Tutorial 02

L3 – Scripts and Functions

L4 – Classes and File operations

Exercise

- Create a MATLAB script file to solve the following statement –

- Suppose x takes on the values x = 1, 1.2, 1.4, . . . , **LIMIT**. Use MATLAB to compute the array y that results from the function y = 7 sin(4x).
- Use MATLAB to determine how many elements are in the **array y** and the value of the third element in the **array y**.

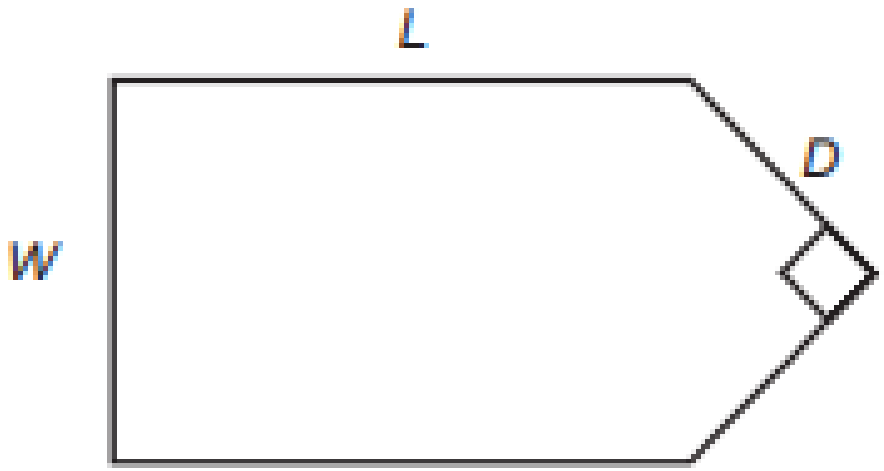
- **LIMIT** is specified by the user.

ANSWER

- clear all;clc;
- Limit = input('Please enter the limit');
- X=1:0.2:Limit;
- Y=7*sin(4*X);
- n = numel(Y); % Number of array elements
- strdisp_1 = strcat('Number of array elements =',num2str(n));
- disp(strdisp_1); % display the number of elements.
- strdisp_2 = strcat('The value of third element is',num2str(Y(3)));
- disp(strdisp_2); % display the value of third element in the array.

Exercise

- A fence around a field is shaped as shown in Figure. It consists of a rectangle of length L and width W and a right triangle that is symmetric about the central horizontal axis of the rectangle. Suppose the width W is known (in meters) and the enclosed area A is known (in square meters).
- Write a MATLAB script file in terms of the given variables W and A to determine the length L required so that the enclosed area is A.
- Also determine the total length of fence required. Test your script for the values W = 6 m and A = 80 m2 .



ANSWER

- clear all; clc;
- Width = input('Please enter width (m) =');
- Area = input('Please enter area (m2) =');
- Len_gth = (Area - ((Width^2)/4))/Width;
- Peri_meter = 2*Len_gth+Width+2*(Width/sqrt(2));
- fprintf('Perimeter of Fence: %f', Peri_meter);

Exercise

- Create a function file **FACT** to calculate factorial of a given number **X**.
- Use the function FACT to create another function file PNC to calculates and returns permutation and combination of (n, r).
- $$P(n,r) = \frac{n!}{(n-r)!} \qquad {}_nC_r = \frac{n!}{r!(n-r)!}$$
- Do not use any **LOOPS** in the factorial function.

ANSWER

- function fact_orial = **Fact_orl**(n)
- **fact_orial** = **prod**(1:n); %Product of array elements MATLAB function
- end
- ---------- Script File ----------
- % accepting input from user
- n = input('Enter value of n: ');
- r = input('Enter value of r: ');
- nCr = Fact_orl(n)/(Fact_orl(r)*Fact_orl(n-r));
- fprintf('nCr: %d', nCr);
- nPr = Fact_orl(n)/Fact_orl(n-r);
- fprintf('nPr: %d', nPr);

Exercise (refer file – T02-Live_script.mlx

- Use live script editor to run the following TASKs within a single live script. See execution of individual sections. **Add comments**.

- TASK 1 : Create a row vector named X that contains the values 1, 2, and 3, in that order.

- TASK 2 : Create a row vector named y with integer values from 1 to 10 using the : operator.

- TASK 3 : Create two variables A and B using numeric sliders.
- ADD them and MULTIPLY them

Exercise

- Create a MATLAB class file that accepts student marks in different subjects (more than one subject) and returns GPA.
- The GRADES and CREDITS must be the properties of the OBJECT.
- The function should calculate GPA.
- Show a display message saying that ‘The GPA for Student X is 2.67’.
- Example : GRADES(1) = [9 10 8 7 9]; - Grades of Student 1 in 5 subjects.
- CREDITS = [3 3 4 2 1];

$$GPA = \frac{\sum GRADES \ * CREDITS}{\sum CREDITS}$$

ANSWER

GPAfind.m

```
clear all;clc;

Grades =[ 9 10 8 7 9 ];
Credits = [ 3 3 4 2 1];

a = CalcGPA;
a.Grades = Grades;
a.Credits = Credits;
t=a.GPAcalc();
```

CalcGPA

```
classdef CalcGPA
    properties
        Grades
        Credits
    end
    methods
        function GPA = GPAcalc(obj)
            GPA = sum(obj.Grades.*obj.Credits)/sum(obj.Credits);
        end
    end
end
```

Exercise

- Write a MATLAB function to copy all **.m** files in the current folder to a new subfolder named '**BACKUP**'.
- Zip the subfolder '**BACKUP**' and move the ZIP file one folder UP/ABOVE the current folder.

- mkdir('BACKUP');
- copyfile('*.m','BACKUP');
- **zip('../BACKUP.zip', '*.m', 'BACKUP');**
- OR
- **zip('../BACKUP.zip','BACKUP');**