CS 221 Lecture

Tuesday, 27 September 2011

"A computer is like an Old Testament god, with a lot of rules and no mercy." -- Joseph Campbell

Today's Agenda

1. Administrivia
2. Assigned Reading
3. Lab Quiz Prep
4. In-Class Quiz Prep
5. Flowcharts
6. Arrays

1. Administrivia

- Lab Quiz Thursday
  - Open-book, closed-neighbor
  - Covers everything you have done in Lab so far
  - See the Practice Quiz
- In-Class Quiz Next Tuesday
  - Closed-everything
  - Covers everything in Lecture up to now
    - NOT arrays, flowcharts (today)

2. Reading the Textbook is Helpful.

- Chapter 1
  - 1.1 Projectile example, Analytic vs. "algorithmic" (Numerical) solutions
  - 1.3 Variables, precision, accuracy, significant digits
- Chapter 2 "Excel Fundamentals"
  - 2.1 Basic Excel user interface
  - 2.2 Entering, formatting data
  - 2.3 Formulas, operators, order of operations
  - 2.4 Built-in functions (SIN(), COS(), SUM(), etc.)
  - 2.5 Conditional expressions using the IF() function

Reading the Textbook is Helpful.

- Chapter 3 "MATLAB Fundamentals"
  - 3.1 MATLAB interface: windows, command line
  - 3.2 Computing with the command-line interface
  - 3.3 M-files (scripts), using input() to interact with user
- Chapter 4 "MATLAB Programming"
  - 4.1 Flowcharts (also in today's lecture)
  - 4.3 conditional ("branching") statements: if, if-else, if-elseif-else

Each section contains useful examples.

3. Lab Quiz is this Thursday.

- You will have 50 minutes (only).
- Two problems: one Excel, one MATLAB.
- Suggestion: Do the Practice Quiz!

4. In-Class Quiz is Next Tuesday.

- Last 30 minutes of lecture
- Pencil and paper (no calculators)
- Fair game:
  - Lecture material
  - Textbook material

  ...through last week's lecture (no flowcharts or arrays)

In-Class Quiz Topics

- Binary representation of numbers
  - Convert binary to decimal and decimal to binary
- Precision, accuracy and significant digits
  - Know the distinction between accuracy and precision
  - Know how to determine number of significant digits
- Boolean expressions and operators: and, or, not
- Conditional expressions
  - IF() function in Excel
  - if/if-else/if-elseif-else commands in MATLAB
- Predicting MATLAB and Excel behavior
  - How expressions are evaluated (order of operations)
  - Variable names in MATLAB
  - Recognizing errors in commands

Example Problem

In Matlab, suppose the following commands are typed into the command window:

```
a = 25;
b = 32;
c = 40;
c = c – b;
d = ‘triangle’ ;
e = ‘circle’;
f = ‘triangle’;
```

What does MATLAB print if you type:

```
>> if a < b && c < 12
       x = a + 2 * b
   else
       x = 2 * b – a
   end
?
```

Example Problem

In Matlab, suppose the following commands are typed into the command window:

```
a = 25;
b = 32;
c = 40;
c = c – b;
d = ‘triangle’ ;
e = ‘circle’;
f = ‘triangle’;
```

What does MATLAB print if you type:

```
>> if strcmp(d,e) || strcmp(d,f)
        x = b * 2 + c
    else
        x = sqrt(a + b – 1)
    end
```

?

Example Problems

- Convert 11,999 to binary
- Write down the decimal representation of 10110101101
- What is wrong with the following code?

```
whichshape = input( which shape?, s);
if strcmp(whichshape, hexagon )
    disp( Oh, goody )
else
    disp( you are a square )
```

More Examples

Which of the following is closest to the smallest positive double-precision floating point number that can result from a calculation in MATLAB?

a. -1.79769e+308
b. -2.22507e-308
c. 0.0000000001
d. 2.22507e-308
e. 1.79769e+308

More Examples

In Excel, some cells have the following values:

A1 = 25, A2 = 35, A3 = 20, A4 = -10

What will be shown in cell B1 if it contains the following formula(s):

=A1*A3-$A4

=IF(A1>A2,A3,A4)

