CS 221 Lecture

Tuesday, 4 October 2011

"There are 10 kinds of people in this world: those who know how to count in binary, and those who don't."

Today's Agenda

1. Announcements
2. You Can Define New Functions
3. Arrays Let You Name a Group of Related Values
4. Repetition (Iteration) Adds Power
5. Quiz

1. Announcements

- Remaining Quiz Dates:
  - In class: 25 October, 22 November
  - In lab: 3 November, 1 December
- Final Examination:

  10:30-12:30 Thursday, 15 December
- Midterm grades available nlt 21 October

2. You Can Define New Functions.

(Text Section 3.4)

- We've seen scripts (m-files)
  - Save and name a computation so it can be repeated
- Problem:
  - If you want to vary the values used in the computation, you have to read them from the keyboard (or elsewhere)
- What we want:
  - Define a computation that is a function of some input variables

  Note: "input variables" = "parameters" = "arguments"
  - and get the result out (without printing it)
- Like `sqrt( )`, `sin( )`, `input( )`, etc.

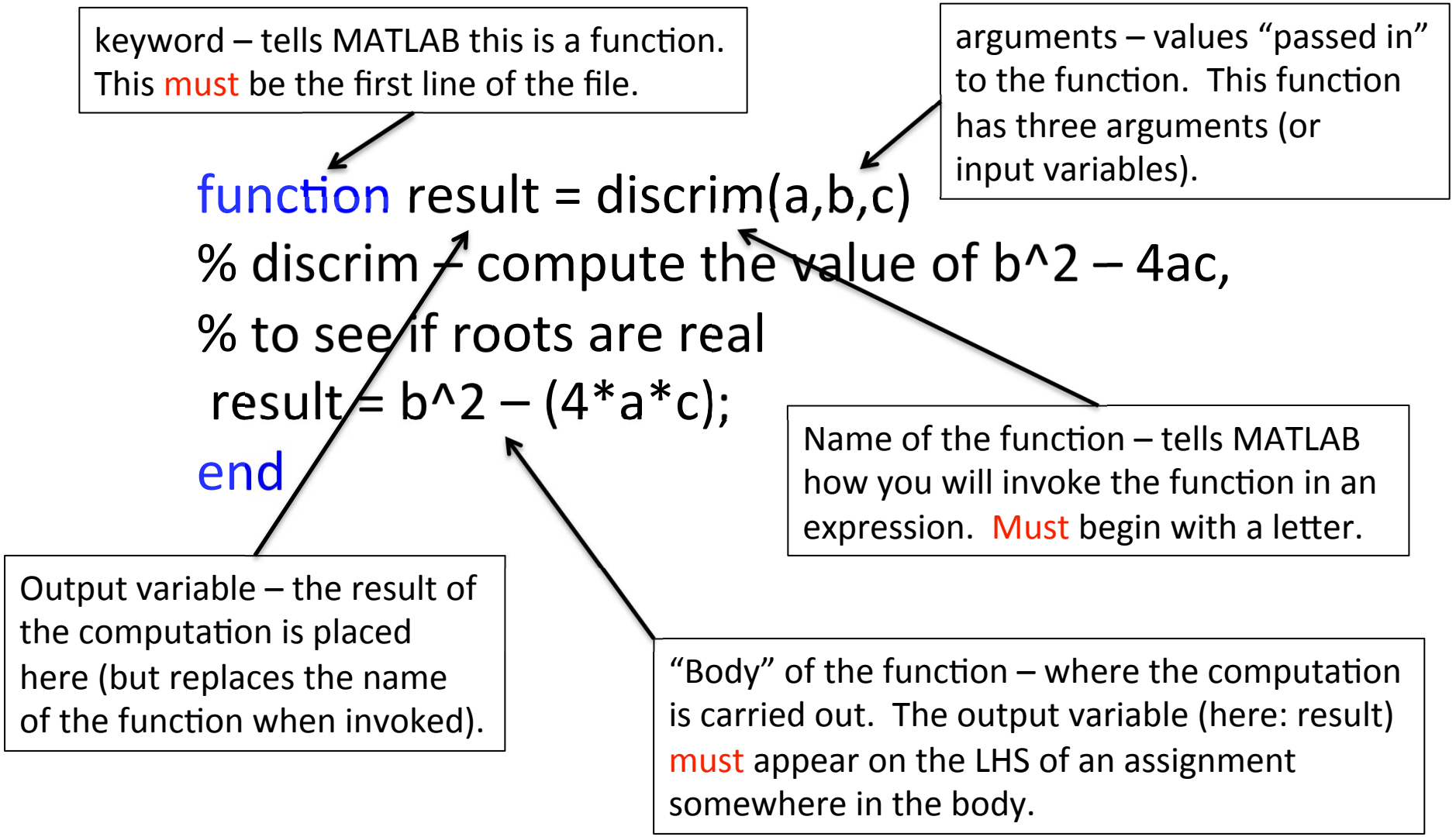
You Can Define New Functions.

