```
% J Hundley
% assign07a.m
%{
   Modify assign06b.m to use user defined functions.
   Assume that the distance to the building along the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees
   plus/minus 3 degrees.
   --Read degrees using load() from data file into a vector degrees.
a) compute the height of the building for given the distance and each acceptable angle
   display a label and the table of angles and heights from the results of a).
b) get and print average angle in the vector
%}
clc, clear all

%*****CONSTANT*****
DISTANCE    = 120;   % meters
RADIAN_FILE = 'dataAssign07.txt';
%getHeight   = @(radians) DISTANCE .* tan( radians );

%*****INPUT*****
if ~exist( RADIAN_FILE, 'file' )
    fprintf( 'File not found\n' )
else
    % read data file
    theta = load( RADIAN_FILE );

    %*****COMPUTATION*****
    % compute the height of the building for given distance and each angle
    %heightAll = DISTANCE .* tand( theta );
    heightAll = DISTANCE .* tan( theta );

    % get average angle of angles in vector
    meanTheta  = mean( theta );
    % compute the height of the building using the average angle
    heightMean = DISTANCE .* tan( meanTheta );

    %*****OUTPUT*****
    % print title, headers and table of angles and heights
    fprintf( ' Angle     Height\n' )
    fprintf( 'Radians    Meters\n' )
    fprintf( ' %6.4f     %5.2f \n', [theta'; heightAll'] )

    % print a average angle and height using average angle
    fprintf('\nThe height %5.2f meters for the average angle %6.4f radians.\n',...
        heightMean, meanTheta )
    % print the number of degrees read
    fprintf( '%.0f radians were read from the data file.\n', length( theta ) )

end % end good file open
```

```matlab
% J Hundley
% assign07b.m
%{
   Modify assign07a.m to use anonymous, user-defined, and subfunctions functions.
   Assume that the distance to the building along the ground is 120 meters
   and the angle measured alone the line of sight is 30 degrees
   plus/minus 3 degrees.
   --Read degrees using load() from data file into a vector degrees.
a) compute the height of the building for given the distance and each acceptable angle
   display a label and the table of angles and heights from the results of a).
b) get and print average angle in the vector
c) draw a graph of the measurements
%}
clc, clear all

%*****CONSTANT*****
DISTANCE    = 120;   % meters
RADIAN_FILE = 'dataAssign07.txt';
getHeight   = @(radians) DISTANCE .* tan( radians );

%*****INPUT*****
if ~exist( RADIAN_FILE, 'file' )
    fprintf( 'File not found\n' )
else
    % read data file
    theta = load( RADIAN_FILE );

    %*****COMPUTATION*****
    % compute the height of the building for given distance and each angle
    %heightAll = DISTANCE .* tand( theta );
    heightAll = getHeight( theta );

    % get average angle of angles in vector
    meanTheta  = mean( theta );
    % compute the height of the building using the average angle
    heightMean = getHeight( meanTheta );

    %*****OUTPUT*****
    % print angle and height report
    printReport( theta, heightAll, heightMean, meanTheta )

end % end good file open
```

PRIMARY FUNCTION
FUNCTION NAME SAME AS FILE NAME

```
% J Hundley
% printReport.m used by assign07b.m

function [] = printReport( theta, heightAll, heightMean, meanTheta )
% print angle and height report

% print title, headers and table of angles and heights
fprintf( ' Angle     Height\n' )
fprintf( 'Radians    Meters\n' )
fprintf( ' %6.4f     %5.2f \n', [theta'; heightAll'] )

% print a average angle and height using average angle
fprintf('\nThe height %5.2f meters for the average angle %6.4f radians.\n',...
    heightMean, meanTheta )
% print the number of degrees read
fprintf( '%.0f radians were read from the data file.\n', length( theta ) )
% draw a plot of the measurements
plotData( theta, heightAll )
end
```

SUBFUNCTION LOCATED IN SAME FILE
AS PRIMARY FUNCTION THAT USES IT

```
function [] = plotData( theta, heightAll )
% plot angle and height measurements

measureNum = 1:length( theta );
[hAx,hLine1,hLine2] = plotyy( measureNum, theta, measureNum, heightAll );
title( 'Angles .vs Heights' )
xlabel( 'Measurement number' )

hLine1.LineStyle = '--';
hLine2.LineStyle = ':';
ylabel(hAx(1),'Angles (radians)') % left y-axis
ylabel(hAx(2),'Heights (meters)') % right y-axis
end
```

Read all instructions
before beginning your work.