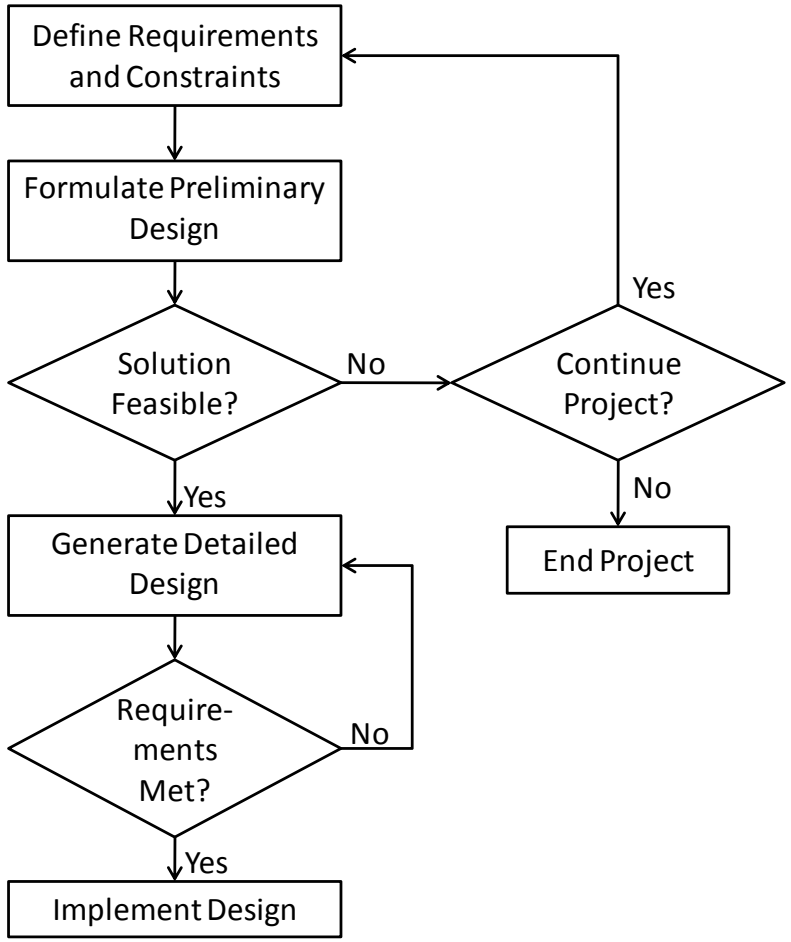
=IF(AND(A1>A2,A3>A4),A1*A2,A$4)

5. Flowcharts

- Flowcharts are diagrams that illustrate the paths followed in a sequence of computations
- Flowcharts are a great tool for planning complex algorithms
- Flowcharts are also very useful for documenting and explaining an algorithm, even relatively simple ones
- For many of the simple programs we will write, drawing a flowchart may seem to be unnecessary, but learning to create and read flow charts is a valuable skill