- Function m-files allow you to do this!
  - First line has the form:

    function <output vars> = <name>(<input vars>)
- Example:
  - In our quadratic equation solving script:

    compute $b^2 - 4ac$ to check whether real roots exist
  - Turn this into a function: discrim(a,b,c)
    - Put the commands to do the computation in a file discrim.m
    - Make sure MATLAB can find the file

Example: Defining discrim()

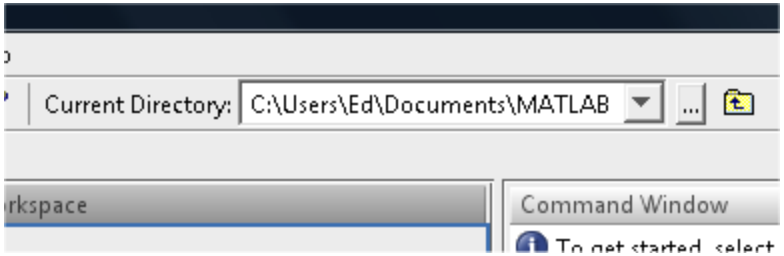


Invoke a Function by Writing Its Name

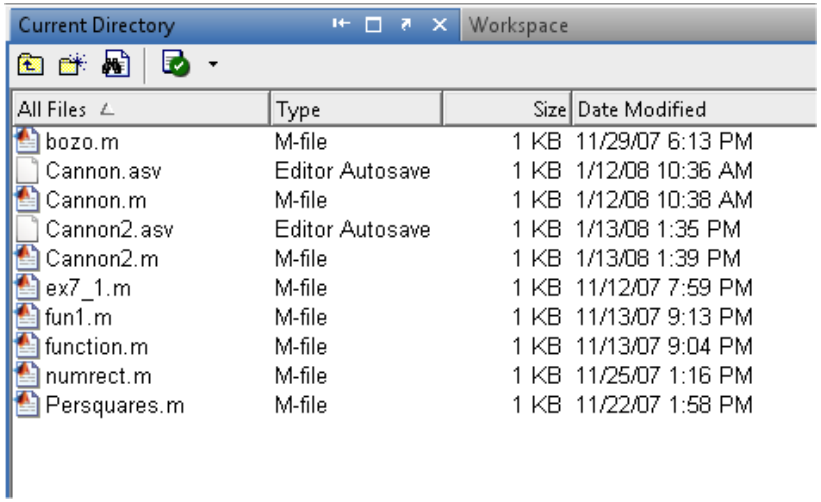
```
>> a = 3;
>> b = -4;
>> c = -2
>> if discrim(a,b,c) >= 0
>>     r1 = (-b + sqrt(discrim(a,b,c)))/(2*a)
>> else
>>     disp('Sorry, no real roots');
>> end
```

Current Directory

- At the top of the screen is the name of the current directory, where by default your new m-files will be stored.

