

**University of Al–Mustansiriyah, College of Engineering**  
**Department of Mechanical Engineering**

# **Computer Skills**

(Matlab Programming)

**(A course intended for first class students)**

First Semester, Academic Year 2020–2021

Lecturer:

**Dr. Aouf A. Al–Tabbakh**

**(Ph.D. Mech. Thermal Sci.)**

# Chapter One

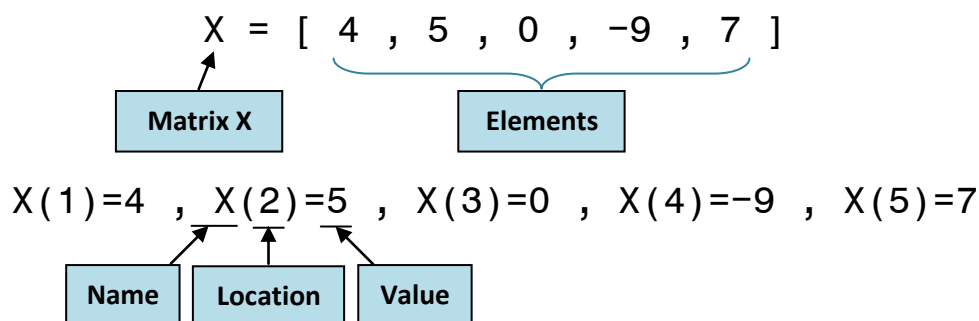
## Introduction

**Matlab** is a short for **Matrix Laboratory**. The language is first introduced in 1984 by *Mathworks Company* and optimized to deal with matrices.

### 1.1 Some important definitions

**Computer Programming:** the process of writing programs to instruct the computer to perform certain logical or mathematical tasks.

**Matrix:** is a group of numbers at which every element in the group has a value and location.

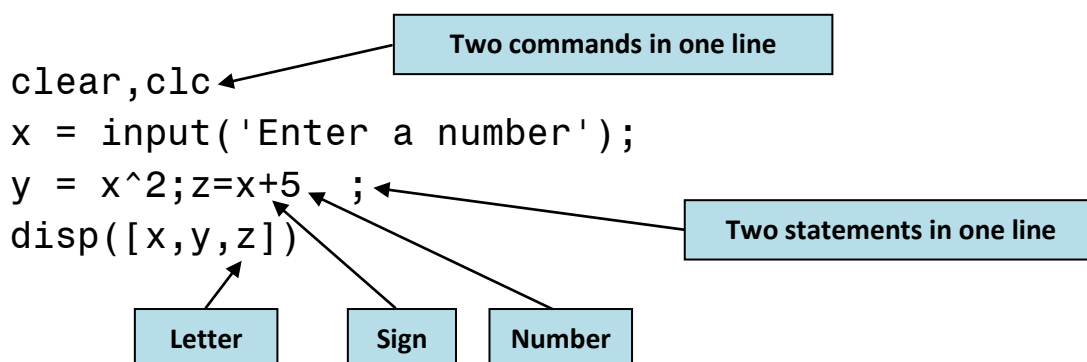


**Program:** a text of several sentences (statements) written in one line or several lines to instruct the computer to do certain task.

**Statement:** a single line in a program made of: commands, symbols and spaces.

**Command:** a word or several related words to do certain mathematical or logical tasks.

**Symbol:** a character that is either a letter, a number or a mathematical or punctuation sign.



**Data:** any information communicated with computer. Data are either numerical (digital) or text (string).

**Variable:** a symbol used to store data in a program.

## 1.2. General rules in variable definition

1. Must start with a letter.
2. Any letter, number or underscore sign ( \_ ) can be used after the first letter in the variable name.
3. Some words cannot be used as variable names because they are reserved for command names. To know them the command **iskeyword** is used.
4. Built-in functions names can be used as variable names. But their original function is temporarily disabled. To restore the original function the command **clear** followed by the variable name is used.

## 1.3 Parts of a program

Any program is composed of three main parts

**Input Part:** which is associated with entering the data for the subsequent processing by the program.

**Processing Part:** this is the main part that occupies most of the program size. It contain:-

- Mathematical equations
- Logical comparisons (conditions)
- Repetition loops

**Output Part:** this part is responsible on presenting or displaying the results of the program in conceivable and clear format (design).

## 1.4 Matlab Graphical User Interface (GUI)

**Command Window:** the main window for writing simple statements and for displaying the results.

**Editor:** the window to write systematic programs that can be saved for future use.

**Command History:** a window that records all jobs performed in other windows.

**Current Folder:** a browser to select the folder containing the main program, the associated functions and results files.

**Workspace:** this window displays list of all variables used in the program with complete information about each variable.

**Graph windows:** each graph generated by Matlab is displayed in its suitable window which is not the command window.

## Chapter Two

### Data Input Methods

The first part of any program is the input part where the data intended to be processed by the program is entered. Any variable in the program must be defined (initialized) before being used. Undefined (uninitialized) variable generates an error when running the program. Even the variable of zero value must be initialized.

