Remember:

* Semicolons
* == for equivalence, not assignment
* Matlab is in radians
* Use % to comment
* End if statements, while and for loops
* && = both must be true, || means either must be true, xor means exactly one must be true, ~ means not
* Update your test variable inside the while loop so it doesn’t run indefinitely

Other notes:

* Log is ln, log10 is log base 10
* Exp(x) is e^x
* 2/11 is 0.1818, 2\11 is 5.5

Vectors:

* X = 1:10 gives x = [1 2 3 4 5 6 7 8 9 10]
* X = 1:0.5:4 gives x = [1 1.5 2 2.5 3 3.5 4]
* Empty vectors: x = [] or x = 1:0
* Linspace(0,pi/2,10) gives a vector of 10 equally spaced points from 0 to pi/2 inclusive
* Logspace(0,2,10) gives a vector of 10 equally logarithmically spaced points from 10^0 to 10^2 inclusive (interval between points increases linearly)
* x = [1 2 3], x’ = [1; 2; 3]
* r(3) displays 3rd element of r
* r(2:4) gives 2nd, 3rd, and 4th elements
* r(1:2:7) gives the 1st, 2nd, and 7th elements
* r([1 7 2 6]) gives the 1st, 7th, 2nd, and 6th elements
* r([1 7 2]) = [] removes elements 1, 7, and 2 from the set
* table = [x’ y’] creates a matrix with row vectors x and y as the columns

strings:

* place in quotation marks: s = “Hi, Mom”

functions:

* size function returns dimensions of an array – 1st element is # rows, 2nd is # columns
* length function is length of the longest part of an array
* var = input(‘text’) saves whatever you type in as a response to text as the value of var
* disp([‘You are ‘ num2str(AgeYears) ‘ years old today!’]);
* var = dlmread(‘filename.txt’)

Operations:

* dot product of two vectors: dot(x,y)
  + .\* will multiply corresponding elements to get a new vector: x = [1 2]; y = [3 4]; x .\* y = [3 8]; dot(x,y) = 11
* A(2:3, 1:2) returns matrix with values that make up the intersection of the 2nd and 3rd rows and the 1st and 2nd columns (4 values)
* sum(r(3:end)) returns sum of all elements of r from 3rd one to the last one
* to find vector x such that A\*x = b, write x = A\b
* a(:,2) = [] deletes the 2nd column of a
* for loops
  + for i = j:k goes j, j+1, … , k
  + for i = j:m:k goes j, j+m, j+2m, … , k
  + for i = v moves through each element in vector v

Functions:

* Function output = name(input)
* Output can be single variable or list of variables listed in brackets
* Must assign a value to your output somewhere in the code
* Put @ in front of function names to call a function that already exists in a file
* Fh = @(x) 1-x or fh = @(x) (sin(x))^2
* Feval(fh, x) evaluates function handle fh at variable/number x

Random numbers:

* Seed your generator with the same thing to get the same random numbers
* X = a + (b – a) \* rand(); returns a random number between a and b instead of between 0 and 1
* X = rand(); y = N\*x (gives number between 0 and N); roll = ceil(y) or roll2 = 1 + floor(y) returns a uniformly distributed integer from 1 to N