Introduction to Computing for Engineers 050IU

Script and Function Files

Dr. Nguyen Ngoc Truong Minh

MATLAB Function Files

- A MATLAB function file (called an **M-file**) is a text (plain ASCII) file that contains a MATLAB function and, optionally, comments.

- The file is saved with the *function name* and the usual MATLAB script file extension, **".m"**.

- A MATLAB function may be called from the command line or from any other M-file.

MATLAB Function Files

- The syntax for a MATLAB function definition is:

  function [val1, … , valn] = myfunc (arg1, … , argk)

  where *val1* through *valn* are the specified returned values from the function and *arg1* through *argk* are the values sent to the function.

- Since variables are local in MATLAB the function has its own memory locations for all the variables and only the values (not their addresses) are passed between the MATLAB workspace and the function.

MATLAB Function Files

- Three forms of the use of a function in MATLAB are:

```
>> VAR = function_name(arg1, arg2, …);
>> [VAR1,VAR2,...] = function_name(arg1, arg2, …);
>> function_name(arg1, arg2, …);
```

- A MATLAB function, like a mathematical function, is a rule where given a certain input or inputs, the rule tells you how to compute the output value or how to produce an effect (e.g., the plot function produces a figure).
- The inputs & outputs are called the “arguments” to the function.
- The inputs arguments can be an array.

MATLAB Function Files: Structure

function [***output_arg1, output_arg2, …] = function_name(****input_arg1,input_arg2, …****)***

% The line above is called the function definition line.

% Simple functions have only one output variable..

% A line that begins with a “%” symbol is called a comment.

% Comments are ignored by Matlab but necessary for humans.

{ Body of the function }

Note: In the body of the function, you MUST define all of the output variables or MATLAB will give you an error.

Example of a MATLAB Function File

```
function [ a , b ] = swap ( a , b )
% The function swap receives two values, swaps them
    temp = a;
    a = b;
    b = temp;
end

function [ r , g ] = swap ( c , d )
% The function swap receives two values, swaps them
     r = d;
     g = c;
end
```

Example of a MATLAB Function File

- To use the function a MATLAB program could assign values to two variables (the names do not have to be a and b) and then call the function to swap them. For instance, the MATLAB commands:

```
>> x = 5 ; y = 6 ;    [ x , y ] = swap ( x , y )
```

result in:

```
x =
   6
y =
   5
```

MATLAB Function Files

- Referring to the function, the comments immediately following the function definition statement are the "help" for the function. The MATLAB command:

```
>> help swap
```

The function swap receives two values, swaps them, and returns the result. The syntax for the call is [a, b] = swap (a, b) where the a and b in the ( ) are the values sent to the function and the a and b in the [ ] are returned values which are assigned to corresponding variables in your program.

Why write “functions” instead of “scripts”?

- Modular Programming: Break complicated tasks up into pieces (functions).
- Functions can “call” other functions: This means you don’t have to re-write the code for the function again and again.
- Variables in Functions are “local”:
  - *All variables in the function are “local” by default.*
  - *That means if you have a variable in your workspace with the same name as the variable in a function, then assigning a value to the variable in the function has no affect on the variable in the workspace.*
  - *That is, a function cannot accidentally change (destroy) the data in your workspace.*

What is a computer program?

Programming -

- **A process for obtaining a computer solution to a problem.**

- **A computer program is a sequence of instructions that tell the computer what to do.**

Programming Steps

1. Problem Definition

2. Analyze the problems

   (i.e., write down the appropriate equations, determine the user input and user output, ….)

3. Develop Algorithm

   (processing steps to solve problem)

