COMPUTER APPLICATIONS

Control Flow and Operators

Lecturer Ahmed Wael

OVERVIEW

- MatLab is not only functional software, but also a programming language.
- MatLab has some decision making structures for control of command execution.
- Such control commands are known as control flow commands which they are:
  - *If – else – end*
  - *For …. end*
  - *While …. end*

if – else – end control flow command

- MatLab support the variant of "if" constract:

  if … end

  if … else … end

  if … elseif … else … end

- You should note the following:
  - "elseif" has no space between "else" and "if" (one word)
  - No semicolon (;) is needed at the end of lines containing "if", "else", "end"
  - Indentation of "if" block is not required, but facilitate the reading.

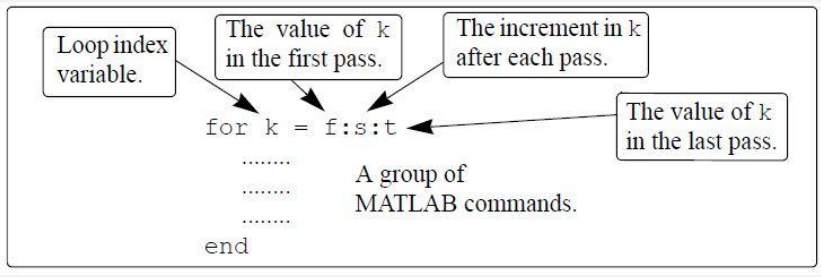
RELATION AND LOGICAL OPERATORS

Operator	Description
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
~=	Not equal to
&	AND operator
|	OR operator
~	NOT operator

EXAMPLE #1

```
% The largest number among  others program
>> h = input('Entre the first number')
>> l = input('Entre the second number')
>> k = input('Entre the third number')
>> if h > l && h > k
            fprintf(' The largest number is =  %5.2f',h)
    elseif l > h && l > k
            fprintf(' The largest number is =  %5.2f',l)
    else
           fprintf(' The largest number is =  %5.2f ',k)
    end
```

for - end LOOP



EXAMPLE #2

```
% This program illustrates the use of FOR-END loop command
>> n = input('Entre the value of itterations  ');
>> s = 0;
>> for k = 1:n
            s = s + (-1)^k*k/2^k;
   end
>> fprintf('The sume of the series is: %f', s, '%n')
```

EXAMPLE #3

A vector is given by $V = [5, 17, -3, 8, 0, -7, 12, 15, 20, -6, 6, 4, -7, 16]$. Write a program as a script file that doubles the elements that are positive and are divisible by 3 or 5, and, raises to the power of 3 the elements that are negative but greater than –5.

EXAMPLE #3

```
>> v = [5 17 -3 8 0 -7 12 15 20 -6 6 4 -2 16];
>> n = length(v);
>> for k = 1:n
        if v(k) > 0 & (rem(v(k),3) == 0 | rem(v(k),5) == 0)
                v(k) = 2*v(k);
        elseif v(k) < 0 & v(k) > -5
                v(k) = v(k)^3;
        end
    end
>> v
```