COMP1200-MatLab - assign 07
Due 4:45 pm – Friday – October 25, 2019
Submit `assign07a.m` `assign07b.m` and
`printReport.m` **via Canvas**

NOTE:
Your submitted file(s) MUST be
spelled and cased as instructed.
[-5 points for not doing so.]

Before you start writing your program:

Read the complete instructions.

***Problem*:**

Trigonometry can be used to find the height of a building. Suppose you measure the angle between the line of sight and the horizontal line connecting the measuring point and the building. You can calculate the height of the building with the following formulas:

```
          tan(theta)=h/d          h=d*tan(theta)
```

Assume that the distance to the building along the ground is 120 meters.

Exam02
Wednesday,
Oct. 30

***Program*: assign07a.m**

Edit your assign06b.m to use a file of **radians**. **All requirements for assign06 still apply.**
I suggest you begin by saving you assign06b.m as assign07a.m. and make the needed changes.
.

***Problem Constants*:**

distance `120 % meters`
file name `'dataAssign07.txt'`

***Problem Inputs*:**

angles (theta) in radians from data file

Problem Outputs:

heights of the building for given the distance and angles
average angle
height of the building using the **average angle**

Other variables:

as needed

Equations:

See above.

Output:

```
 Angle     Height
Radians    Meters
 0.5566     74.67
 0.5661     76.26
 0.4845     63.16
 0.5669     76.39
 0.5375     71.52
 0.4814     62.69
 0.5004     65.62

The height 69.93 meters for the average angle 0.5276 radians.
7 radians were read from the data file.
```

Do not use commands and statements beyond what has been taught on class.

New commands:
Note: angles are radians not degrees
Set decimals as shown
with columns right aligned.

Continue to use:
all from assign06
Use descriptive variables.

***Program*: assign07b.m**

Edit your assign07a.m to include the following anonymous function and user-defined function definition. You can use parts of assign07a.m but remove statements and comments that do not apply to the assign07b.m requirements.

assign07b.m will complete all requirements except printing the output report.
A user-defined function, `printReport()`, will print the output report.

In `assign07b.m`:

Complete all requirements except printing the output report.
Use the following anonymous function to compute the heights for the vector of angles and average angle. An anonymous function should be place near the top of the script with the constants.

```
        getHeight = @(radians) DISTANCE .* tan( radians);
```

Use `printReport()` to print the output report.

Do not use commands and statements beyond what has been taught on class.

New commands:
anonymous function
user-defined function
subfunction
NO global variables

Continue to use:
`fprintf()`
Do not use `disp()`

In **`printReport()`** include all statements and comments needed to print all the output and draw a graph.
Use the given subfunction to plot the measurements.

Function should be named as given and save in a file **`printReport.m`**.
Variable names may be different, but the order and quantity should be as given.

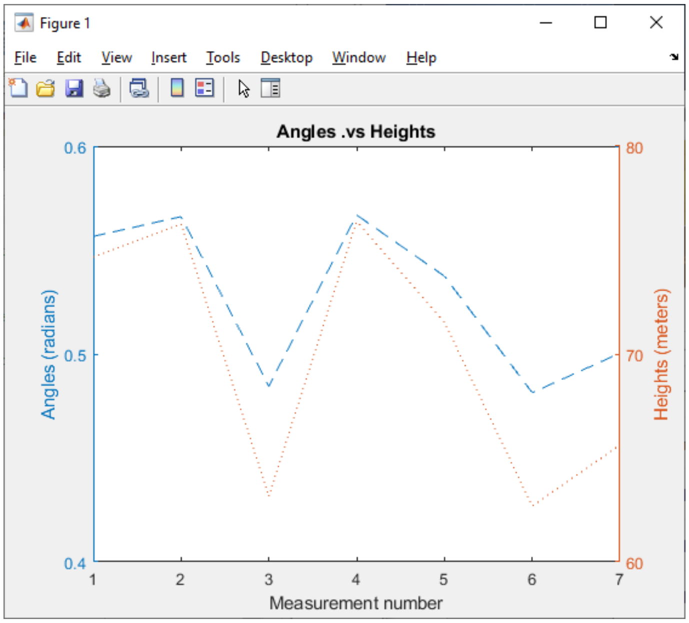
```
function [] = printReport( theta, heightAll, heightMean, meanTheta )
% print angle and height report
```

