Assignment #3 GEOG 567 – Introduction to Programming in GIS

Due Date: Tuesday, 21 October @ 11:00 a.m.

In this lab assignment, we will be working with Python to create a user-defined function to convert coordinate values between degree-minute-second and decimal-degrees formats and to create a script that will utilize the conversions using data stored in text files. Your task is to create a Python script that will be to perform the following:

1. Create a user-defined function called DMStoDD to convert degree-minute-second format coordinates to decimal degrees. The function should take three arguments (degrees, minutes, seconds numeric values) and return a single item: a numeric value representing the equivalent decimal degrees.

2. Create a Python script that utilizes your function above. The script will be capable of opening a comma-delimited text file (see sample file **dmscoords.txt** attached) and reading the file for input. Store your script and user-defined function in the same file.

3. Prompt the user to enter in the pathname to a text file storing the coordinates to be converted, and ask the user where to store the output file (name the output file **ddcoords.txt**).

4. Read all lines of data from the input file, perform the conversion using your user-defined function, and store the output in DD format in the output file.

Programming Tips:

The readlines() method is a convenient way to input all the lines of data from the coordinates file. Note that the first line is a header line, and not data. When you create the output file with DD coordinates, please also output a header line as the first line in the file as Latitude, Longitude.

To obtain and process the coordinates stored within the text file, I suggest you use the **readlines()** method I discussed in lecture to input the information stored in the file. Once you have obtained the file contents with the readlines() method, a good approach would be to set up a loop to obtain the coordinate information one line at a time. Think hard about whether you need to use count-controlled or event-controlled loop! Inside the body of each loop, you will need to convert each line of information into a sequence (e.g., list or tuple) of individual values that you can work with. A handy function for breaking-apart one long string of values separated by commas or whitespaces (i.e., space characters or tabs) is the **split()** method, which creates a python list of values automatically from a row of information (e.g., series of pixel values or strings of text).

Below is an example to illustrate how the split() method works to parse values from a line of text stored as a string:

```
>>> myline = 'aaa,bbb,ccc'
>>> myvalues = myline.split(',')
>>> print myvalues
['aaa', 'bbb', 'ccc']
```

In the example above, the string variable **myline** stores a line of text containing three values (aaa, bbb, and ccc) all separated by commas. Using the split() method (executed by using the statement *<string_variable_name>*.split(), a Python list object is returned containing the individual values from that string. If the items in the string are separated by spaces or tabs (i.e. whitespaces), no arguments to the split method are required. However, to work with comma-delimited strings, you must provide the delimiter character (i.e., a comma character) as an argument. Also note that this method always returns a list of individual string values, so if you want to work with numbers, you will also have to perform a conversion from string to numeric values.

Further reading:

If you want more information on using the split() method or processing data within strings, see Section 4.12 Working with Strings (particularly page 72) in the textbook.

Submission:

Remember to follow the program development cycle when you create your script and to provide clear internal documentation when you develop your code. Your submission for this assignment will be similar in format to Assignment #2:

- Steps 1 and 2 of the Program Development Cycle: similar to lab assignment #2
- Step 3: submit your Python code (i.e. your .py script)
- Steps 4 and 5 of the Program Development Cycle: similar to lab assignment #2