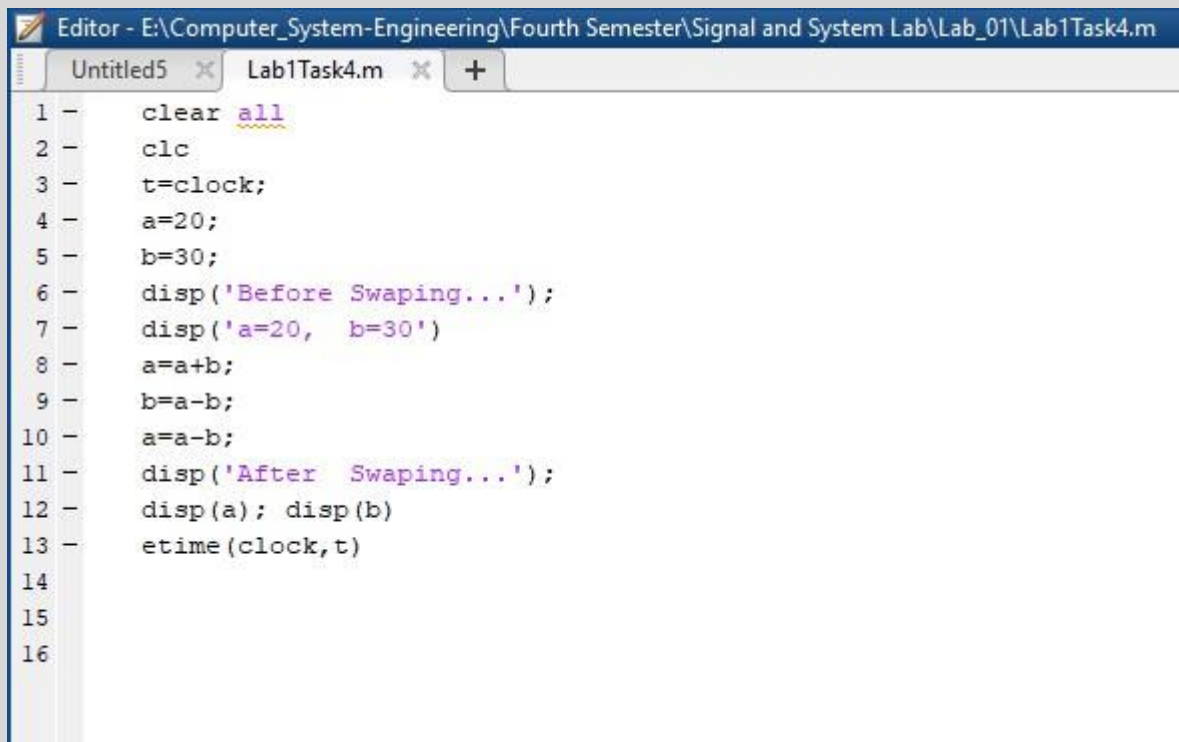
Grade	Grade Point
A	4.00
A-	3.67
B+	3.33
B	3.00
B-	2.67
C+	2.33
C	2.00
C-	1.67
D+	1.33
D	1.00
F	0

Circuit & System-1 Grade =3.33
Credit Hour = 3
Circuit & System-1 Lab Grade =3.67
Credit Hour = 1
Computer Programming Grade = 2.33
Credit Hour3
Computer Programming Lab Grade = 3.67
Credit Hour = 1
DE Grade = 3.00
Credit Hour = 3
EDG Grade = 2.67
Credit Hour = 2
EDG Lab Grade = 3.00
Credit Hour = 1
CPS Lab Grade = 2.33
Credit Hour = 2
2.8950

