Bangladesh University of Professionals (BUP)

Department of Information and Communication Technology (ICT)

Faculty of Science and Technology (FST)

Program: B.Sc in Information and Communication Engineering (ICE)

Course Name : Numerical Methods Lab

Course Code : ICE-2210

Student's Name:	Arpa Tasnim
Class Roll:	2054901079
Registration No:	20010729
Section:	B
Session:	2019-2020

LAB REPORT- 03

Introduction: Matlab provides various types of loops to handle looping requirements including: while loops, for loops, and nested loops. If you are trying to declare or write your own loops, you need to make sure that the loops are written as scripts and not directly in the Command Window.

While Loop:

The while loop repeatedly executes statements while a specified condition is true. The syntax of a while loop in MATLAB is as following:

```
while <expressions>
    <statements>
end
```

For loop:

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times. The syntax of a for loop in MATLAB is as following:

```
for index = values

    <program statements>

            …

end
```

Break statement:

The break statement terminates execution of for or while loops. Statements in the loop that appear after the break statement are not executed. In nested loops, break exits only from the loop in which it occurs. Control passes to the statement following the end of that loop.

Functions:

A function accepts one or more MATLAB variables as inputs, operates on them in some way, and then returns one or more MATLAB variables as outputs and may also generate plots or just generates plots only. This functions could be Pre-defined/inbuilt or user defined to solve a given problem.

A user-defined function is a MATLAB program that is created by the user, saved as a function file, and then can be used like a built-in function. The function can be a simple, single mathematical expression or a complicated and involved series of calculations. In many cases it is actually a subprogram within a computer program. The main feature of a function file is that it has an input and an output. This means that the calculations in the function file are carried out using the input data, and the results of the calculations are transferred out of the function file by the output. The input and the output can be one or several variables, and each can be a scalar, vector, or an array of any size.

Solution of Example 1

Editor - E:\MATLAB\example1.m

example1.m | Example2.m | Example3.m | Example

```
1    %Arpa Tasnim, 2054901079
2 -  clc;
3 -  clear all;
4 -  x=-3
5 -  if x>0
6 -   a=10
7 -   elseif x<0
8 -   a=11
9 -   elseif x== 0
10 -  a=12
11 -  else
12 -  a=14
13 -  end
```

Command Window

```
x =

    -3


a =

    11

fx >>
```

Solution of Example 2

Editor - E:\MATLAB\Example2.m

example1.m | Example2.m | Example3.m | E

```
1    %Arpa Tasnim, 2054901079
2 -  clc;
3 -  clear all;
4 -  x=-10;
5 -  while x<-8
6 -  x=x+1
7 -  end
```

Command Window

```
x =

    -9


x =

    -8

fx >>
```

Solution of Example 3

Editor - E:\MATLAB\Example3.m

example1.m | Example2.m | Example3.m | Example4.m | cal_pow.m

```
1     %Arpa Tasnim, 2054901079
2 -   clc
3 -   clear all
4 -   x=0
5 -   for i=1:3
6 -    x=x+1
7 -   end
```

Command Window

```
x =

     0


x =

     1


x =

     2


x =

     3
```

Solution of Example 4

Editor - E:\MATLAB\Example4.m

example1.m | Example2.m | Example3.m | Example4.m | cal_pow.m

```
1    %Arpa Tasnim, 2054901079
2 -  clear all
3 -  x=-10
4 -  while x<0
5 -   x=x+2
6 -   if x == -8
7 -   break
8 -   end
9 -  end
```

Command Window

```
>> Example4

x =

   -10


x =

    -8

fx >>
```

Solution of Example 5

Function file:

Editor - E:\MATLAB\cal_pow.m

cal_pow.m | Example5.m | Ex_var.m | Exer

```
1    %Arpa Tasnim, 2054901079
2
3    function y=cal_pow(x)
4 -  y=1+x^2
5 -  end
```

Main file:

Editor - E:\MATLAB\Example5.m

cal_pow.m | Example5.m | Ex_var.m | Exercise1a.m | Exercise1b.m

```
1    %Arpa Tasnim, 2054901079clear all
2 -  x=0:1:3
3
4 -  t=length(x)
5 -  for i=1:t
6 -  val(i)=cal_pow(x(i))
7 -  end
8
```

Command Window

```
>> Example5

x =

     0     1     2     3


t =

     4


y =

     1


val =

     1     2     5    10


y =

     2


val =

     1     2     5    10
```

```
y =

     5


val =

     1     2     5    10


y =

    10


val =

     1     2     5    10

fx >>
```

Solution of Exercise1

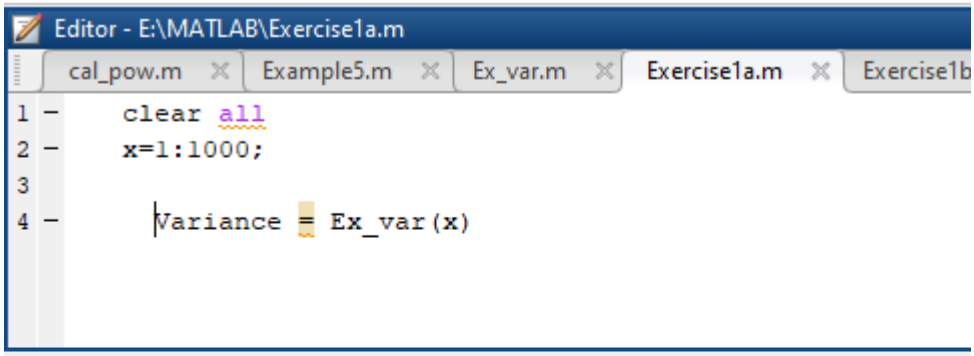
Function file:

