Section 3.6 – User written functions

Functions – are also m-files but can be used by your program with input & output arguments. It also has the .m extension. The variables used inside the function are "local" unless you make them global. Rules:

1. **Must begin with the word function then the output argument, equals sign, name of function, input arguments within parenthesis. For example: function a=area_c(r)**
2. **Always put comments explaining it next. Example: % area_c computes area of a circle(r is the radius) When user types help area_c, these comments are displayed**

More rules… (see p.79-80)

Chapter 3.6 – User written functions (cont'd)

Create an m-file function:

```
function a=area_c(r)
%
% area_c computes the area of a circle
% r is the radius (can be a scalar or vector)
a=pi*r.^2;
disp('calculation done!');  % this display not needed
```

Save the function in a file named: area_c.m

```
%try using it on the command line now:
>> clear
>> area_c(10)
```

Create another function called vol_box to calculate the volume of a box with dimensions x, y, z.

User written functions

Example:

Must begin with the word function then list output arguments [in brackets if more than 1], equals sign, name of function, input arguments within parenthesis

```
function [h, l,  Y]=high_low(X)
% high_low finds the maximum (h) and the minimum (l)
% also takes input vector X and sorts into output Y
h=max(X);
l=min(X);
Y=sort(X);
```

Don't forget comments to use as help.

Save the function in a file named: high_low.m

```
% try it
>> A=[4, 5, 0, 2, 15, 3, 10]
>> [top, bot, Z] = high_low(A)
```

The variables used inside the function are "local" unless you make them global.

Rules p.79-80

User written functions – more examples:

1 input & 1 output

```
function r =convert_to_rad (x)
% converts degrees to radians
r=x*(pi/180);
```

Save the function in a file named: convert_to_rad.m

```
% try it just to do a calculation:
>> convert_to_rad(45)
% or try it with a variable:
>> y_angle=input('Enter the angle in degrees: ')
>> y_radians=convert_to_rad(y_angle)
```

The variables used inside the function are "local".

User written functions (cont'd)

```
function area =calc_area (x,y)
% calculates area of a rectangle of dimensions x and y
area=x*y;
```

Save the function in a file named same as this but .m

```
% Use it in the command window … examples:
>> calc_area(10, 15)
% or try it with variables in an m-file
length=input('Enter the length in feet: ');
width=input('Enter the width in feet: ');
Area_rectangle=calc_area(length, width)
```

*** You can have 0, 1, 2, or more inputs … all listed AFTER the function name (in parenthesis).**

*** You can have 0, 1, 2, or more outputs … if 2 or more outputs, enclose [in brackets] … output(s) always appear BEFORE the equal sign!**