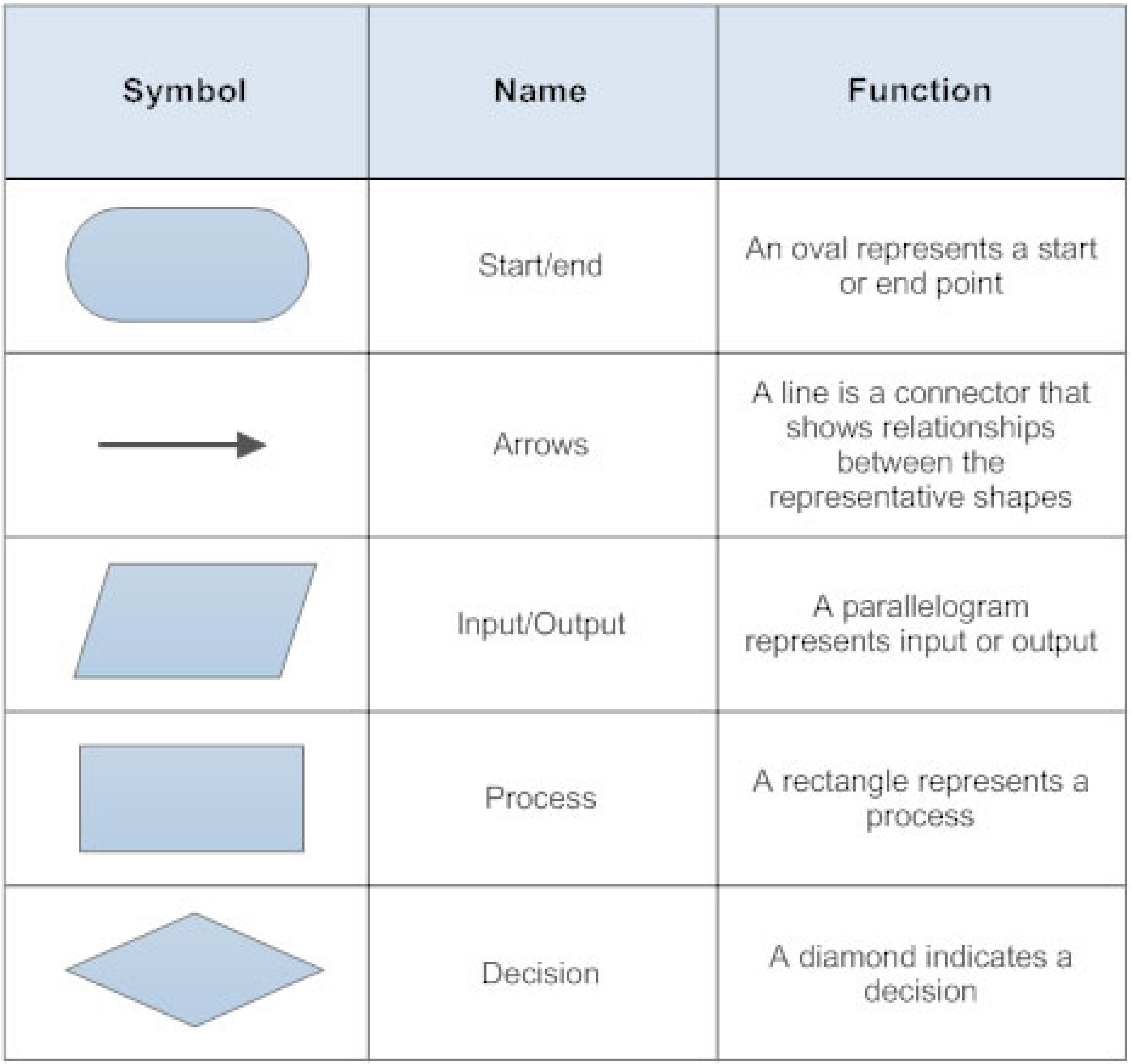
4. Write the "Program" (Code)

   (instruction sequence to be carried out by the computer)

5. Test and Debug the Code

6. Run Code

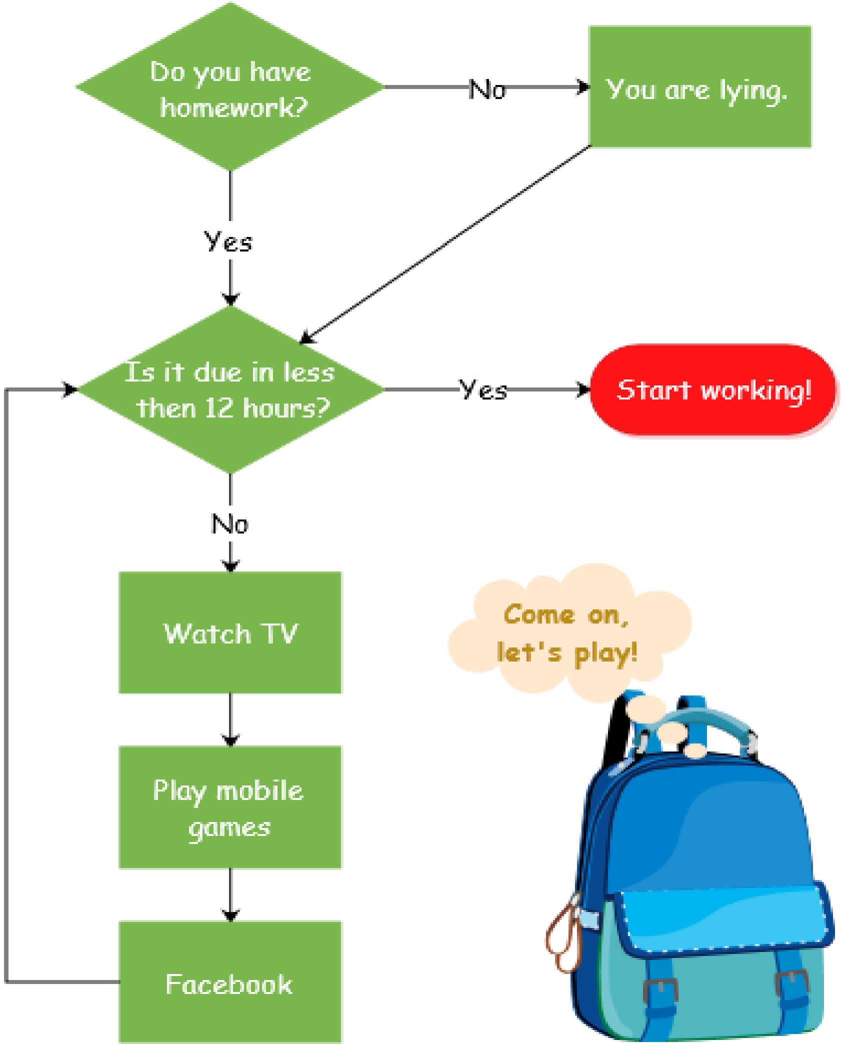
Flowchart



Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point
→	Arrows	A line is a connector that shows relationships between the representative shapes
▱	Input/Output	A parallelogram represents input or output
▭	Process	A rectangle represents a process
◇	Decision	A diamond indicates a decision

Flowchart

Should I do My Homework Now?



Do you have homework?

No

You are lying.

Yes

Is it due in less then 12 hours?

Yes

Start working!