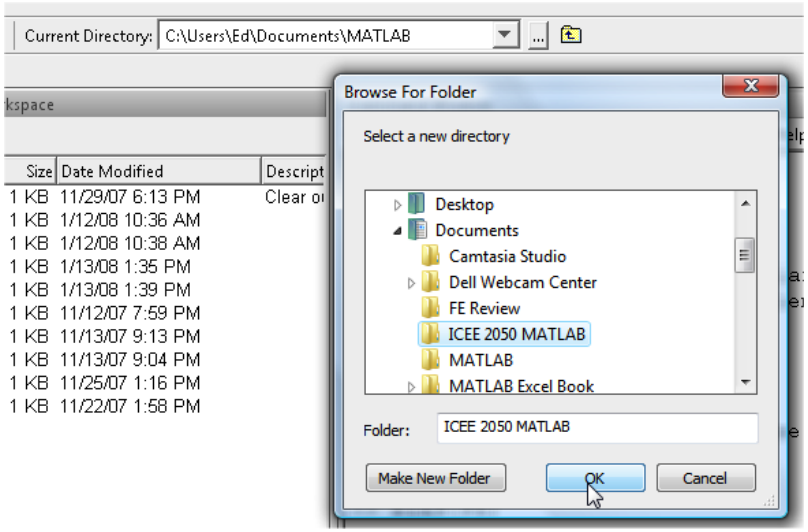


- A list of MATLAB files in the current directory can be displayed by clicking the "Current Directory" tab (this window can be toggled between the Workspace and Current Directory)



Current Directory

- You may want to store your MATLAB files for each class or project in a specific folder
- If so, create the directory in Windows and then browse to it from the MATLAB interface to set it as the current directory

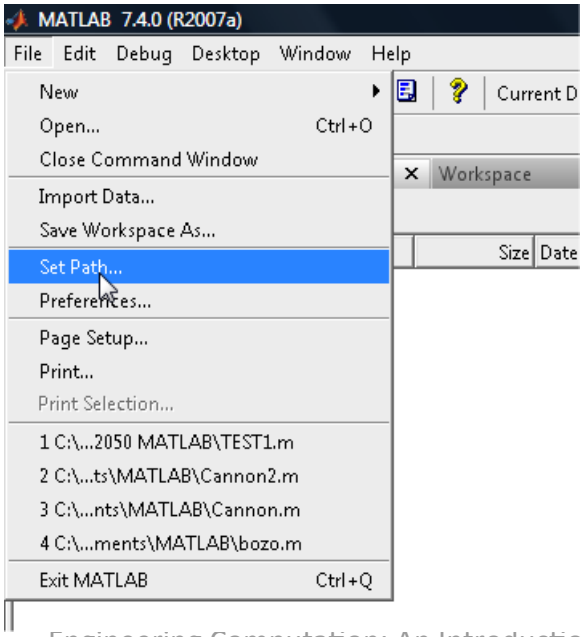


Path

- Files that you create in the new folder will run as long as that folder is set as the current directory in MATLAB
- However, if another folder is set as the current directory, files from the folder that you just created will not run unless its address is added to the MATLAB path
- The path is a list of locations that MATLAB searches to find files

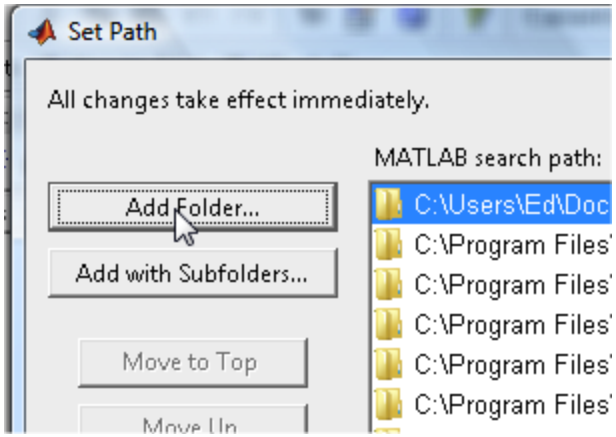
Path

- When you enter a command at the prompt, MATLAB looks for a file matching the command name, beginning with the first location in the path.
- To add you new folder to the Path, select File: Set Path…

