



# Al-Rasheed University College

## Dept. of Computer Tech. Engineering

Second Class

Computer Applications ( CoAp200 )

By

Assist. Lecturer *Mohemmed Fadhil Abbas*

Fourth Lecture

# Matrix

**Example :** if  $x = [1,5,7,9,13,20,6,7,8]$ , then

- a) replace the first five elements of vector  $x$  with its maximum value
- b) reshape this vector into a 3 x 3 matrix.

**solution**

A)

```
>> x(1:5)=max(x)
```

b)

```
>> y(1,:)=x(1:3);
```

```
>> y(2,:)=x(4:6);
```

```
>> y(3,:)=x(7:9);
```

```
>> y
```

# Entering Matrix - workspace window

**Example:** Generate the following row vector

$b=[1, 2, 3, 4, 5, \dots, 9, 10],$

then transpose it to column vector.

- **solution**

```
>> b=1:10
```

```
b =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>> b=b';
```

# Expressions

Like most other programming languages, MATLAB provides mathematical

*expressions, but unlike most programming languages, these expressions involve entire*

matrices. The building blocks of expressions are:

1- **Variable**

2- **Numbers**

3- **Operators**

4- **Functions**

# Expressions

## 1- Variable

MATLAB does not require any type declarations or dimension statements. When MATLAB encounters a new variable name, it automatically creates the variable and allocates the appropriate amount of storage. If the variable already exists, MATLAB changes its contents and, if necessary, allocates new storage. For example,

```
num_students = 25
```

creates a 1-by-1 matrix named `num_students` and stores the value 25 in its single element.

Variable names consist of a letter, followed by any number of letters, digits, or underscores. MATLAB uses only the first 31 characters of a variable name. MATLAB is case sensitive; it distinguishes between uppercase and lowercase letters. **A and a are not the same variable.**

# Expressions

## 2- Numbers

MATLAB uses conventional decimal notation, with an optional decimal point and leading plus or minus sign, for numbers. *Scientific notation uses the letter ( e ) to specify a*

*power-of-ten scale factor. Imaginary numbers use either i or j as a suffix. Some examples of legal numbers are*

3 -99 0.0001 9.6397238 1.60210e-20

6.02252e23 1i 3+5j

# Expressions

## 3 - Arithmetic Operators

Operator	Description
+	Plus
-	Minus
*	Matrix multiply
.*	Array multiply
^	Matrix power
.^	Array power
\	Backslash or left matrix divide
/	Slash or right matrix divide
.\	Left array divide
./	Right array divide
()	Specify evaluation order

# Expressions

## 4- Functions

MATLAB provides a large number of standard elementary mathematical functions, including `abs`, `sqrt`, `exp`, and `sin`. Taking the square root or logarithm of a negative number is not an error; the appropriate complex result is produced automatically. MATLAB also provides many more advanced mathematical functions, including Bessel and gamma functions. Most of these functions accept complex arguments. For a list of the elementary mathematical functions, type

```
>> help elfun
```

For a list of more advanced mathematical and matrix functions, type

```
>> help specfun
```

```
>> help elmat
```



# Expressions

## 4- Functions

- Some of the functions, like `sqrt` and `sin`, are *built in*. *They are part of the MATLAB core* so they are very efficient, but the computational details are not readily accessible. Other functions, like `gamma` and `sinh`, are implemented in M-files. You can see the code and even modify it if you want.