No

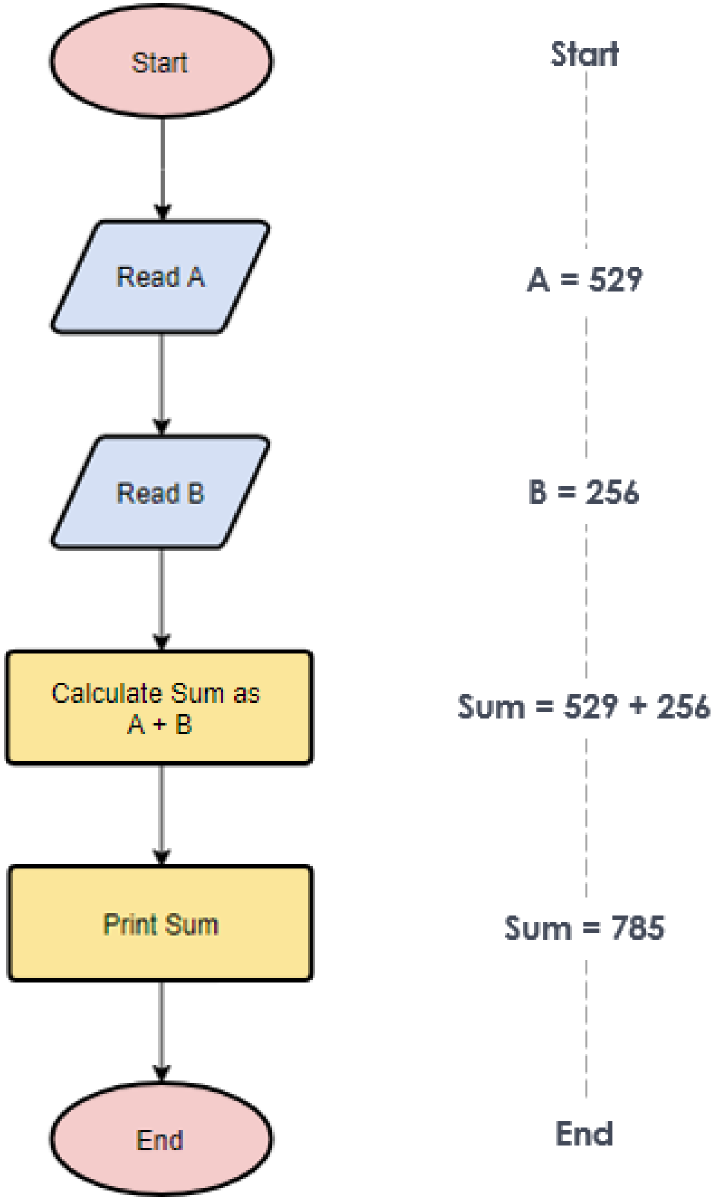
Watch TV

Play mobile games

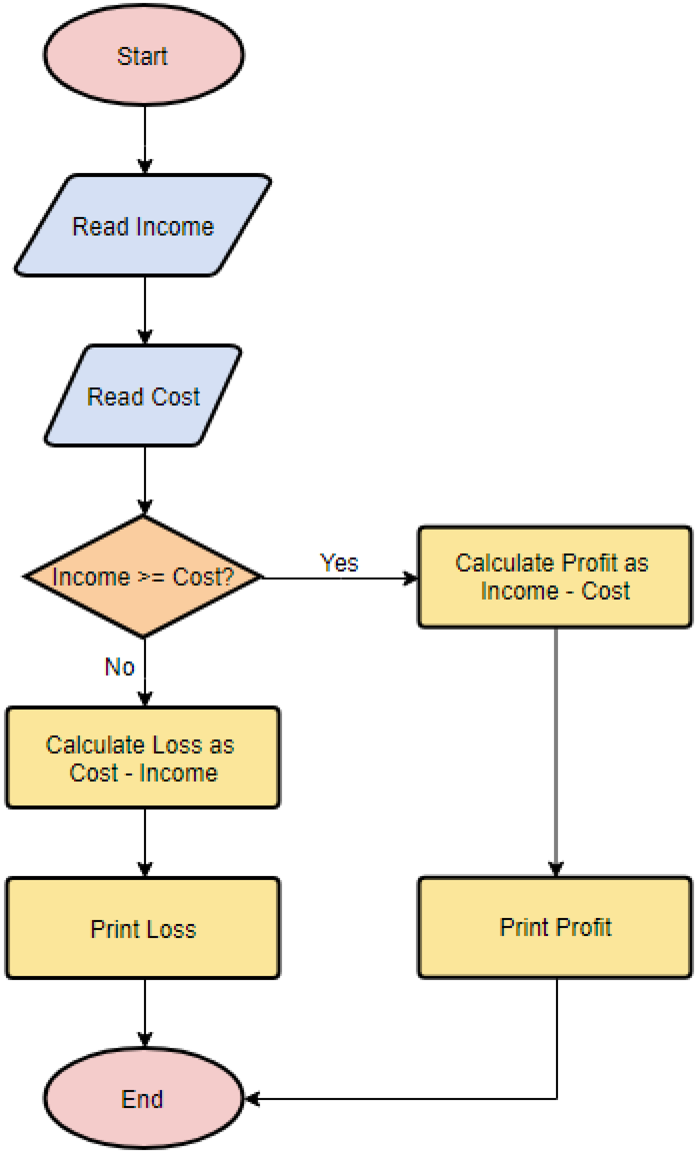
Facebook

Flowchart Example – Simple Algorithms

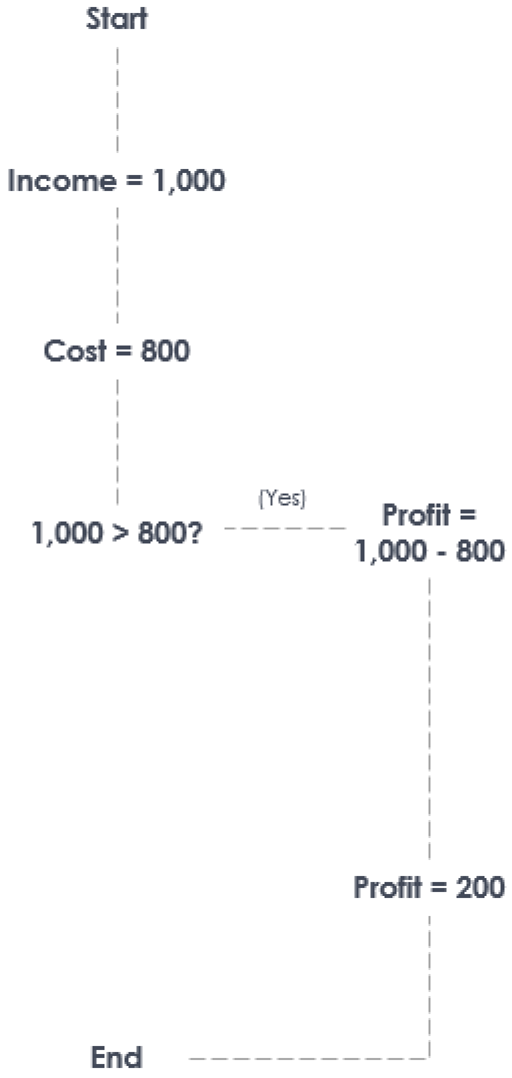
Find the sum of 529 and 256



Flowchart Example – Calculate Profit and Loss



Find the profit/loss when income = 1,000, cost = 800



Programming Example

1. Problem Definition

