Class notes for 8-20-2018

* Make sure to document your code effectively. You won’t remember everything.
* Don’t focus on efficient coding too much. Learn the basics and we’ll get better at it later.
* Functions:
  + Creates a function “object” that won’t execute the code until “called”
  + Basic form:
    - def [name] ([parameters]):
    - indent code block to establish scope of the function
    - parameter can have default value (ie def add (x, y=0):)
  + you can batch your returns and your declarations
    - i.e. A, B = function(x,y)
      * As long as the return returns two items (i.e. return 1,2), you can batch call a function.
      * Can do different variable types.
  + You can create functions in a function but you’ll only be able to use the function within the function. Just don’t do it.
  + Functions can see variables outside but outside can’t see in.
  + Functions without a return will return a None value.
  + Once the function is complete, the local variables created will disappear. You can modify a global variable outside of the function by using variable.append(value). Don’t make a habit of doing it though.
  + Think of functions as taking out a new scratch paper to run calcs. Once it’s done, you toss that paper.
  + Begin practice of setting your main as a function (def main() ). Include at the end of the file “if \_\_name\_\_ == “\_\_main\_\_”:  
     main()”
    - This creates a failsafe that allows you to even import your main function and to prevent accidental occurrences of not running your main when it isn’t a main.
    - Assists with debugging and encapsulating the code