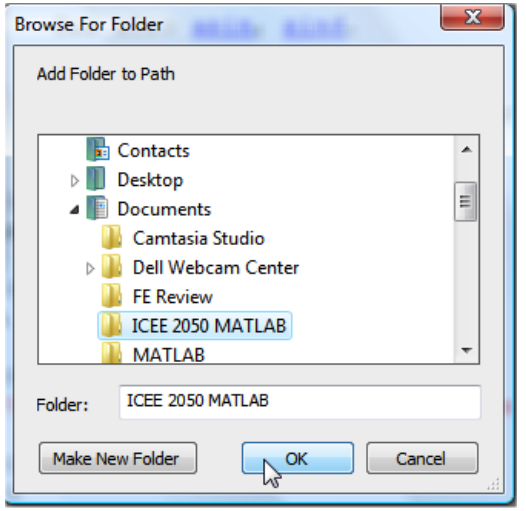


Path

- Select Add Folder…

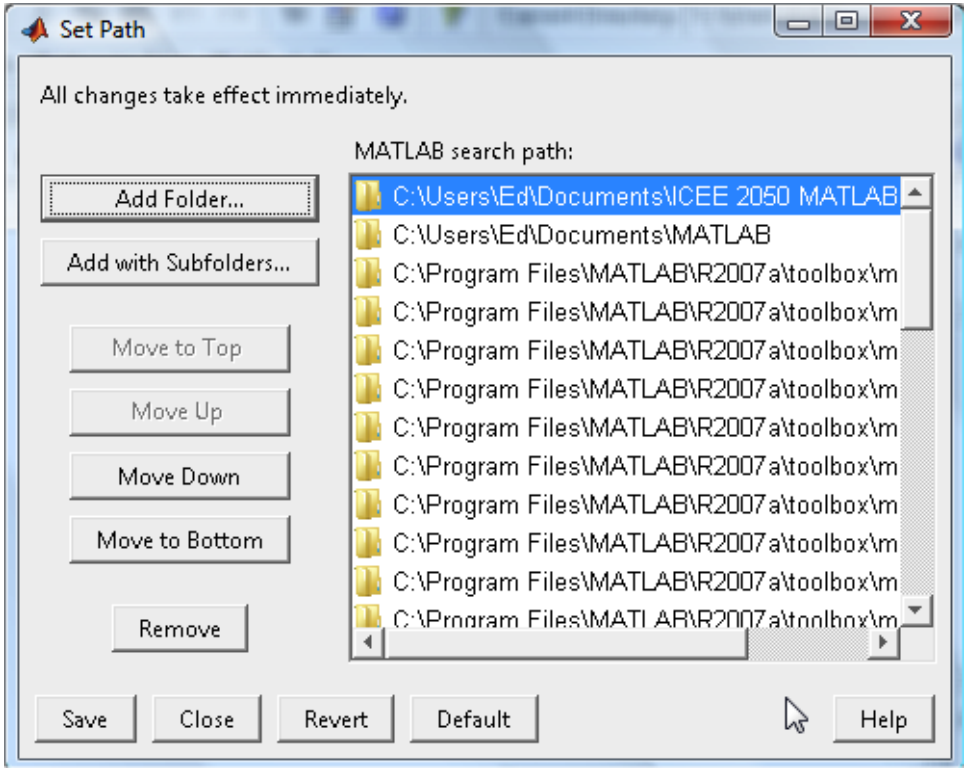


- Browse to your desired folder and click OK…



Path

- Click Save and Close. Note that your new folder is now the first location searched by MATLAB



3. Arrays Let You Name A Group of Related Values

(Text Sections 3.5, 3.6)

- Consider a civil engineer, taking soil measurements along a 1-km boundary line
  - One sample every 10m
  - A set of 101 samples
- How can you compute with these values in MATLAB?
  - E.g., you want to find max, min, average, ...
  - Don't want to have to come up with 101 different variable names!
    - sample1, sample2, sample3, ...

(Note: in Excel you don't have to name them at all!)