Write a function that converts a temperature in Fahrenheit to Celsius.

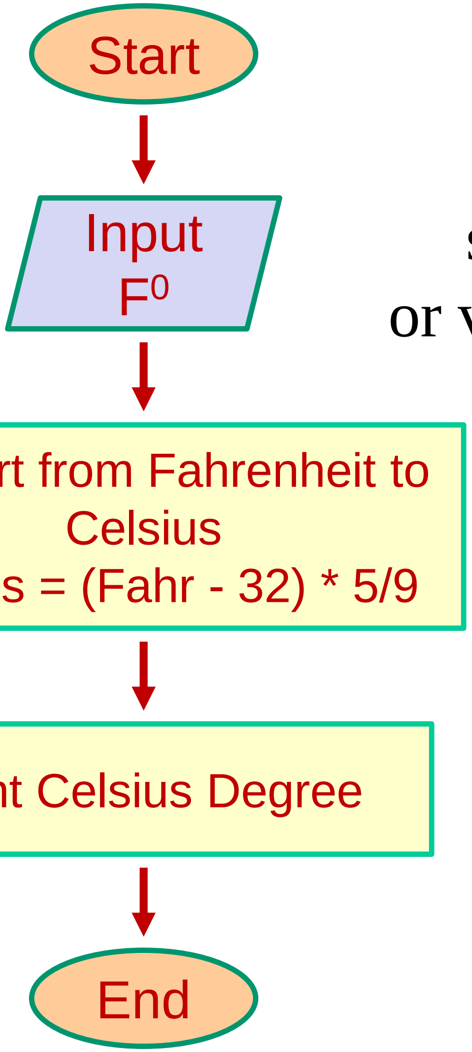
2. Problem Analysis

- User to input a temperature in Fahrenheit
- Output to the user temperature in Celsius

Use the fact that Celsius = (Fahr - 32) * 5/9

Programming Example

3. Develop Algorithm (processing steps to solve problem)


Start
Input
F0
Convert from Fahrenheit to Celsius
Celsius = (Fahr - 32) * 5/9
Print Celsius Degree
End