fx >>

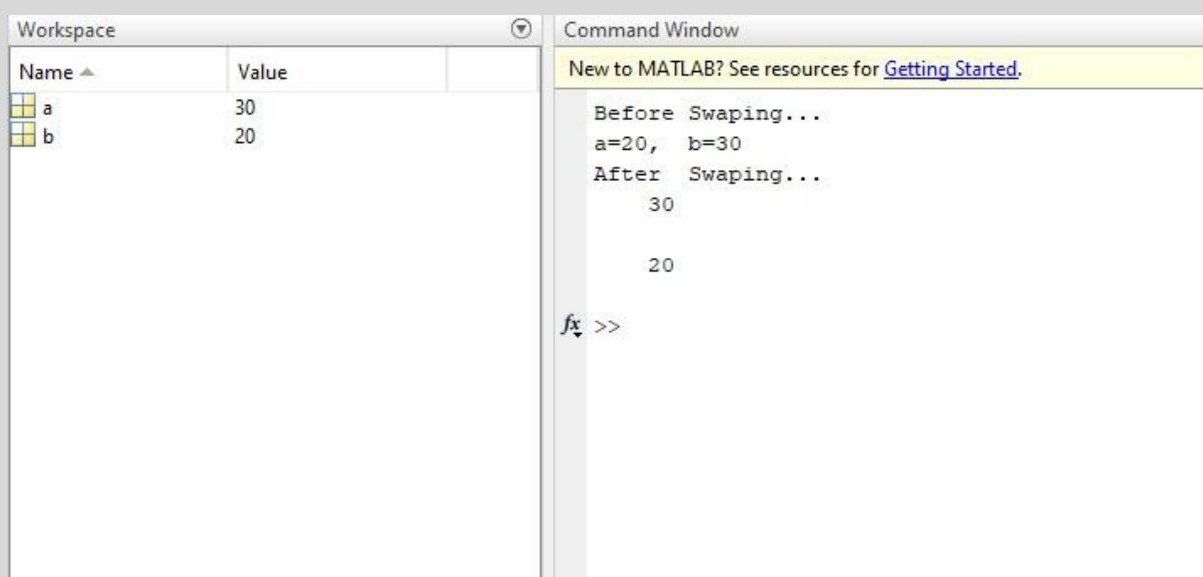
-----TASK 04-----

Write a simple code to swap the values of two variables of double type using M-file. Create the logic in such a way that no third variable is used. Show the etime for this code.



```
Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab_01\Lab1Task4.m
Untitled5  Lab1Task4.m  +
1 - clear all
2 - clc
3 - t=clock;
4 - a=20;
5 - b=30;
6 - disp('Before Swaping...');
7 - disp('a=20, b=30')
8 - a=a+b;
9 - b=a-b;
10 - a=a-b;
11 - disp('After Swaping...');
12 - disp(a); disp(b)
13 - etime(clock,t)
14
15
16
```

Task-4 Output:



Workspace	
Name	Value
a	30
b	20

Command Window

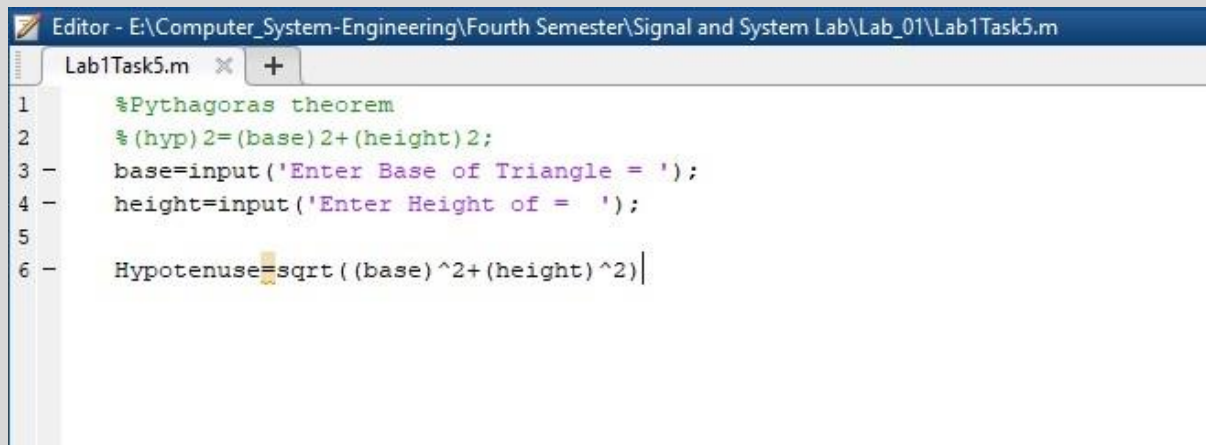
New to MATLAB? See resources for [Getting Started](#).

```
Before Swaping...
a=20, b=30
After Swaping...
    30
    20

fx >>
```