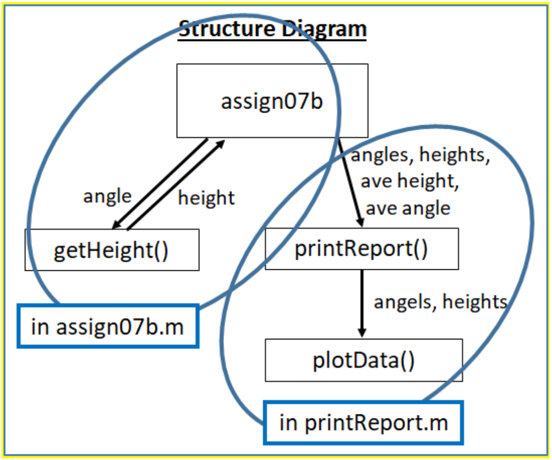
Subfunction, **`plotData()`**, is located in the primary function, **`printReport.m`** file, and is used by the primary function to draw a graph.

```
function [] = plotData( theta, heightAll )
% plot angle and height measurements
measureNum = 1:length( theta );
[hAx,hLine1,hLine2] = plotyy( measureNum, theta, measureNum, heightAll );
title( 'Angles .vs Heights' )
xlabel( 'Measurement number' )
hLine1.LineStyle = '--';
hLine2.LineStyle = ':';
ylabel(hAx(1),'Angles (radians)') % left y-axis
ylabel(hAx(2),'Heights (meters)') % right y-axis
end
```

Printed output:

Same an assign07a.m





ALL script files

Instructions for all assignment scripts:

- □ See Standards for Documentation of MATLAB Programs on the Canvas Resources page.
- □ Insert comments at the top and throughout each file.
  - o Include the follow comments at the beginning of this (and ALL) files.
    - % submitter's name, **GROUP # or "none"**
    - % other group members' names or "none"
    - % **`program file name`**, ex. assign02a.m
    - % due date of the assignment
    - % **statement about collaboration REQUIRED.**
    - % a short narrative about what the file does
  - o Use the algorithm given as comments throughout your program.
- □ Observe the instructor's rule for naming variables.
  - o Use ALL CAPS for constants variable names.
  - o Start other variables with lower case.
  - o Use descriptive variable names.
- □ Use Sample Input/Output as a guide.
- □ Code clarity:
  - o Indent blocks as needed. **Use Smart Indent.**
  - o Divide your solution program code into sections as noted in the algorithm. Use blank lines as needed to group statements.
  - o Use section comments as well as the algorithm step comments.
  - o Remove statements from previous assignments that do not apply to the current requirements.
- □ Use comments to show units.
- □ **Use the CONSTANT and variable names, not numbers.**
  Exceptions are incrementers (or counters) and numbers without identity.
- □ No extra output, i.e. use semicolons

GRADE OF ZERO **for a file if submitter name not part of Canvas group.**

(-3pts) No CURRENT GROUP# or "none".
(-3pts) For your own protection, type "none" for other group members if submitting alone.
(-5pts) Five point penalty for not joining your Canvas group.

(-5pts) Zero points for comments if no collaboration statement.

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points for not doing so.]

Submit via Canvas:

`assign07a.m`	MATLAB script file
`assign07b.m`	MATLAB script file
`printReport.m`	MATLAB function file