single number
or vector of numbers

Programming Example

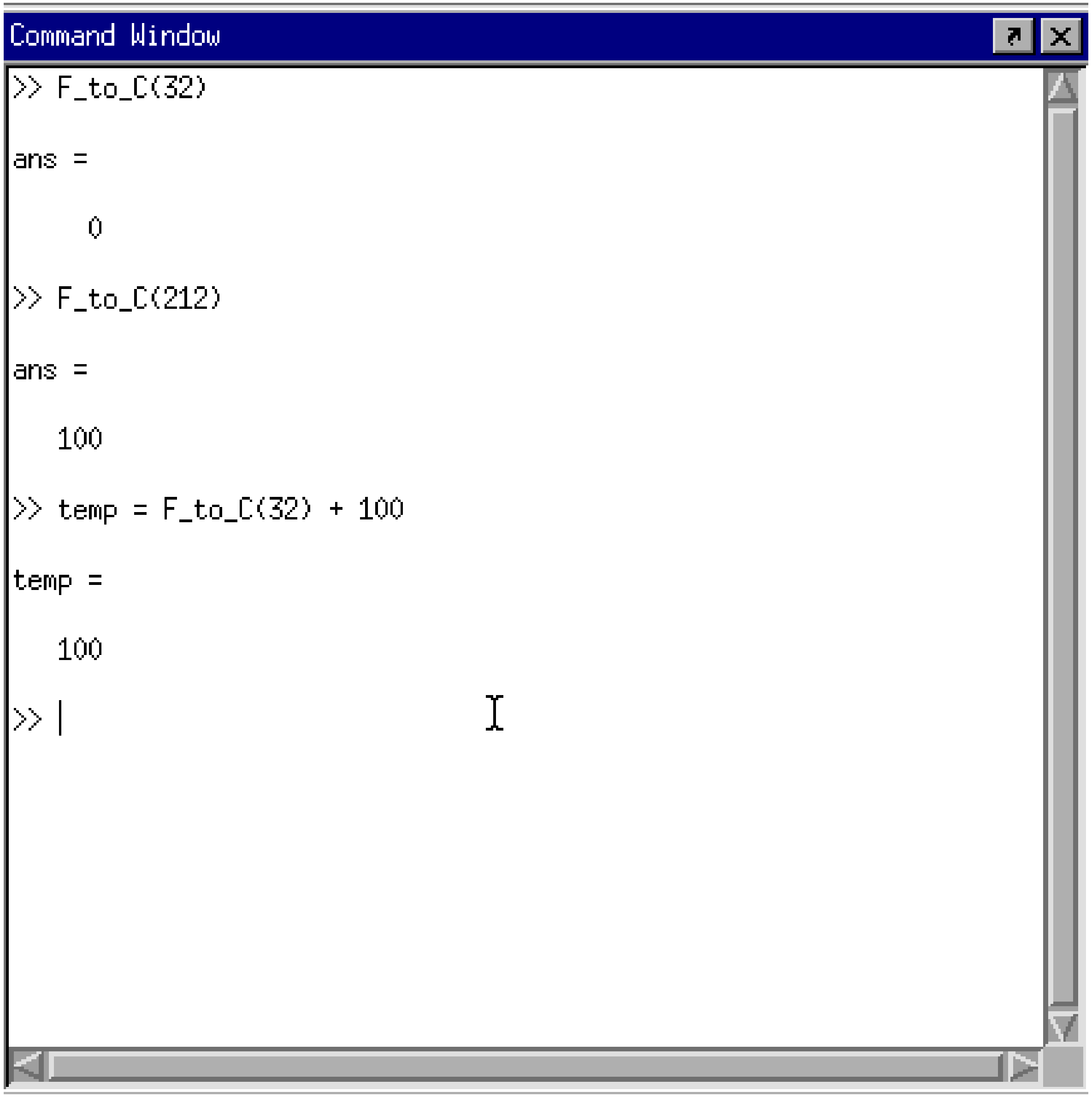
4. Write the “Function" (Code)
(instruction sequence to be carried out by the computer)

Any sequence of MATLAB commands can be put in a file. The file suffix should end with a “.m”. The sequence of commands will be executed (from the top of the file down, command by command). Open the Matlab editor and create a new file. Then type (and save) the following:

```
function celsius = F_to_C(fahr)
% This function converts Fahrenheit to Celsius.

celsius = (fahr -32)*5/9;
```

Click “File” and then “Save As” to name the file “F_to_C.m”.

Programming Example



```
>> F_to_C(32)

ans =

     0

>> F_to_C(212)

ans =

   100

>> temp = F_to_C(32) + 100

temp =

   100

>> 
```