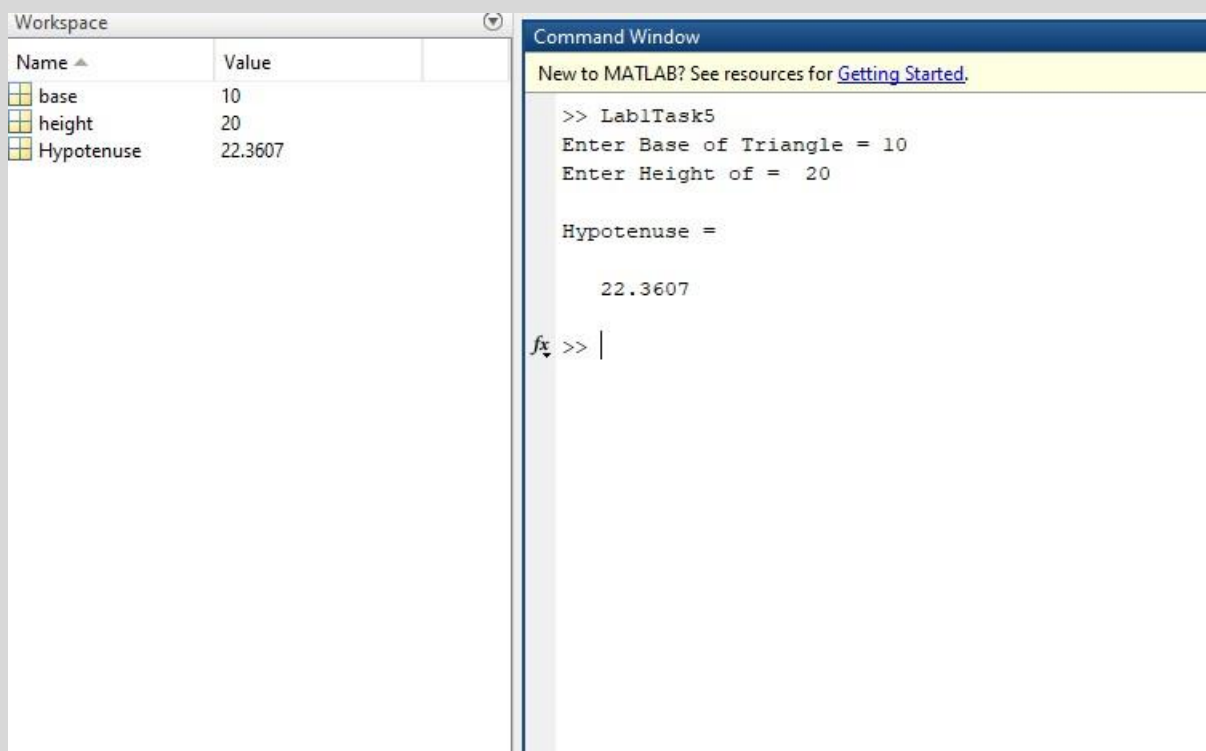
-----TASK 05-----

Implement the Pythagoras theorem in MATLAB that takes input from the user.



```
Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab_01\Lab1Task5.m
Lab1Task5.m  x  +
1      %Pythagoras theorem
2      % (hyp)2=(base)2+(height)2;
3      base=input('Enter Base of Triangle = ');
4      height=input('Enter Height of = ');
5
6      Hypotenuse=sqrt((base)^2+(height)^2)
```

Task-5 output:



The image shows the MATLAB interface with the Workspace and Command Window. The Workspace window displays the variables created during the execution: 'base' with a value of 10, 'height' with a value of 20, and 'Hypotenuse' with a value of 22.3607. The Command Window shows the execution of the script 'Lab1Task5', which prompts the user to enter the base and height, and then displays the calculated hypotenuse value.