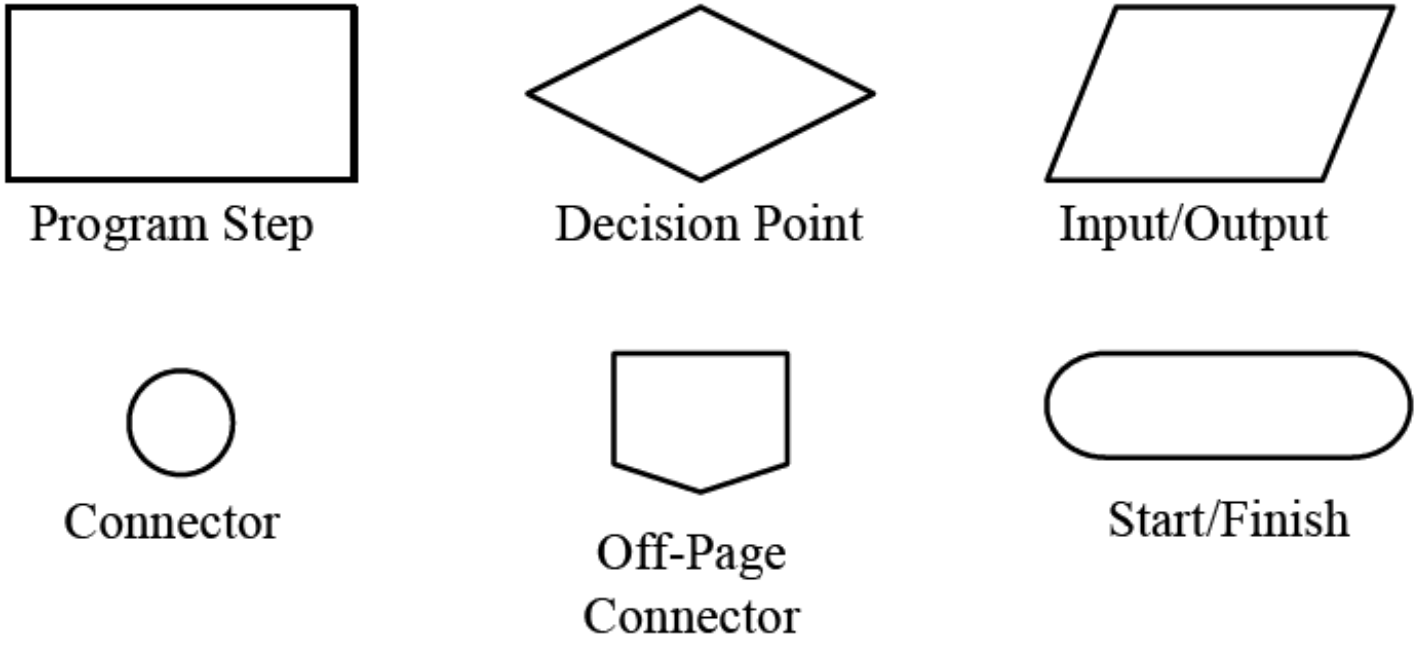
Flowcharts

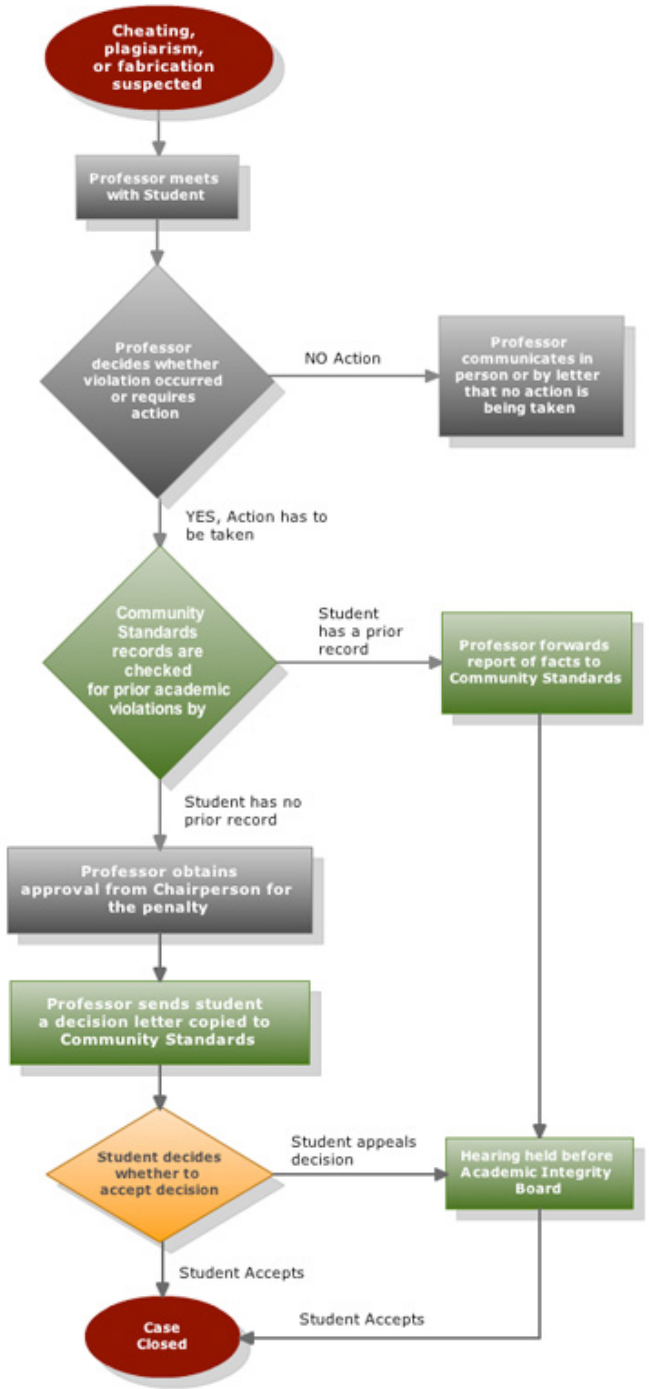
- Flowcharts are often used to illustrate process sequences in manufacturing operations and decision-making sequences in management
- Consider the flowchart of a company's product design process:



There is a Flowchart "Vocabulary".

Some symbols may vary, but the diamond-shaped Decision Point is pretty much a universal standard.