Programming Example

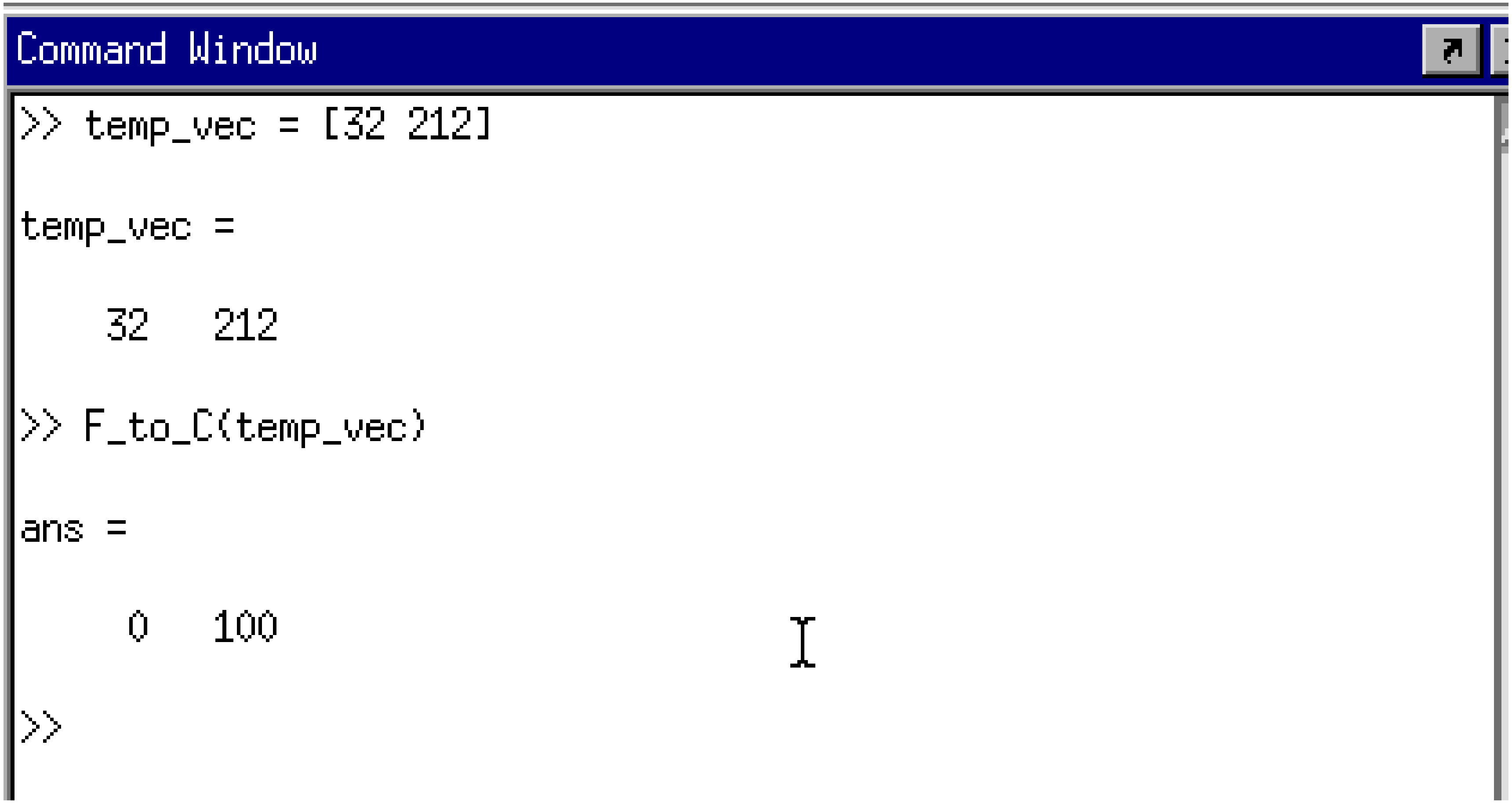
5. Test and Debug the Code

If the program works correctly then it has no “bugs”, so bring the MATLAB editor back up and close out the MATLAB program. Does the program work with only scalar input or does it work with vector values? (see next slide)

6. Run Code

Since two points determine a linear function, we know the function F_to_C works correctly.

Programming Example



```
>> temp_vec = [32 212]

temp_vec =

    32   212

>> F_to_C(temp_vec)

ans =

     0   100

>>
```

Programming Example

Improved documentation for the F_to_C function.

```
function celsius = F_to_C(fahr)
 % function celsius = F_to_C(fahr)
 % This function converts Fahrenheit to Celsius.
 % input fahr can be a scalar or vector of temps in degree F
 % output celsius is a scalar or vector of temps in  degree C
 % Programmer: ABC
 % Date:07/14/2020
celsius = (fahr -32)*5/9;
```

QUIZ (10 min)

1. Problem Definition

Write a script file that computes the time for a falling object to hit the ground.

2. Problem Analysis

Use the fact that

$$\textbf{Height(t)} = \textbf{height}_0 + \textbf{velocity}_0 * \textbf{t} + \tfrac{1}{2} * \textbf{g} * \textbf{t}^2$$

- *where height(t) is the height of the object at any given time t (in seconds)*
- *g is the acceleration due to gravity, -9.8 m/s^2.*
- *height$_0$ is the height at time t_0 = 0.*
- *velocity$_0$ is the velocity at time t_0 = 0.*

Therefore, to compute the time to impact, set height(t) = 0 and solve for time. This equation (after doing some algebra) can be written as:

$$0 = \textbf{Height(t)}=\textbf{height}_0 + \textbf{velocity}_0 * \textbf{t} + \tfrac{1}{2} * \textbf{g} * \textbf{t}^2$$

This is a quadratic formula in terms of the variable time. This can be solved to give:

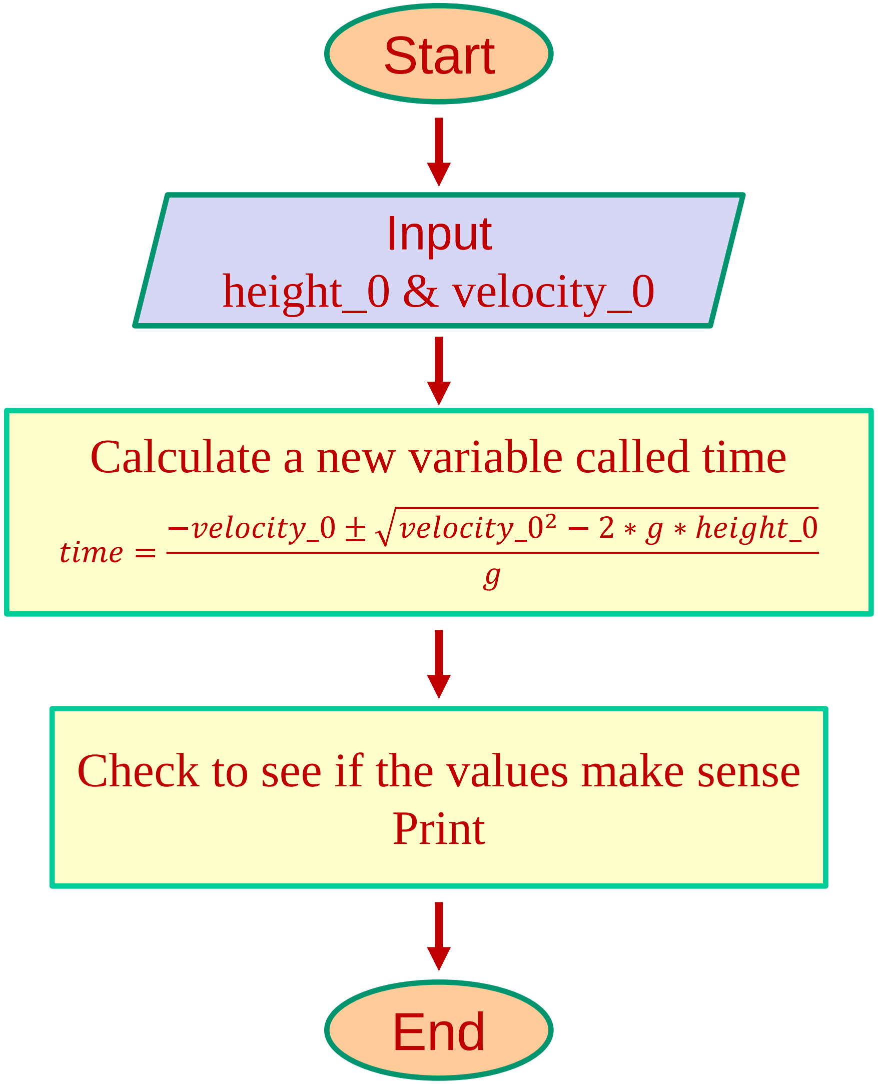
$$time = \frac{-velocity_0 \pm \sqrt{velocity_0^2 - 2 * g * height_0}}{g}$$