Name	Value
base	10
height	20
Hypotenuse	22.3607

```
>> Lab1Task5
Enter Base of Triangle = 10
Enter Height of = 20

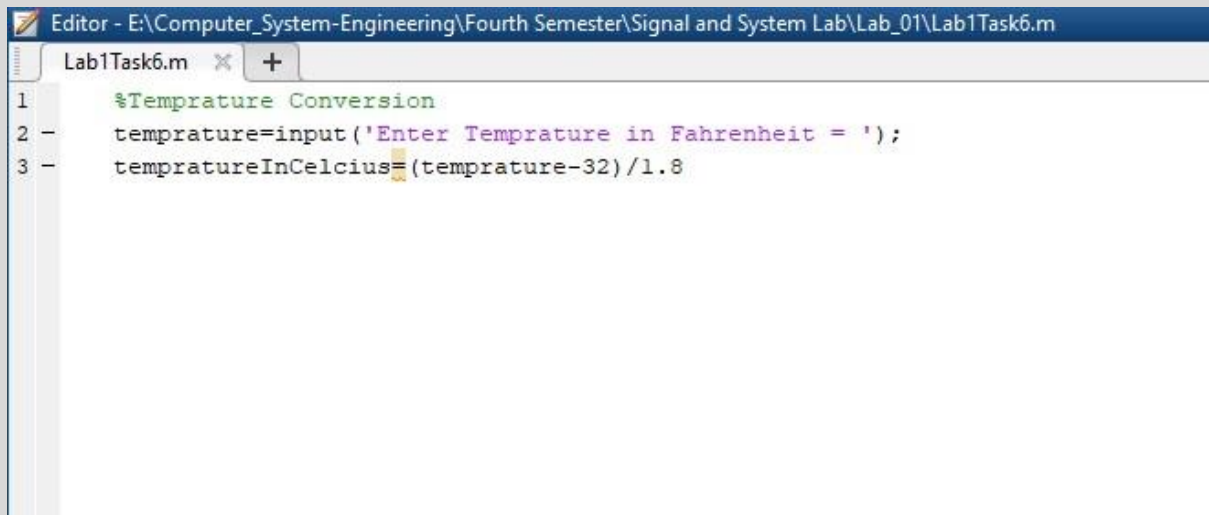
Hypotenuse =

    22.3607

fx >> |
```

