MONASH University

MONASH
ENGINEERING
ENG1060

RELATIONAL OPERATORS

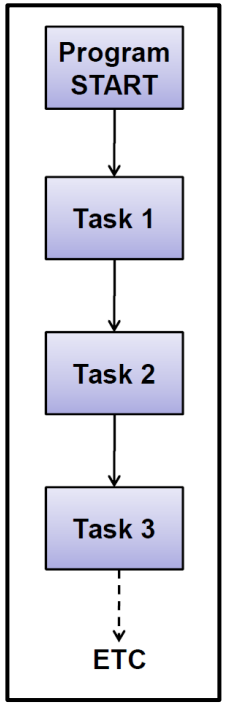
Presented by Tony Vo

Slides by Tony Vo

GROUP OF EIGHT AUSTRALIA

GETTING MATLAB TO MAKE DECISIONS

- Humans are always making decisions
- Computers are dumb by default
  - We need to tell a computer what to do and when to do it
- So far, we have been instructing MATLAB sequentially
  - Linear, one command at a time with no conditions



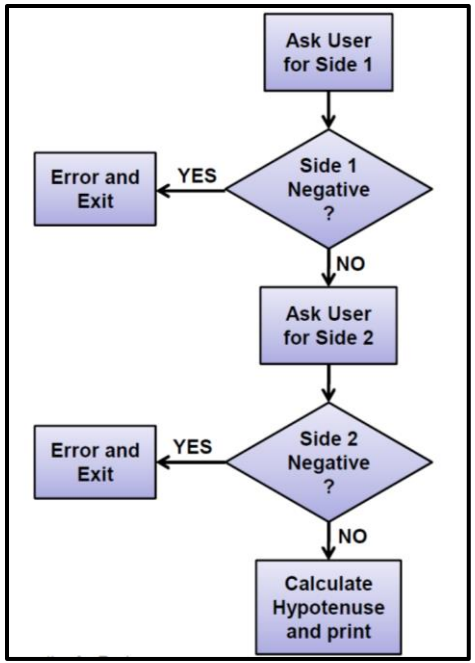
INTERACTIVE PYTHAGORAS M-FILE


```
% Written by Tony Vo, ID: 12345678
% Date created 19/07/2015
% Interactive Pythagoras
clear all; close all; clc;

%getting the information from the user
adj_length = input('Enter the adjacent length: ');
opp_length = input('Enter the opposite length: ');
hyp_length = sqrt(opp_length^2 + adj_length^2);

%printing information
fprintf('The length of the hypotenuse is given by %f\n', hyp_length)
```

- What happens if the user enters a negative number for length?
  - The following code will still compute the length

MAKING PYTHAGORAS SMARTER

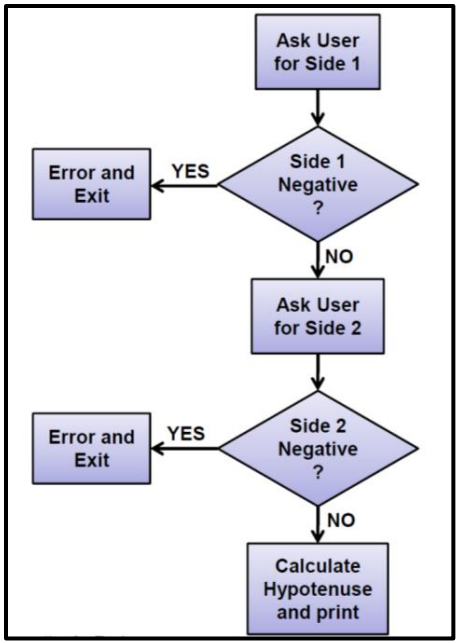
- Make the code exit if an input length is negative



RELATIONAL OPERATORS

- How do we get MATLAB to check if the user has input a negative value?
  - Use relational operators
- Checking if A is negative: A < 0

Relational Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to (double equals)
~=	Not equal to



RELATIONAL OPERATORS

- Relationship operators produce values of the logical data type
  - Logical values can be either true (1) or false (0)

```
>> A = 5;

>> A > 0
ans =
    1

>> A < 0
ans =
    0
```

```
>> A = [ 3 5 7 9 11 ];

>> A > 5
ans =
    0    0    1    1    1

>> A >= 5
ans =
    0    1    1    1    1

>> A ~= 5
ans =
    1    0    1    1    1
```

RELATIONAL OPERATORS WITH MATRICES

- Relational operators also work between matrices
  - Make sure matrices are the same size

```
>> A = [ 5 4 3 2 ]; B = [ 2 4 6 8 ];

>> A ~= B
ans =
    1    0    1    1

>> A < B
ans =
    0    0    1    1
```

```
>> A = [ 5 4 3 2 ]; B = [ 2 4 6 8 ];

>> A <= B
ans =
    0    1    1    1

>> C = (A > B)
C =
    1    0    0    0
```

= VS. ==

- The = operator assigns values

  A = B copies the values of B into A

- The == compares values

  A == B checks if A and B are equal

```
>> A = [ 4 2 0 ];
>> B = [ 2 2 2 ];

>> A == B
ans =
    0     1     0

>> A = B
A =
    2     2     2
```

LOGICAL FUNCTION

- The logical function converts numeric values to logicals
  - Numerically, non-zero is true and zero is false

```
>> A = [ 5 0 2 0 1 ];

>> logical(A)
ans =
    1     0     1     0     1

>> B = [ -5 0 10 0 0 -1]

>> logical(B)
ans =
    1     0     1     1     1
```

USING LOGICALS AS INDICES

- It is useful to use logicals as indices for matrix addressing

```
>> time = 10:15:85
time =
   10    25    40    55    70    85

>> speed = [10 -10 30 60 -20 60]
speed =
   10   -10    30    60   -20    60

>> index = (speed > 0)
index =
    1     0     1     1     0     1
```

```
>> pos_speed_times = time(index)
pos_speed_times =
   10    40    55    85

>> pos_speed = speed(index)
pos_speed =
   10    30    60    60
```

SUMMARY

- Code doesn't always get processed linearly
- Relational operators can be used to test statements
- Relational operators work between matrices

- Is it possible to check multiple conditions in a single line?