FLOWCHART OF ACADEMIC DISHONESTY PROCESS FOR GLIFFY SCHOOL OF DIAGRAMMING

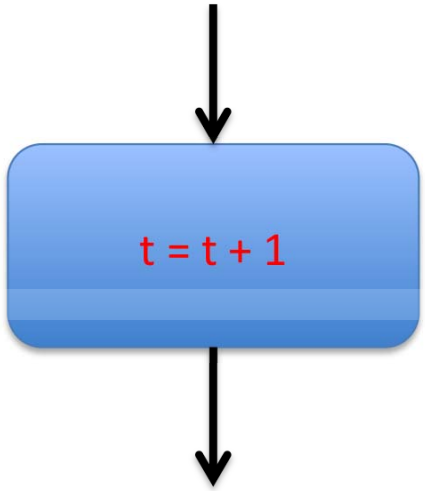
Student specific resolutions to the issues of AcademicDishonesty:

1. Failure of course.
2. Failure the assigment.
3. Redo assignment with the failing grade and a new grade being averaged together.
4. Redo the assignment with no other

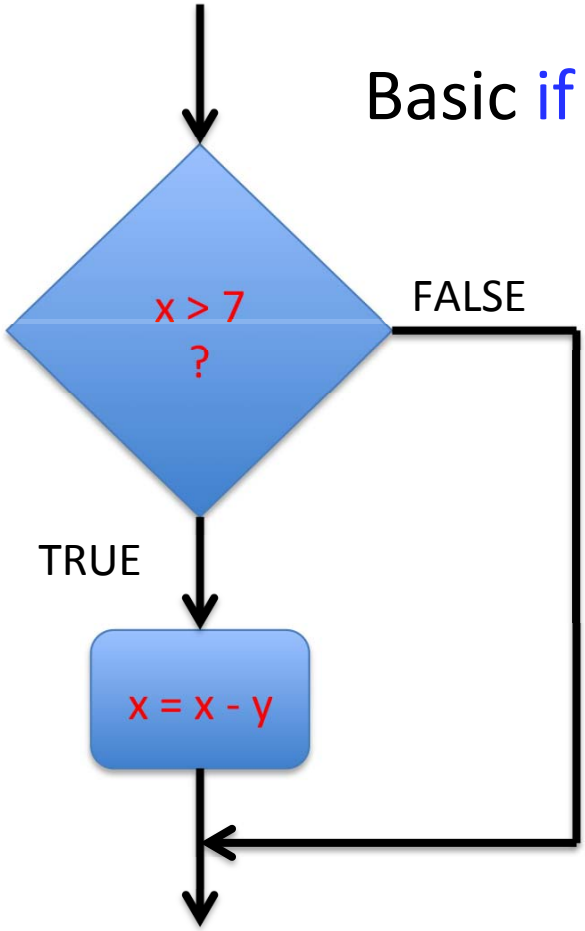
Legend

- RED = Start and Finish of Process
- GRAY= Faculty Action or Decision
- GREEN = Community Standards Involvement
- ORANGE = Student Decision

Flowcharts Show a Sequence of Steps



Assignment
(or other step/command)