-----TASK 06-----

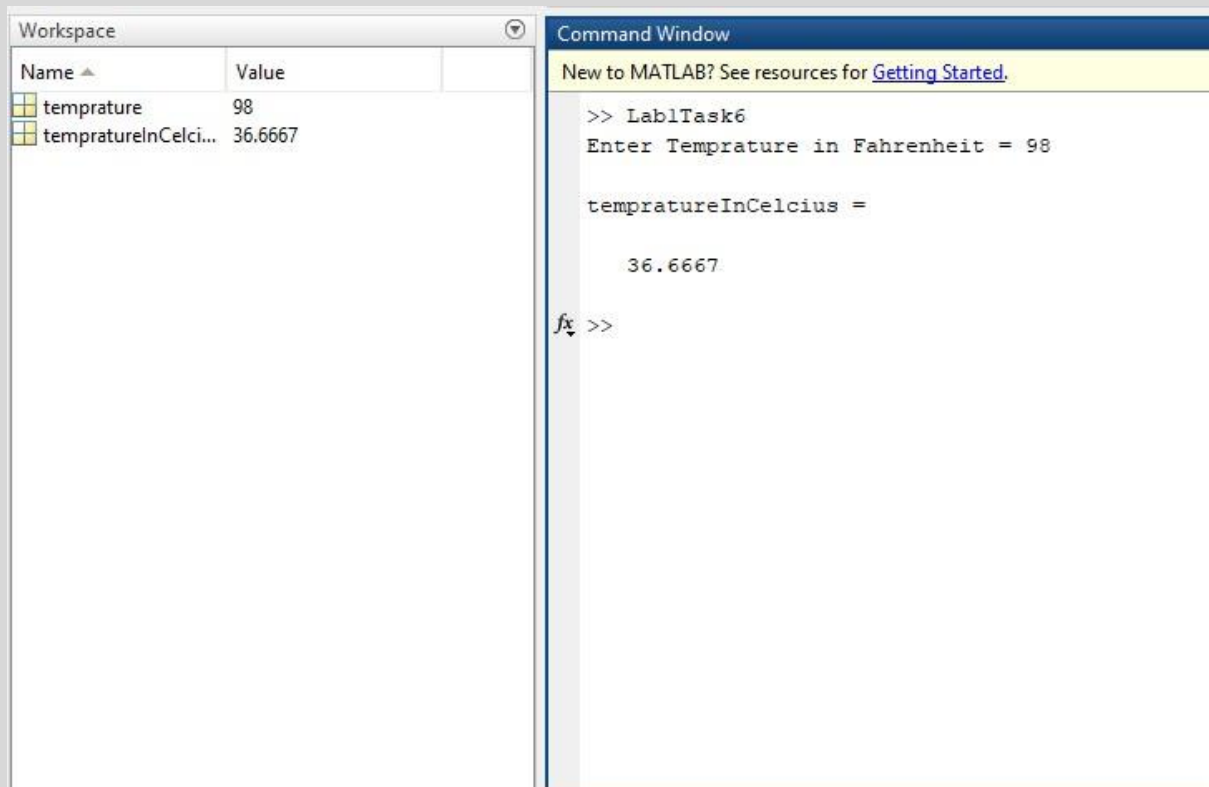
Implement a temperature conversion scenario in MATLAB that takes the temperature from the user in Fahrenheit and displays the output in Centigrade.



The image shows the MATLAB Editor window with a script named 'Lab1Task6.m'. The script contains the following code:

```
1 %Temperature Conversion
2 - temprature=input('Enter Temprature in Fahrenheit = ');
3 - tempratureInCelcius=(temprature-32)/1.8
```

Task-6 Output:



The image shows the MATLAB Command Window and Workspace. The Command Window displays the execution of the script, and the Workspace shows the variables created.

Command Window:

```
>> Lab1Task6
Enter Temprature in Fahrenheit = 98

tempratureInCelcius =

    36.6667

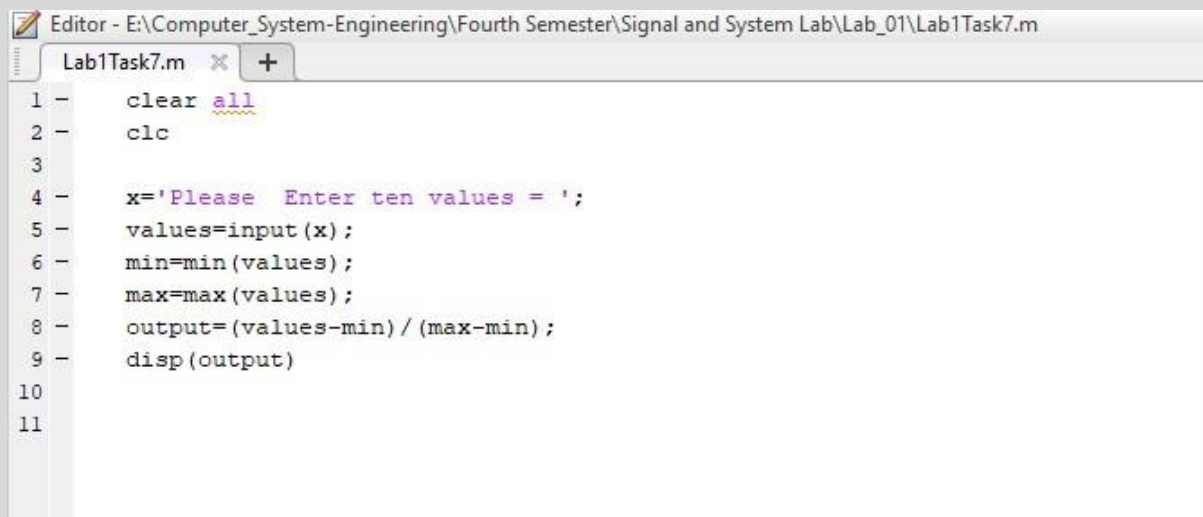
fx >>
```

