Week 12 Activities

Contents

Time required: 120 minutes

1. Create a MATLAB script named **Wk12Lastname.m**
2. Save all programs in this script.
3. Include your name and date at the top of the script file as comments.
4. Put a Section Break between each program.

Reading

Matlab A Practical Introduction to Programming and Problem Solving (Stormy Attaway)

Sections 6.1, 6.2

Tutorial 1: GUI Menu Sin, Cos, or Tan Function If

The syntax of MATLAB's menu GUI. You can have any number of options.

choice = menu('instructions', 'option1', 'option2', 'option3')

choice - the returned value is stored in this variable

instructions - text/instructions for the user

option1 - text that appears on top button. Returns a value of 1

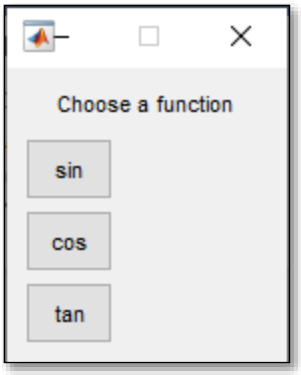
option2 - text that appears on second button. Returns a value of 2

option3 - text that appears on the third button. Returns a value of 3


```
% Prints either sin, cos, or tan of x
% uses the menu function to choose
x = input('Enter a value for x: ');
choice = menu('Choose a function', 'sin','cos','tan');
if choice == 1
    fprintf('sin(%.1f) is %.1f\n', x, sin(x))
elseif choice == 2
    fprintf('cos(%.1f) is %.1f\n', x, cos(x))
elseif choice == 3
    fprintf('tan(%.1f) is %.1f\n', x, tan(x))
else
    disp('Error!')
end
```

Example run:

```
Enter a value for x: 23.1
```



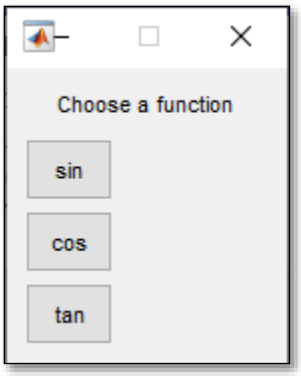

```
tan(23.1) is 2.0
```

The same menu program using switch.

```
%%
% Prints either sin, cos, or tan of x
% uses the menu function to choose
x = input('Enter a value for x: ');
choice = menu('Choose a function', 'sin','cos','tan');
    switch choice
        case 1
        fprintf('sin(%.1f) is %.1f\n', x, sin(x))
    case 2
        fprintf('cos(%.1f) is %.1f\n', x, cos(x))
    case 3
        fprintf('tan(%.1f) is %.1f\n', x, tan(x))
    otherwise
        disp('Error!')
end
```

Example run:

```
Enter a value for x: 23.1
```



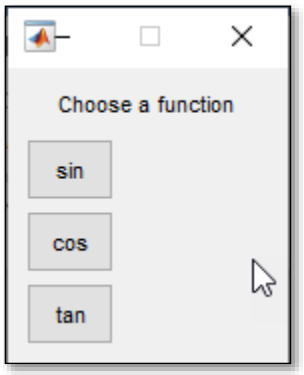

```
tan(23.1) is 2.0
```

Tutorial 2: GUI Menu Sin, Cos, or Tan Function Switch Case

```
%%
% Prints either sin, cos, or tan of x
% uses the menu function to choose
x = input('Enter a value for x: ');
choice = menu('Choose a function', 'sin','cos','tan');
    switch choice
        case 1
        fprintf('sin(%.1f) is %.1f\n', x, sin(x))
    case 2
        fprintf('cos(%.1f) is %.1f\n', x, cos(x))
    case 3
        fprintf('tan(%.1f) is %.1f\n', x, tan(x))
    otherwise
        disp('Error!')
end
```

Example run:

```
Enter a value for x: 22.6
```




```
sin(22.6) is -0.6
```

Tutorial 3: Function Types Examples

Functions can be in separate files, or in the same file at the end of the file. An easy way to create these examples is to create each as a separate file which includes the function.

NOTE: Functions can work with a single variable or a vector.