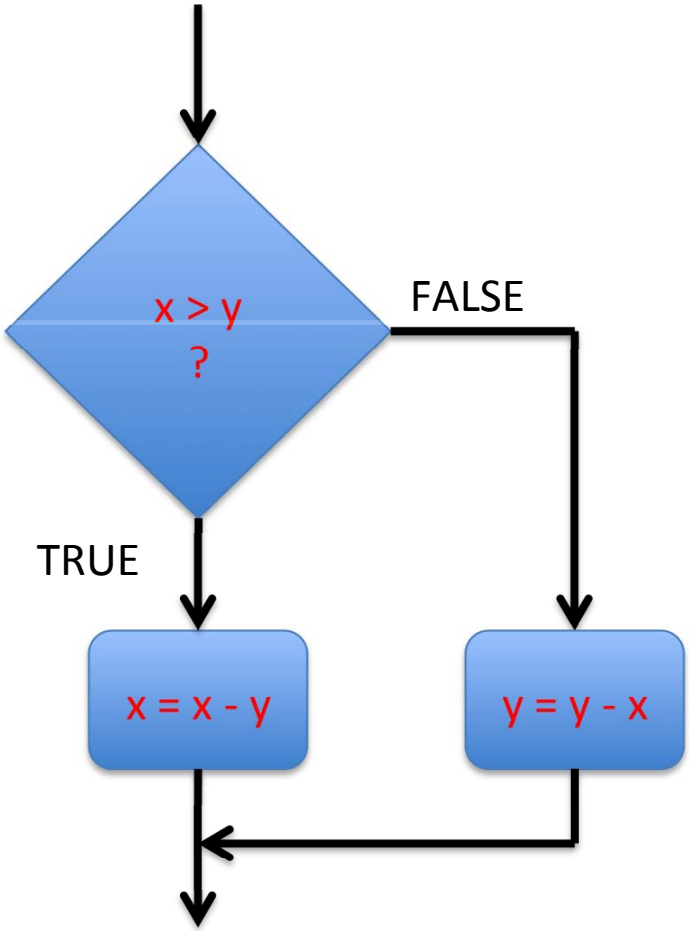


Basic if (no else)

```
if   x > 7
   x = x – y;
end
```

Flowcharts

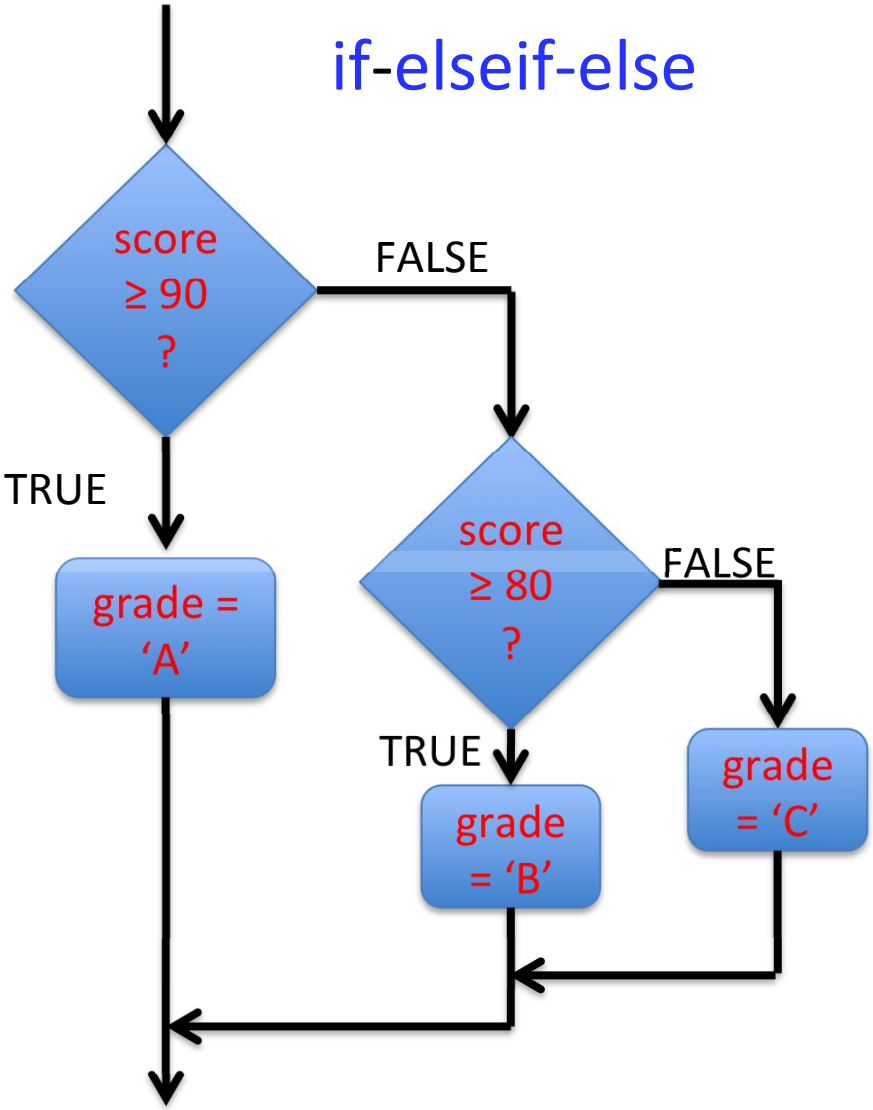
if-else




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

Flowcharts

if-elseif-else



6. Arrays in MATLAB (Text, Sections 3.5, 3.6)

- In a spreadsheet, we do not need to name every variable: the cell address (row/column) defines the storage location