Arrays Can Be Two-Dimensional

- Suppose our civil engineer took measurements on a 100m x 100m grid (with 10m spacing)
  - 11 x 11 = 121 samples in all
- Could store in a 121-element array as before
  - Problem: figuring out which element goes where in the grid
- Solution: store the values in a 2-Dimensional array!
  - Location in array (row, column) corresponds to location in measurement grid

Reference Elements of Two-Dimensional Arrays with Two Indices

- Let "gridsamples" be an array with
  - 11 rows and 11 columns
  - "m x n array" always means m rows, n columns
    - Remember: row first, then column
- To access an individual element, give its row and column indices:
  - sample(3,4): element in third row, fourth column
  - Example: to round off the "middle" element:

    sample(6,6) = round(sample(6,6))

MATLAB Views Everything as a 2-D Array

- Scalars (individual values): 1 x 1 arrays
- 1-D arrays are called vectors
  - row vector: 1 x N array
  - column vector: N x 1 array

You Can Create Arrays in Many Ways

- Direct entry

  >> depths = [60.1 70.2 88.1; 55.2 33 44];

  – semicolons separate rows; spaces separate columns

- Built-in functions

  rand(m,n) creates an m x n array of random numbers between 0 and 1

  - rand(n) creates an n x n array

  ones(m,n) creates an m x n array of 1's

  zeros(m,n) creates an m x n array of 0's

- load command/function

  "load foo.txt" creates an array "foo" with contents

  See the help function...

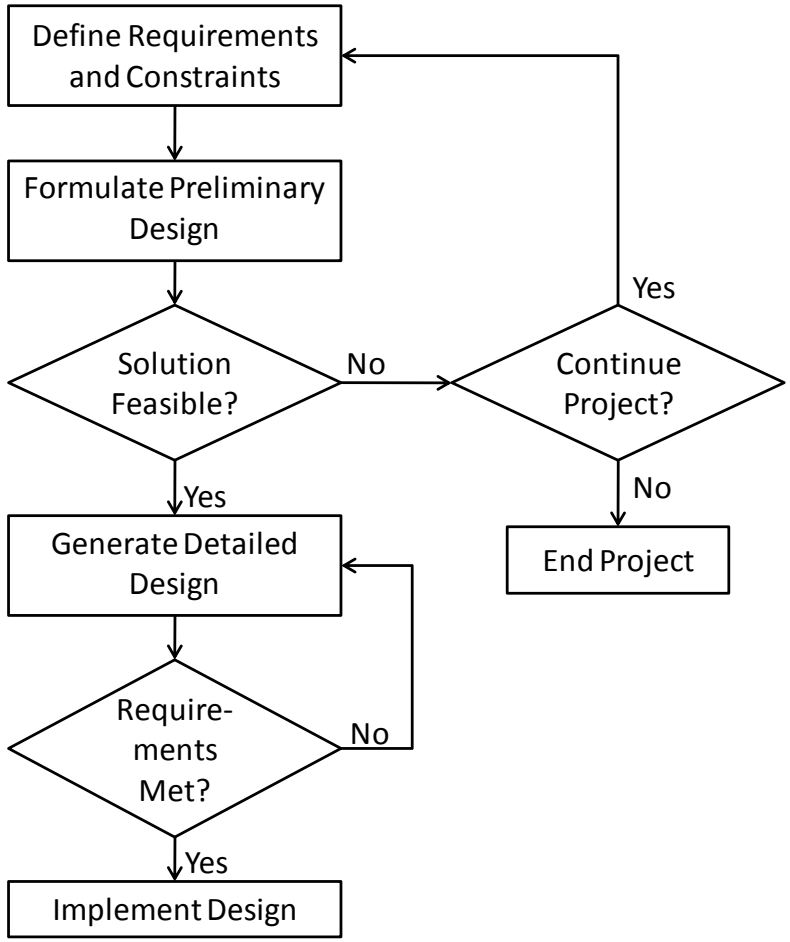
4. Repetition (Iteration) Adds Power

(Text Section 4.2.2)

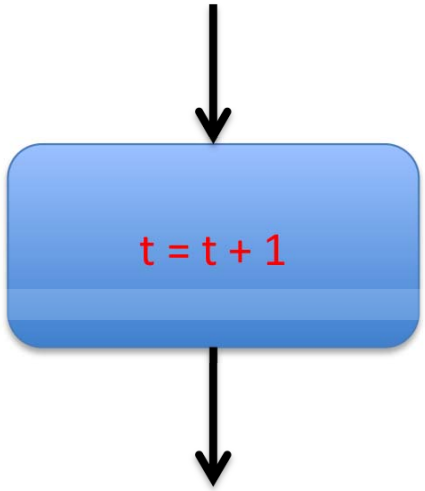
- So far, we have seen programs (scripts) that do each step once (or possibly not at all).
- To do interesting things, we need to be able to repeat steps of a computation over and over – in other words, to iterate
- All interesting programming tools have a way to do this
  - Iteration statements make the language as powerful as it can be (in a theoretical sense)