Function with one output and one output:

```
%% Function with one input and one output
% Filename: Wk12FunctionOneOutput.m

% Input vector or single variable
values = [12, 4, 8.9, 6, 3];

% Call stat function, return vector or single variable
mean = stat(values);
disp(mean);

% Function returns mean of vector values
function m = stat(x)
    n = length(x);
    m = sum(x) / n;
end
```

Example run:

```
6.7800
```

Function with multiple inputs and outputs:

```
%% Function with multiple inputs and outputs
% Filename: Wk12FunctionMultipleOutputs.m

% Input
x = 3;
y = 2;

% The function calculates both nCr and nPr of inputs n and r.
[p, c] = perm(x, y);
disp(p);
disp(c);

% Function return p = nPr and c = nCr
function [p, c] = perm(n, r)
    p = factorial(n)/factorial(n - r);
    c = p*factorial(r);
end
```

Example run:

```
p =

     6


c =

    12
     6
    12
     6
    12
```

Function calling a function:

```
%% Wk12FunctionMultiple.m
% A function calling a function
% stat2() function calculates the standard deviation of the input vector.
% stat1() calculates the mean of the input vector.

values = [12, 4, 8.9, 6, 3];

value = stat2(values);
disp(value);


% Function returns standard deviation of vector x
function sd = stat2(x)
    m = stat1(x);
    n = length(x);
    sd = sqrt(sum((x - m).^2 / n));
end

% Function returns mean of vector x
function m = stat1(x)
    n = length(x);
    m = sum(x) / n;
end
```

Example run:

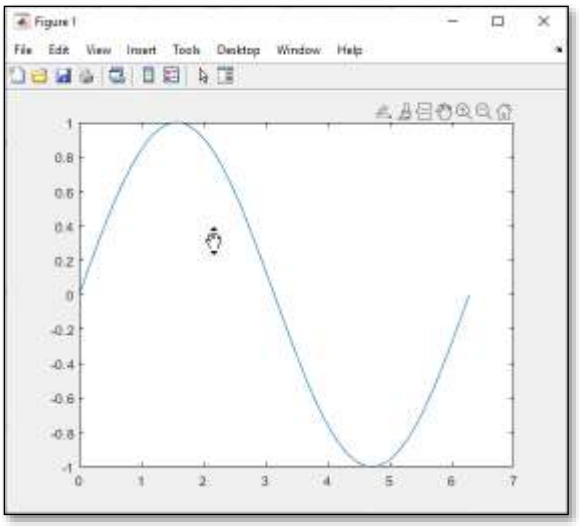
```
3.2975
```

Function with no input or output:

```
%% Wk12FunctionNoImput.m
% Function with no input or output
sin_plot()

% Plotting sin(x) function
function sin_plot()
    x = linspace(0, 2*pi, 100);
    y = sin(x);
    plot(x, y);
end
```

Example run:



Assignment 1: Right Triangle Function

For a right triangle with sides a, b, and c, where c is the hypotenuse and θ is the angle between sides a and c, the lengths of sides a and b are given by:

a =c * cos(θ)

b =c * sin(θ)

1. Write code that calls a function to prompt the user and read in values for the hypotenuse and the angle (in radians) and returns those values

2. Call a function to calculate and return the lengths of sides a and b.
3. Print all values in a sentence format. For simplicity, ignore units.

Here is an example of running the script; the output format should be as shown here:

```
Enter the hypotenuse: 2.36
Enter the angle: 45
For a right triangle with hypotenuse: 2.4
and an angle: 45.00 between side and the hypotenuse,
side A: 1.24  side B: is 2.01
```

Assignment 2: m/s to ft/s

In the SI system, speed is measured in meters per second (m/s). A foot per second (ft/s) is equivalent to 0.3048 m/s.

1. Prompt the user to enter speed in meters per second.
2. Create a function that takes meters per second as an input and returns the equivalent speed in feet per second.

Example run:

```
Convert m/s to ft/s
Enter the speed in m/s: 12.305
12.305 m/s is 40.371 ft/s
```

Assignment Submission

1. Submit properly named and commented script files.
2. Attach a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.