VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

User Defined Functions
Selection Statement

NEF1104 – Problem Solving for Engineers

Week 2: Session 5

Build-in functions

MATLAB has many built-in functions

Use help command to find them out and see how to use them

Example:

sqrt, mod, rem

log, log10, exp

sin, cos, tan

asin, acos, atan

User-defined functions

Functions that the programmer defines, and then uses, in either the Command Window or in a script

User-defined functions

A function in MATLAB that returns a single result consists of the following:

- Function header (the first line), comprised of:
  1) the reserved word function
  2) output argument
  3) name of function
  4) input arguments
- A comment that describes what the function does
- The body of the function
  1) all statements
  2) an output argument
- end at the end of the function

NOTE:
All the user defined functions need to be placed in the MATLAB default folder so other programs can read it.
Normally, the default folder is C:\Documents:\Matlab

functionname.m

```
function output_argument = function_name(input arguments)
% Comment describing the function Statements here;
% these must include putting a value in the output argument
end % of the function
```

```
% Call a function:
```

```
function_name(input arguments)
```

Example User-defined function

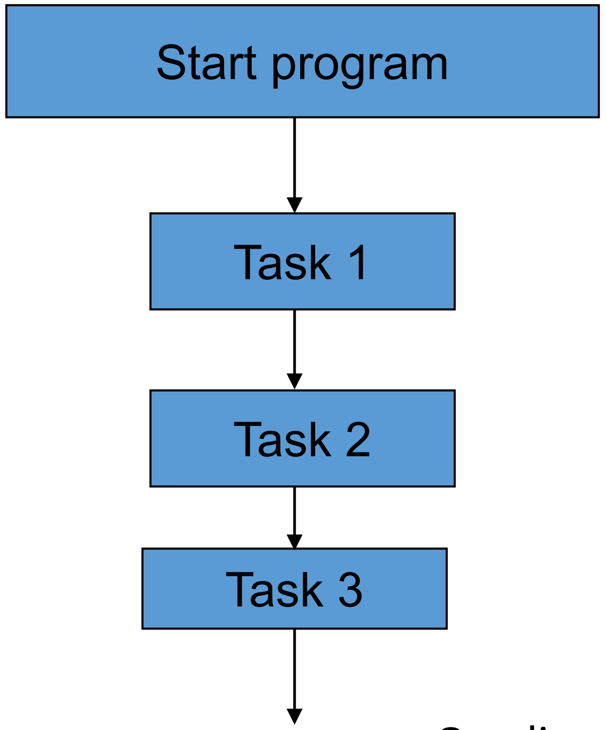
calcarea.m

```
function area = calcarea(rad)
% calcarea calculates the area of a circle
% Format of call: calcarea(radius)
% Returns the area
area = pi * rad * rad;
end
```

- The radius of a circle is passed to the function to the input argument rad.
- The function calculates the area of this circle and stores it in the output argument area.
- To call this function:

```
area = calcarea(5)
```

Flowcharts and Control



One line after another…

Selection Statement

- Decision making
- Relational and logical operators
- if statement
- if-else statement
- switch-case statement

Decision making

- Process is not always in one way, and Humans make a lot of decisions.

What if …

- How to tell a computer what to do and how to go about doing it
- Conditions and Actions

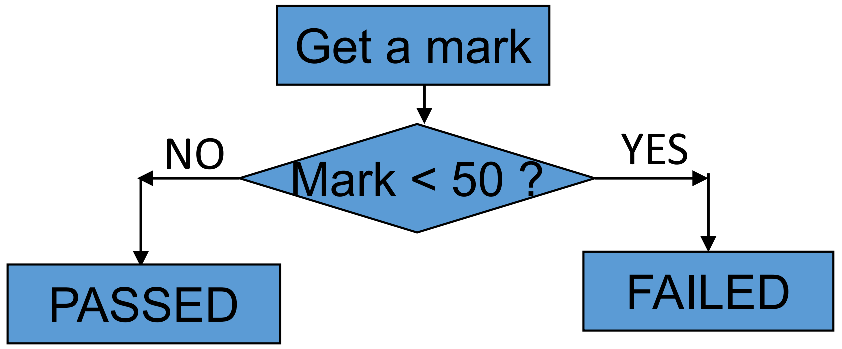
What is the “condition”?

