The person responsible for version control will follow the instructions in <https://github.com/viljoed/gis4x07/raw/master/GIS4107_ExerciseSetup.docx> to create a private GitHub repository called gis4107-day08 and clone this repository to acgis\gis4107\_Intro2Prog\day08\lab.

Each of the functions described below will have an associated test function with a “test\_” prefix. For example, the getInitials function will have a test\_getInitials function containing a number of calls to getInitials. The main function will contain a call to the test functions (e.g. test\_getInitials() with no arguments). Use the ExerciseTemplate.py provided as a starting point.

1. CSIS has discovered an encoded string in text files that is important to national security (“Tx6op3”). They would like you to write a Python function called hasXcode that has one parameter named inText. This function will return True if the encoded string is found in inText and False if it is not found. Create a ScanDocs.py module and add this function to it. Add a function called test\_ hasXcode with calls to hasXcode. Add a call to test\_ hasXcode in main.
2. After using your function, CSIS decides it would also be useful to create a function to return the position in the string where the encoded string begins. Add a function called getXcodePosition to ScanDocs.py that has one parameter named inText. This function will associate 1 with the first character in inText. If the code is not found, this function will return -1.
3. After using this function, they discover other patterns that they want to find. They request a more generic function called getPatternPosition that has two parameters named pattern and inText. Add this function and its test function to ScanDocs.py. Like getXcodePosition, this function will associate 1 with the first character in inText. If the pattern is not found, this function will return -1
4. Create a StringUtils.py module. Add a function called getInitials that has a parameter called fullName and will return the initials. Here are some example argument / return value pairs: John Samuel Wobbly / J.S.W., Dylan McDermott / D.M, Nora Young / N.Y. that you should use in your tests.
5. Phone numbers in the U.S. and Canada have the form NNN-NNN-NNNN. Create a PhoneUtils.py module. Add a function called isValidPhoneNumber that has one parameter named phoneNumber. It returns True if the phone number has all numbers and has the format shown above (i.e. 3 numbers, a dash, etc). It returns False if any character is not a number (except the dashes) or it does not have the above format.
6. Some advertisements have phone numbers like 613-6BY-JOVE. Add a function to PhoneUtils.py called phoneNumberHasLetters that has one parameter named phoneNumber. It returns True if the phone number contains 1 or more letters and False if it does not. If it does not have the NNN-NNN-NNNN format, return False. If it has the NNN-NNN-NNNN format, the first three digits are numbers, and any of the remaining characters are letters, return True.
7. Your company has decided to add an underscore and an upper or lower case three letter suffix to feature class names to distinguish between point, line, and polygon (PNT, LIN, PLY). For example, PROVINCES\_ply or Provinces**\_PLY**. They are building a feature class reporting tool in Python and would like one of the columns to contain “Point”, “Line”, or “Polygon” depending on the suffix used in the name of the feature class. You have been asked to code a function called getFeatureTypeFromName that has one parameter named fcName (for the feature class name) and will return “Point”, “Line”, or “Polygon” depending on the suffix. If it is not one of PNT, LIN, or PLY (or pnt, lin, ply) and not preceded by an underscore ( \_ ), then the function will return “Unknown”. Create a module called FCReporter.py and add this function and associated test functions to this module. Remember to write enough test calls to ensure your function is working properly.  
     
   NOTE: The name of the feature class does not have a constant length.
8. A GPX file (<http://www.topografix.com/gpx_manual.asp>) contains latitude and longitude coordinates in a string like the following:

<trkpt lat="45.3888995" lon="-75.7472631">  
  
Create a GpxUtils.py module. In this module, write a function called getCoordsFromGpx that has one parameter (gpx) will return both the longitude and latitude values. Given the above GPX trkpt element, this function would return (-75.7472631,45.3888995). If the GPX in the function call does not contain trkpt, return None.  
  
NOTE 1: Assume the lat and lon are numbers with decimals but do not assume that the number of decimals is constant.  
NOTE 2: A function can return more than one value. The syntax is **return value1, value2**

1. Assume a file contains a list of longitudes and latitudes. Each record has the following form:  
     
    DDD MM SS W|E DD MM SS N|S  
     
   For example:  
     
    075 45 03 W 45 23 05 N\n  
   or  
    075 45 03 w 45 23 05 n\n  
     
   Create a DmsConverter.py module. Add a function called getEW that has one parameter called dmsRecord that will accept a record like the one above. This function will return an uppercase “W” or “E” depending on what is embedded in the record. That is, w, W, e, or E.
2. Add a function called getNS to DmsConverter.py that has one parameter called dmsRecord that will accept a record like the one above. This function will return an uppercase “N” or “S” depending on what is embedded in the record. That is, n, N, s, or S.
3. The dms format shown in this example record is useless in a GIS. Add a function to DmsConverter.py called dms2dd that takes this record as input (via a parameter) and returns a decimal degree, comma separated pair of coordinates (longitude, latitude). If the cardinal direction character is W or S, then the value of the coordinate should be negative otherwise it should be positive. Use the getEW and getNS functions you coded above. There is a valid domain of values for degrees, minutes, and seconds (e.g. 0 to 180 E/W or 0 to 90 N/S) for degrees and 0 to 59 for minutes and seconds). Your code should check that the numbers are valid. In test\_dms2dd, create a number of test calls with different values to make sure your code works for all cases. If any of the values are not valid, have the function return None.