	A	B	C
1	Initial displacement y_0	3	inches
2	Natural frequency ω	6.28319	rad/sec
3	Damping coefficient ξ	0.1	
4			
5	Damped frequency ω_D	6.25169	rad/sec
6			
7	Time t, seconds	y, inches	
8	0	3	
9	0.1	2.45016	
10	0.2	1.08578	
11	0.3	-0.5072	
12	0.4	-1.73	
13	0.5	-2.1875	
14	0.6	-1.8057	
15	0.7	-0.8209	
16	0.8	0.3423	
17	0.9	1.24526	
18	1	1.59461	
19	1.1	1.33032	

Storage locations for time and displacement values

Arrays Identify Multiple Values

- In MATLAB and other programming languages, the variables must be named and storage of results must be planned
- An array is a variable that contains multiple values
  - This allows you to refer to many values all at once
  - To refer to individual values, the array's name is followed by an *index* (or indices, for multi-dimensional arrays) in parentheses

Indexing Refers to Elements of Arrays

- Think of the index value as an "address"
- Important: Indices must be positive integers, and must start with one
- Example: To store these results, we would define

  $t(1) = 0, t(2) = 0.1, t(3) = 0.2 \ldots$

  $y(1) = 3, y(2) = 2.45016 \ldots$

Time t, seconds	y, inches
0	3
0.1	2.45016
0.2	1.08578
0.3	-0.5072
0.4	-1.73
0.5	-2.1875
0.6	-1.8057
0.7	-0.8209

Common Mistakes Using Indices with MATLAB

- Using an index of zero:

```
>> t(0) = 0
??? Subscript indices must either be real positive integers or
logicals.
```

- Using an index not an integer: setting $t(0.1)$ equal to a value results in the same error as above

Common Mistakes Using Indices with MATLAB

- Progressing indices by values other than 1. For example, if we take a temperature reading every 10 seconds, and set the first value as $T(10)$, then MATLAB automatically fills in zeros (the default value) for the first nine values of T:

```
>> T(10) = 100
T =
0     0     0     0     0     0     0     0     0   100
```

  (No error is reported, but this result is probably not what is desired)

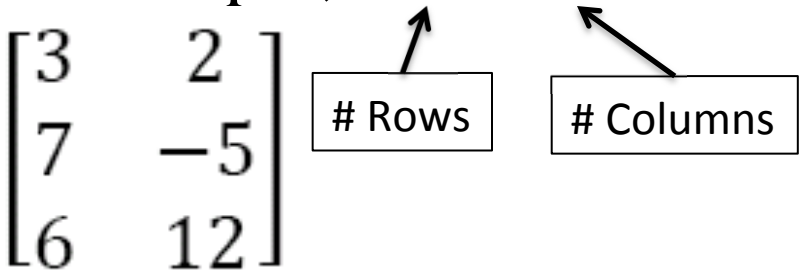
- All of these errors result from confusion between independent variables and indices. Thinking of indices as addresses helps avoid these errors.

Multi-Dimensional Arrays

- A variable may have more than one index. A two-dimensional array is similar to the data storage of a spreadsheet. Instead of a row number and a column letter, a two-dimensional array has two index numbers
- Examples: $t(1,5)$ $R(3,3)$ $temp(1,20)$
- Efficient way to store data: if both indices range from 1-10, then the number of values stored is 10 X 10 = 100

Matrices

- One- and two-dimensional arrays are also called *matrices*
- Many mathematical operations can be performed with matrices - the name MATLAB comes from "Matrix Laboratory"
- The size of a matrix is defined by the numbers of rows and columns. Example; a 3 X 2 matrix:



$$\begin{bmatrix} 3 & 2 \\ 7 & -5 \\ 6 & 12 \end{bmatrix}$$

Vectors

- One-dimensional matrices are referred to in MATLAB as *vectors*
- Vectors can be classified as *column vectors* (all values arranged in a single column) or *row vectors* (all values arranged in a single row)
- Caution: in mechanics classes, the term "vector" has a different definition: a vector is defined as a physical quantity which is defined by magnitude and direction

Entering Arrays in MATLAB

- Consider the matrix:

$$A = \begin{bmatrix} 3 & 5 & 1.6 \\ -2 & 0 & 4 \end{bmatrix}$$

- To enter this matrix in MATLAB:

```
>> A = [3,5,1.6;-2,0,4]
```

or

```
>> A = [3 5 1.6; -2 0 4]
```

Commas or spaces separate elements within a row, semi-colons separate rows

Arrays

- We will work with matrix math later, with applications such as solving simultaneous equations
- We will create and manipulate large arrays of results using programming *loops*. Loops control the execution of repetitive calculations, and the results are often stored in arrays