- True (1) or false (0)
- Relational operators
- Logical operators
- Combinations of those

Get help from MATLAB Help window

Example

At the end of the semester, if you score up to 50 marks, you pass the subject.
However if you get less than 50, you get an F, which is bad.
When a student’s result is shown, we can easily see whether he/she passes or not.



If statement

VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

An if statement represents a decision

- The decision depends on a condition.
- Only use scalars in the condition.

Syntax:

```
if condition
        < my_commands>
End
```

If the condition is true, then < my_commands> are run.
If the condition is false, then < my_commands> are skipped.

Let's write some code that checks to see if a number is negative.

```
if number<0
   disp('Negative number');
end
```

If-else statement

VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

- It is common to want to do something else when an if statement condition is false.
- That is there are two options, you choose to take one of them based on a given condition.
- This is done by using if-else structure

NOTE:
Do Not write a condition for else

Syntax:

```
    if condition
                < do actions for the true condition >
    else
                < do actions for the false condition>
    end
```

Example:
Input a result, when the result is less than 50, display “FAIL”, otherwise display “PASS”

```
if result<50
   disp('Fail');
else
   disp('Pass');
end
```

If – elseif – else chain

For more complicated decision making process, you can use elseif statement. Which will form a if – elseif – else chain.

Syntax:

```
if condition1
        <action1>
elseif condition2
        <action2>
else
        <action3>
end
```

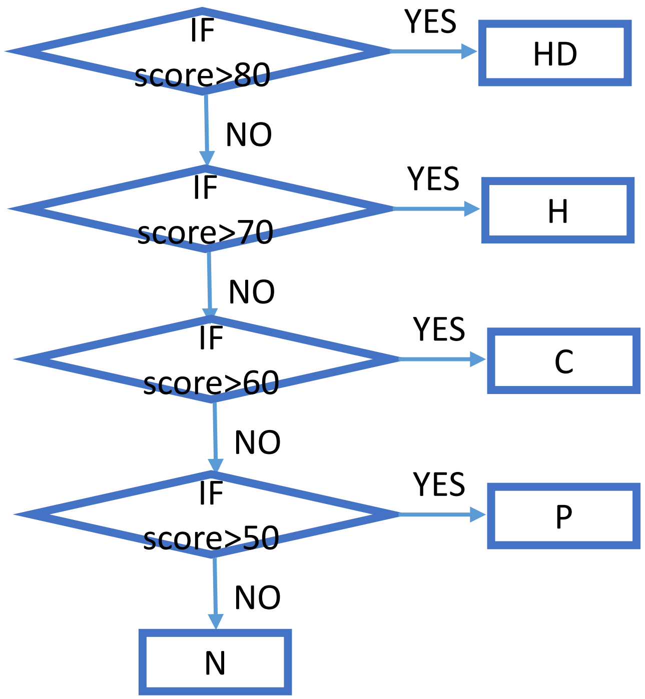
- If the condition1 is true, do action1 then go to the end
- If the condition1 is false, check condition2. If condition2 is true, do action2 then go to the end
- If the condition2 is false, do action3 then go to the end.

It is possible to have many “elseif”s

If – elseif – else chain

Example:
Again consider the semester's final result.
When it is over 80, HD is graded.
If it is over 70, H is graded.
If it is over 60, C is graded.
If it is over 50, P is graded.
If it is less than 50, N is graded.

```
if score>80
    disp('HD')
elseif score>=70
    disp('H')
elseif score>=60
    disp('C')
elseif score>=50
    disp('P')
else
    disp('N')
end
```



Switch statement

VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

For multi-option decision making, there is another way to do it through switch statement.

Syntax:

```
switch switch_expression
    case case_expression 1
    case case_expression 2
    case case_expression 3
    otherwise
        statements
end
```

Example:
Display different text conditionally depending on a value entered

```
n = input('Enter a number: ');
switch n
    case -1
        disp('negative one')
    case 0
        disp('zero')
    case 1
        disp('positive one')
    otherwise
        disp('other value')
end
```

Combo on conditions

From time to time you face the case that there are a few conditions need to be met for an action to be taken.
Use logical operators to combine the conditions together.

Example:
Input an integer number, and check whether it is bigger than 3 and less than 7.

```
if num>3 & num<7
    disp('yes')
else
    disp('no')
end
```