Recall Flowcharts

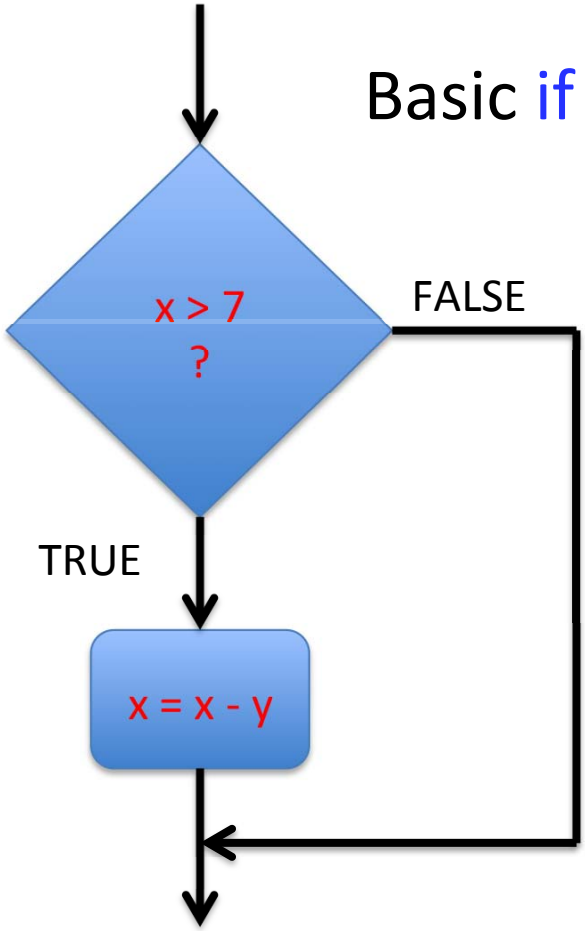
- Show the "flow" of a sequence of steps
- Boxes indicate basic steps; diamonds indicate "branch points"



Flowcharts Show a Sequence of Steps



Assignment
(or other step/command)