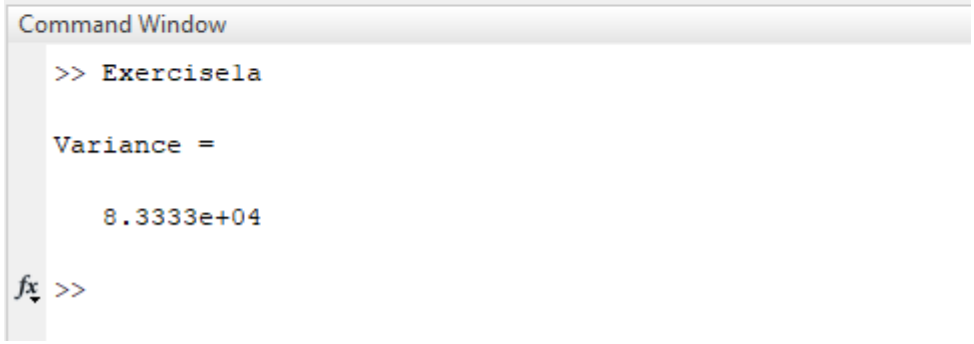
Editor - E:\MATLAB\Ex_var.m*

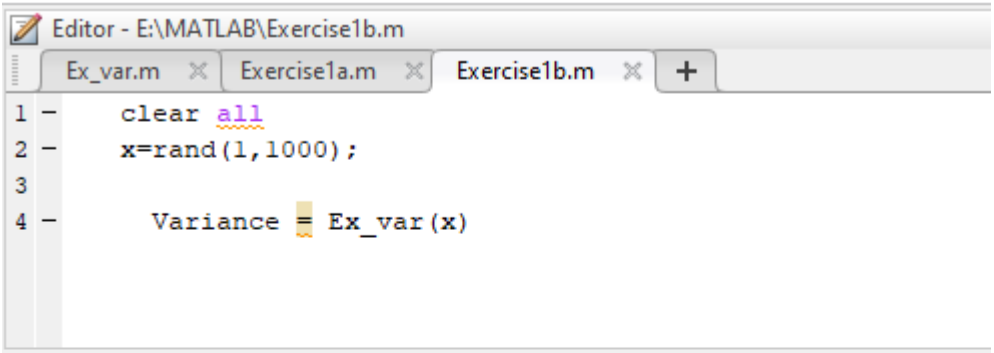
cal_pow.m | Example5.m | Ex_var.m* | Exercise1a.m | Exercise1b.m

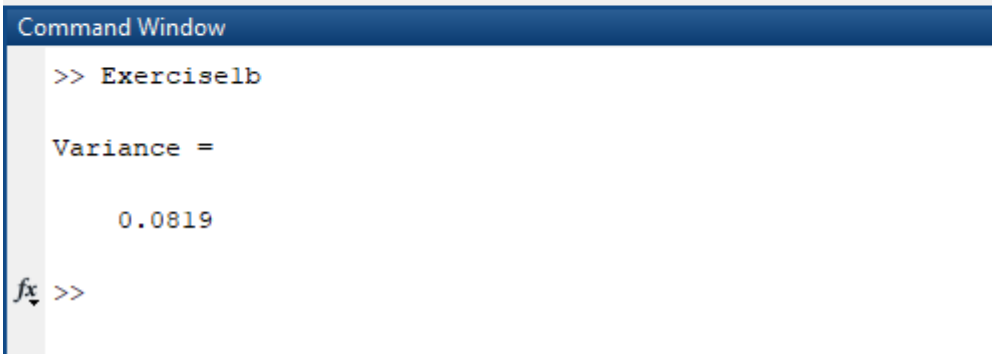
```
1   %Arpa Tasnim, 2054901079clear all
2   %subfunctions are called by this primary function
3   function Var = Ex_var(v)
4   n=length(v);
5   av=AVG(v,n);
6   Var=Variance(v,av,n);
7
8   %this function calculates xAve(average)
9   function av=AVG(x,num)
10  av=sum(x)/num;
11
12  %this function calculates σ
13  function Var=Variance(x,xAve,num)
14  x1=x-xAve;
15  x2=x1.^2;
16  Var= (sum(x2)/(num));
17
```

Main file:


Editor - E:\MATLAB\Exercise1a.m
cal_pow.m
Example5.m
Ex_var.m
Exercise1a.m
Exercise1b
1 clear all
2 x=1:1000;
3
4 Variance = Ex_var(x)


Command Window
>> Exercise1a
Variance =
8.3333e+04
>>


Editor - E:\MATLAB\Exercise1b.m
Ex_var.m
Exercise1a.m
Exercise1b.m
1 clear all
2 x=rand(1,1000);
3
4 Variance = Ex_var(x)


Command Window
>> Exercise1b
Variance =
0.0819
>>

Discussion:

In Exercise-1, a user-defined function, named Ex_var, is written for solving the problem. To demonstrate the use of subfunctions, the function file includes Ex_var as a primary function, and two subfunctions called AVG and Variance. The function AVG calculates xAve(Average value) , and the function Variance calculates σ(which is given in the lab manual). The subfunctions are called by the primary function.

In **(a),** For x, using all the integers from 1 to 1000**.** Declare x=1:1000, and then calling the

user defined function Ex_var and passing the value of x through this function.

In **(b),** For x, using built in function rand. The `rand` function generates arrays of random numbers whose elements are uniformly distributed in the interval (`0`,`1`). `Y=rand(n)` returns an `n`-by-`n` matrix of random entries.

So declare x=rand(1,1000); and by calling the Ex_var(user defined function), pass the value of x.

Reference:

[1]https://www.uwyo.edu/ceas/resources/current-students/classes/esig%20help/Windows%20Help%20Files/MatLab/Loop%20Types.pdf