Workspace:

Name	Value
temprature	98
tempratureInCelci...	36.6667

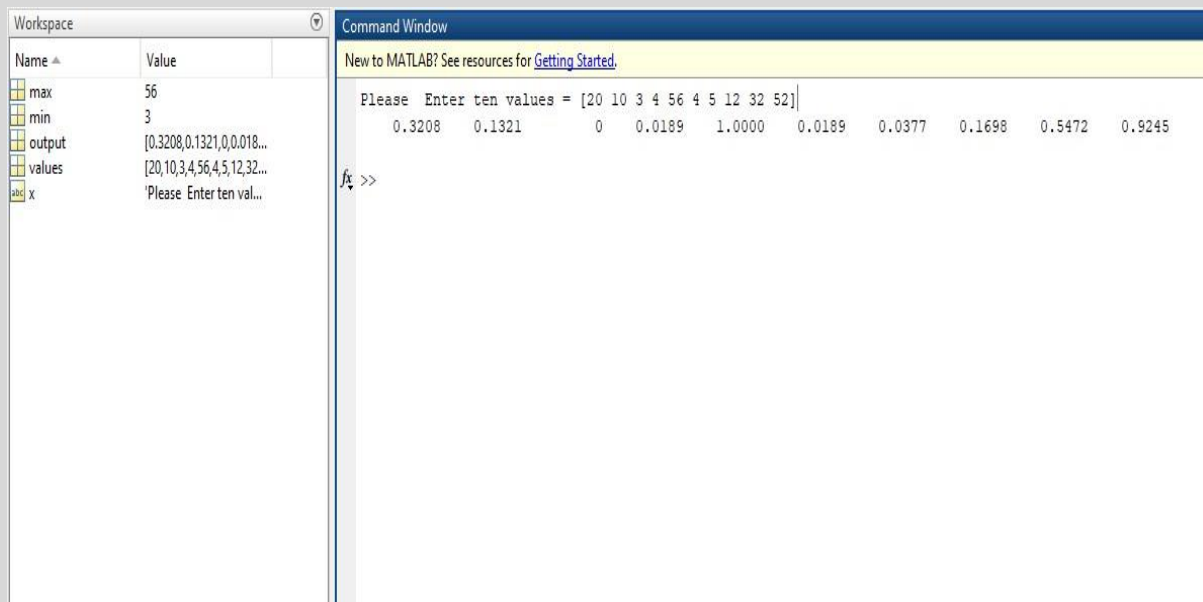
-----TASK 07-----

Devise an algorithm in MATLAB that takes ten inputs from the user and normalizes them between [0-1]. Hints: Find the pair-wise max (maxi) and min (mini) of ten numbers using the max and min built-in command. Find the normalized value for each input using formula $(\text{input} - \text{mini}) / (\text{maxi} - \text{mini})$. Note: Do not use loops or if else structures.



```
Editor - E:\Computer_System-Engineering\Fourth Semester\Signal and System Lab\Lab_01\Lab1Task7.m
Lab1Task7.m x +
1 - clear all
2 - clc
3
4 - x='Please Enter ten values = ';
5 - values=input(x);
6 - min=min(values);
7 - max=max(values);
8 - output=(values-min)/(max-min);
9 - disp(output)
10
11
```

Task-7 Output:



Workspace

Name	Value
max	56
min	3
output	[0.3208, 0.1321, 0, 0.0189, 1.0000, 0.0189, 0.0377, 0.1698, 0.5472, 0.9245]
values	[20, 10, 3, 4, 56, 4, 5, 12, 32, 52]
x	'Please Enter ten val...'

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
Please Enter ten values = [20 10 3 4 56 4 5 12 32 52]
0.3208 0.1321 0 0.0189 1.0000 0.0189 0.0377 0.1698 0.5472 0.9245
fx >>
```