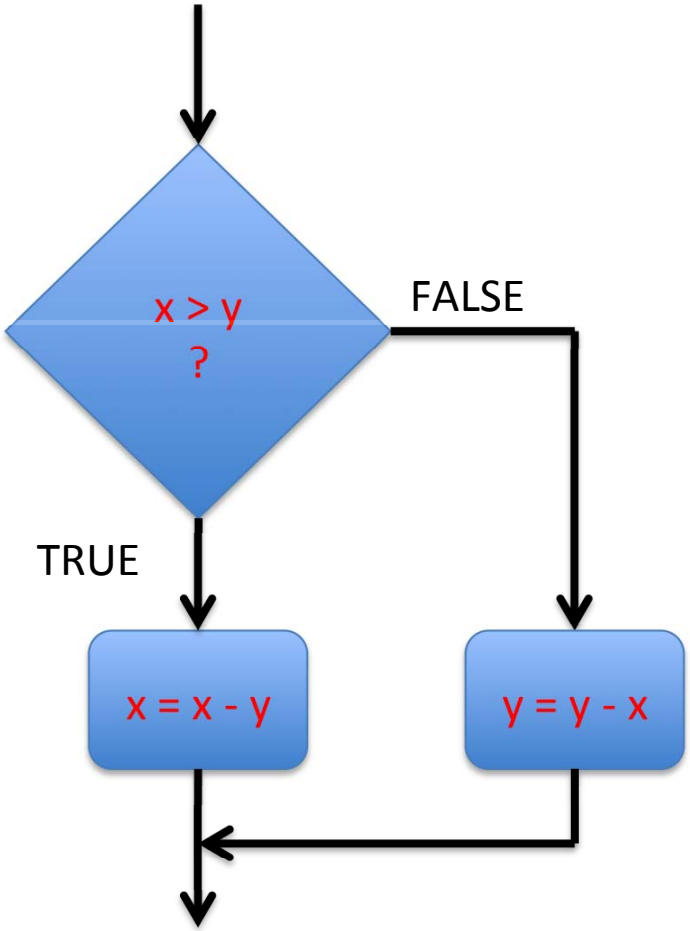


Basic if (no else)

```
if   x > 7
   x = x – y;
end
```

Flowcharts

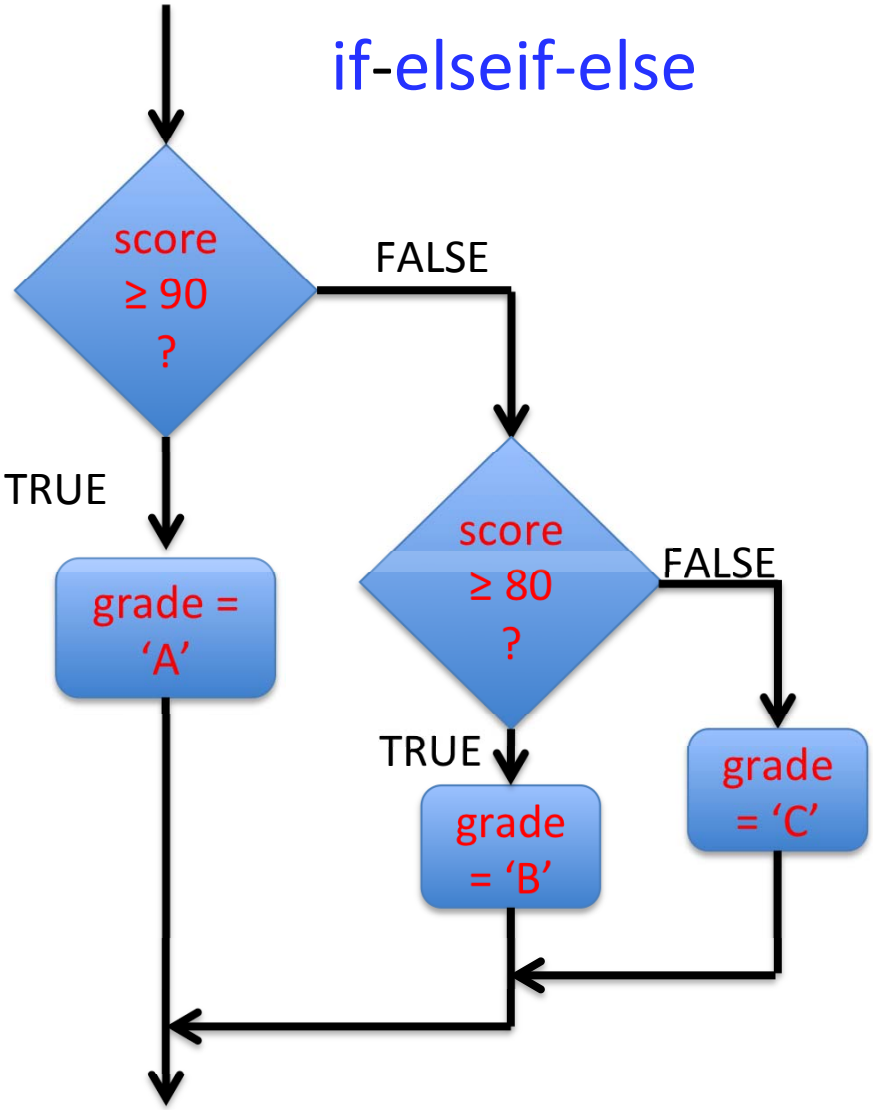
if-else



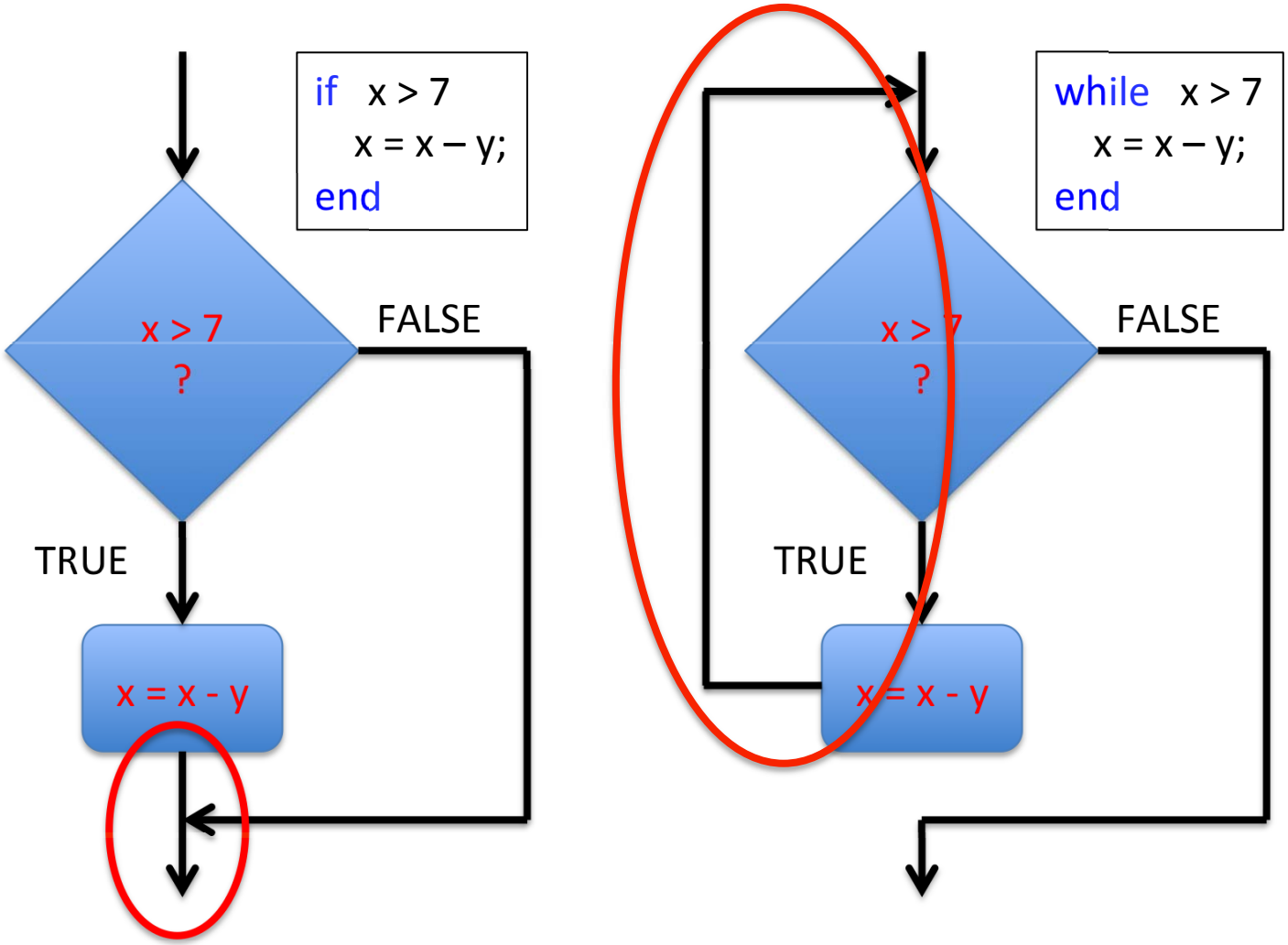

```
if  x > y
    x = x – y;
else
    y = y – x;
end
```

Flowcharts

if-elseif-else



while Statement in Flowcharts



if-statement

while-statement

The while-statement's syntax resembles the if-statement's

while <boolean expression>

    <statement>

end

As long as <boolean expression> evaluates to TRUE (=nonzero), keep executing <statement>

While statements are useful when getting input from the User.

- Instead of quitting (ending the script) when the user makes an error on input, let them keep trying until they get it right!

Summary of What You Learned

- You can define new functions
  - stored in m-files like scripts
  - pass in values via arguments/parameters/input vars
  - return values via output variables
- Arrays provide a way to refer to a group of values together
  - refer to individual values through indexing
- While-statements allow a computation to be repeated arbitrarily many times
  - precisely: until a condition becomes false