User inputs: initial height (height_0) and initial velocity (velocity_0)

User outputs: time to hit ground (time)

Programming Example

3. Develop Algorithm (processing steps to solve problem)



Start

Input
height_0 & velocity_0

Calculate a new variable called time

$$time = \frac{-velocity\_0 \pm \sqrt{velocity\_0^2 - 2 * g * height\_0}}{g}$$

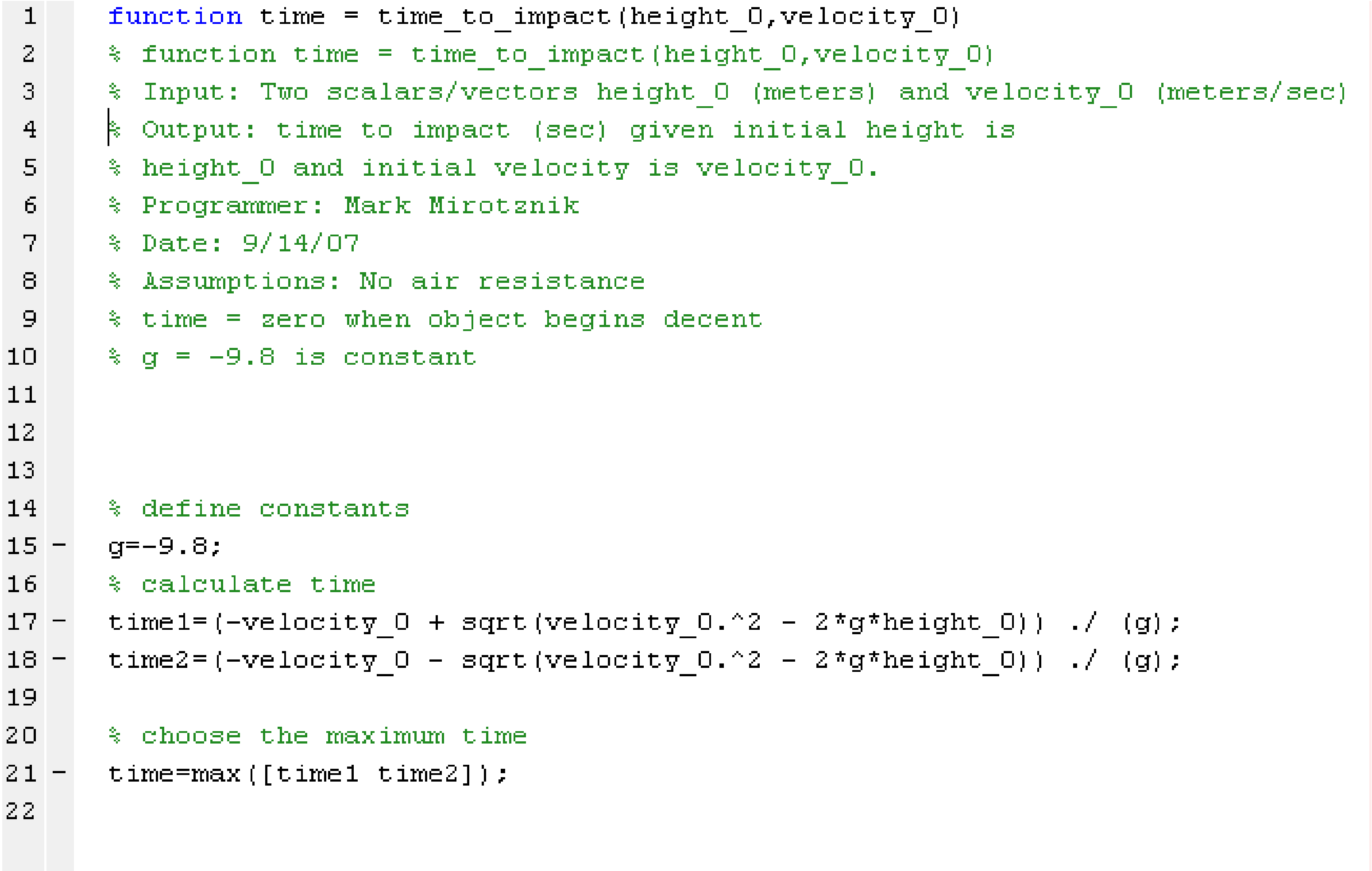
Check to see if the values make sense
Print

End

Programming Example

4. Write the “Function" (Code) (instruction sequence to be carried out by the computer)

Use the Matlab editor to create a file “time_to_impact.m” .

