

**CE-105L Computer Programming for Civil Engineers – Matlab  
File Operations, Functions, Loops and Branching  
Engr. Faisal ur Rehman**

**Notes Available on:  
enggprog.com ( Engineering Programs)**

**Working with files:**

First create an excel file named data.xls. To read excel file and save to a matrix named A

```
>>A=xlsread('data.xls')
```

To save a file in out format:

```
>>save 'mydata.out', A , -ASCII
```

To save a file in delimited format:

```
>>dlmwrite('mudata1.out',A,';')
```

**Functions:**

1. Functions are defined in m-file
2. Name of function is same as that of m-file
3. Functions are executed in command window by calling its name
4. Path of script file and function needs to be the same

**Function Syntax:**

```
function return variable = function name ( input variable1, input variable 2, ...)
```

```
function statements
```

```
end
```

**Function Example:**

**Example 1: Force Function f**

```
function y = f(m,a)
```

```
    y = m*a;
```

```
end
```

Save above script by name of f.m

In command window, run following commands:

```
>> f(3,2)
```

```
>>p=f(4,5)
```

**Example 2: Circle Function for Area and Volume**

```
function [A,V]=circle(r)
```

```
    A = pi*r.^2;
```

```
    V = 4/3*pi*r.^3;
```

```
end
```

Save the script file by name of circle.m

In this example we have 2 output Array/Matrix Variables. The square bracket tells that the output variable is array. In finding power of radius r, dot(.) symbol is used indicating element wise multiplication.

In command window, run following commands:

```
>>circle(5)
```

Above command will return only area

```
>>radius=[1:0.1:10];
```

```
>>[Ar,Vl] = circle(radius)
```

Above command will return Array of Area and Volume

### **Example 3: Function as Sub Routine:**

Function can be defined as sub routine when output variable is omitted.

```
function z(x)
    y1 = 3*x^2 + 2
    y2 = 5*x^2 + 3*x + 8
end
```

Save above script with name of z.m

In command window, run:

```
>> x(4)
```

### **Loops:**

While and For loops are used in Matlab.

### **While Loop Syntax:**

```
while condition
    while statements
end
```

### **While Loop Example:**

```
r=0;
i=0;
while i<=20
    r = r + i;
    i = i +1;
end
disp('sum of numbers from 0 to 20 is: ')
disp(r)
```

Save above script with name of whileEx.m

In command window, run:

```
>>whileEx
```

### **For Loop Syntax:**

```
for counter = m:s:n
    for statements
end
```

where m is starting number, s is the step size and n is the end number.

### **For Loop Example:**

```
r = 0;
for i = 0:20
    r = r+i;
end
disp('Sum of numbers from 0 to 20 is:')
disp(r)
Save above script with name of forEx.m
In command window, run:
>>forEx
```

### **Branching**

If  
Switch

### **If Syntax:**

```
if condition 1
    if statements
elseif condition 2
    else if statements
else
    else statements
end
```

### **If Example:**

```
function evenodd(x)
    if mod(x,2) ==0
        disp('Even Number')
    else
        disp('Odd Number')
    end
end
```

Save above script with name of evenodd.m

**Switch Syntax:**

switch variable (note: variable can be a string or numeric)

```
case value1
    statement 1
case value2
    statement 2
case value3
    statement 3
otherwise
    otherwise statement
end
```

**Switch Example 1:**

```
function getdirec(angle)
    switch angle
        case 45
            disp('North-East')
        case 135
            disp('South-East')
        case 225
            disp('South-West')
        case 315
            disp('North-West')
        otherwise
            disp('Unknown Direction')
    end
end
```

Save above file with name of getdirec.m

You can specify multiple values to check by specifying comma and medium bracket.

**Switch Example 2:**

```
function getdirec1(angle)
    switch angle
        case {0,360}
            disp('North')
        case {90,-270}
            disp('East')
        case {180,-180}
            disp('South')
        case {270,-90}
            disp('West')
        otherwise
            disp('Unknown Direction')
    end
end
```

Save above script by the name of getdirec1.m

## Relational Operators

1. < less than
2. > greater than
3. <= less or equal to
4. >= greater or equal to
5. == comparison
6. ~= not equal to

## Logical Operators

1. & and
2. | or
3. ~ not
4. && Short circuited and
5. || Short circuited or