**Note:** Matlab code is case sensitive, that is, lowercase letters (small letters) are different than uppercase letters (capital letters); e.g.: –

```
X = 5 ; x = 7 ; R = 2 ; r = 4 ;
```

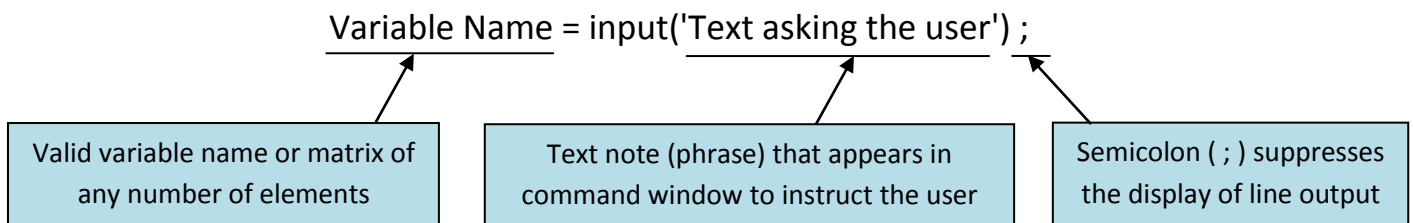
Are four different variables

**Note:** Commands and built-in functions names must always be written in lowercase, e.g.; disp **not** Disp, sin **not** SIN

There are five basic methods to supply a program (code) with a data depending on the number and type of the data, as follows: –

#### **2.1 The command (input)**

When the data are unknown to the programmer and can assume any value, then they cannot be defined within the program. Rather, the general data are supplied by the program user after running (executing) the program. The command telling the computer to expect data after running the program is (input), written as follows: –



#### **Ex.**

```
x = input(' Please enter a value = ') ;  
y = input(' Enter four numbers ') ;  
z = x + y;  
disp(z)
```

after running: -

```
Please enter a value = 2 ↵  
Enter four numbers [3,1,-5,4]  
5 3 -3 6
```

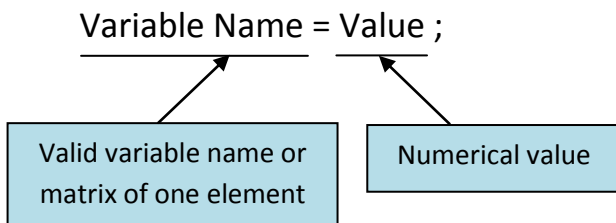
or: -

```
Please enter a value = 2 ↵  
Enter four numbers [3;1;-5;4]  
5  
3  
-3  
6
```

**Note:** Semicolon ( ; ) placed between numbers is used to display the next output in a new line in the command window.

## 2.2 Entering constant and initial values

Any variable used in the program must be initialized or assigned to an initial value within the program text, in addition to any constants that do not change during processing. The general format of variable definition is as follows: -



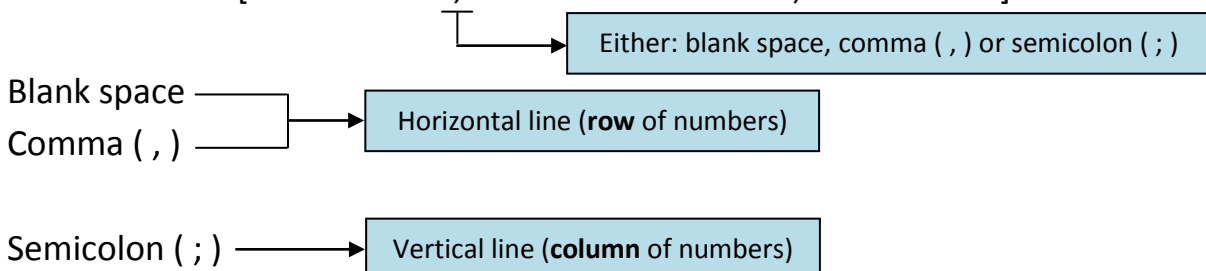
**Ex.**

```
x=1 ; y_4=22.8 ; Ahmed_2002=85
```

## 2.3 Entering arbitrary numbers

Arbitrary numbers can be supplied to the program as one group (matrix) using square brackets as follows: -

Matrix name = [ first number , second number ... , nth number ]



**Ex.**

x=[2 4 8 -3] ↵ spaces  
x = 2 4 8 -3

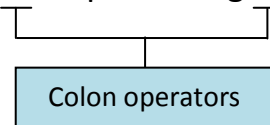
x=[2,4,8,-3] ↵ commas  
x = 2 4 8 -3

x=[2;4;8;-3] ↵ semicolons  
x = 2  
4  
8  
-3

**2.4 Entering arranged numbers using colon operator ( : )**

It is often required to enter numbers that are spaced by constant or fixed intervals. For this purpose, colon operator ( : ) is used as follows: –

Matrix name = start value : step of change : end value



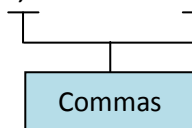
**Ex.**

x=1:10 ↵ → x = 1 2 3 4 5 6 7 8 9 10  
y=0:5:20 ↵ → y = 0 5 10 15 20  
z=1:0.2:2 ↵ → z = 1 1.2 1.4 1.6 1.8 2  
A=100:-25:0 ↵ → A = 100 75 50 25 0  
B=4:-0.8:0 ↵ → B = 4 3.2 2.4 1.6 0.8 0  
C=5:-1:0 ↵ → C = 5 4 3 2 1 0

**2.5 Entering arranged numbers using the command (linspace)**

The command (linspace) can be used to enter arranged data as follows: –

x=linspace(start value , end value , number of elements)



**Ex.**

x=linspace(0,10,5) ↵ → x = 0 2.5 5 7.5 10