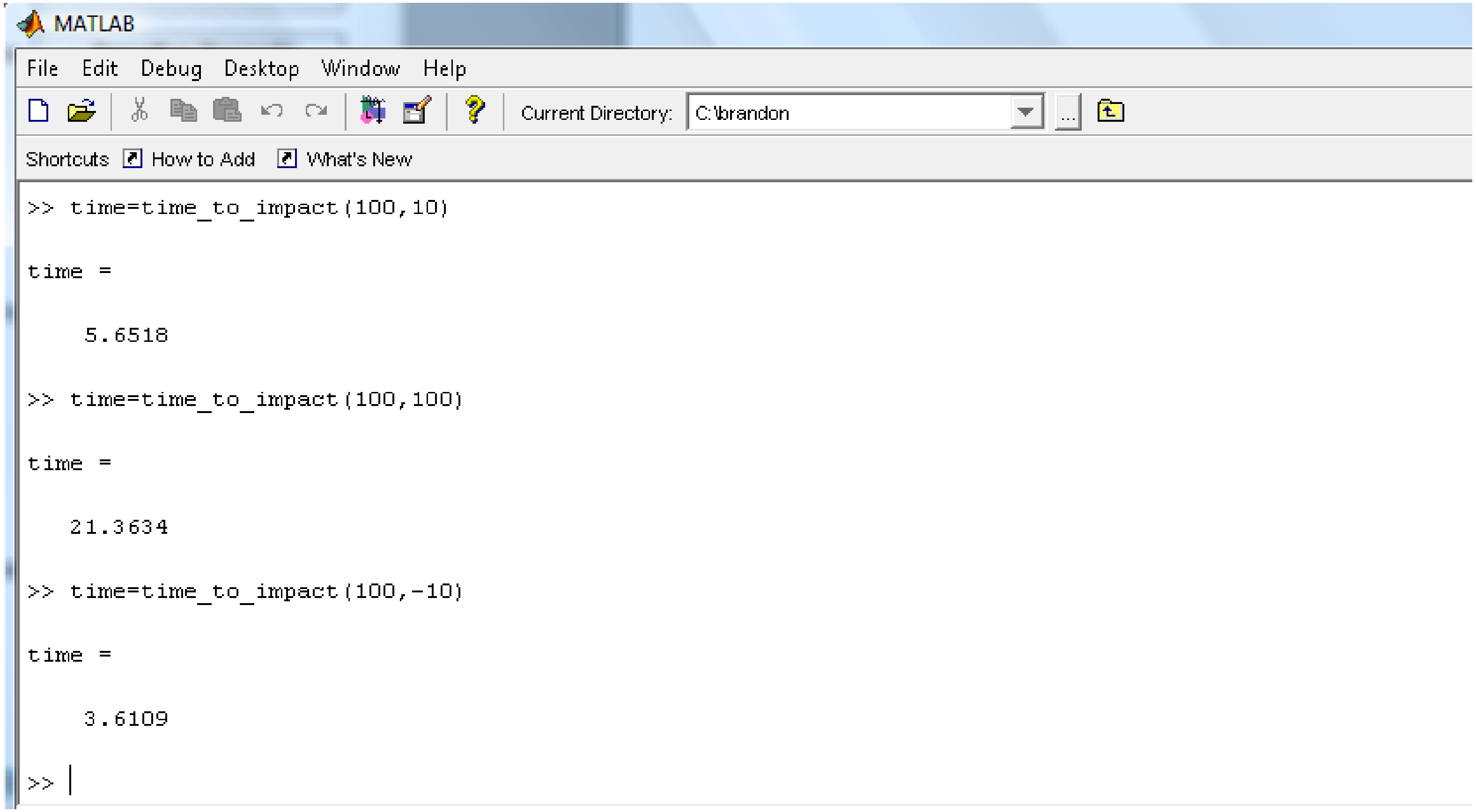


```
function time = time_to_impact(height_0,velocity_0)
% function time = time_to_impact(height_0,velocity_0)
% Input: Two scalars/vectors height_0 (meters) and velocity_0 (meters/sec)
% Output: time to impact (sec) given initial height is
% height_0 and initial velocity is velocity_0.
% Programmer: Mark Mirotznik
% Date: 9/14/07
% Assumptions: No air resistance
% time = zero when object begins decent
% g = -9.8 is constant



% define constants
g=-9.8;
% calculate time
time1=(-velocity_0 + sqrt(velocity_0.^2 - 2*g*height_0)) ./ (g);
time2=(-velocity_0 - sqrt(velocity_0.^2 - 2*g*height_0)) ./ (g);

% choose the maximum time
time=max([time1 time2]);

```

Programming Example

5. Test and Debug the Code

Although the value of the function is assigned to the variable time, when you execute the function, you can assign the value of the function to any variable.(see next slide)



MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\brandon
Shortcuts How to Add What's New

```
>> time=time_to_impact(100,10)

time =

    5.6518

>> time=time_to_impact(100,100)

time =

   21.3634

>> time=time_to_impact(100,-10)

time =

    3.6109

>> |
```