**Final Project Summation**

Well, the end of the GIS Customization course has finally arrived and my final project has been completed. At times, it felt like my code would never run to completion. I can’t count how many revisions I made to my code to get it to complete without exceptions and produce the proper dataset. To be honest, I was more nervous taking this class due to my novice programming and ModelBuilder skills than any other course I have taken in the GIT program thus far. This class was incredibly enlightening and I feel I am leaving the class with the proper knowledge and skill set that will allow me to create scripts and tools to automate numerous geoprocessing workflows. Eventually, I plan on enhancing what I have learned thus far and continue investigating how to create scripts in “Pure Python”. Eventually, I also plan on experimenting with my Raspberry PI and how it can be manipulated with Python scripts. With that said, it is time to discuss some takeaways that resulted from my time in the course and the trials and tribulations of developing my Python script tool.

**What Worked:**

Well, in the end, my script worked without exceptions, utilizing user input and produced useful spatial datasets as I had planned. My script correctly created postal feature classes consisting of centroids and polygons and organized them into appropriate feature datasets. This was the goal I set out to achieve and I was actually able to accomplish it.

**Iteration Worked**

In order to process my data and place the results in the appropriate feature datasets, my script was heavily dependent on iteration. Initially, I began my final project in ModelBuiler and it was extremely easy to use an iterator in this manner. The problem was that a ModelBuilder iterator will not convert correctly into a python script. As a result, I was concerned as to how I would be able to implement iteration in python. After much study, I was able to gain a firm grasp on iterating in Python using SearchCursors and ListFeatureClasses functions. Once I had that figured out, it was easy for my script to iterate numerous times to process the data and place it in the appropriate feature datasets. I have to admit, I much prefer iteration in python over ModelBuilder because nesting iterators in ModelBuilder is more cumbersome and less efficient than in python.

**The Script Tool utilizing the GetParametersFromText () variables worked.**

Creating a Script Tool utilizing the GetParametersFromText() variables was crucial for the script to be truly useful for myself or other users of the script. I really wanted the user of the script to be able to select where they wanted to create the personal geodatabase and provide a name for the database. In addition, I did not know the location where a particular user would download and subsequently load the input data from. The script tool makes it easy for users to specify input and output locations for the database and data. In addition, the GetParametersFromText() variables allow the script to not be dependent on hard-coded paths which rarely work correctly when the script is run on different computers.

**What Didn’t Work:**

Overall my script worked as originally envisioned and designed. However, there were a couple of enhancements that I wished to add to my script tool that I was unable to figure out. The first enhancement that I wished to add was to allow a user to be able to create a new folder location in the filesystem to place the Personal Geodatabase. Essentially I did not want the user to be required to specify a preexisting file system folder to save the database. I also did not want the user to have to create a folder manually to store the database and then select the new folder in the script tool.

Essentially, I wanted the user to perform everything from the script tool. To attempt to solve the problem I made a model in ModelBuilder using the CreateFolder\_management tool with parameters and converted it into a python script. When I attempted to create the folder in a script tool, the variable would never pass through the code and I would receive an exception. After numerous iterations and testing, I decided that the user was just going to have to select a preexisting folder location. I figured it was only a minor inconvenience for the user but not as elegant as I really wanted.

Probably the most important enhancement that I was unable to get to work correctly was in-memory processing. I was never able to successfully implement this processing feature. This was a disappointment because I knew it had the possibility to significantly decrease my script processing times. This was important because when I process the entire United States dataset, it can take many hours to complete. Eventually, I needed to make progress with my script and decided to bail on in-memory processing. However, my script is already is performing faster than the ModelBuilder version. So I am fairly happy with the script at this point. Upon completion of this class, I will double back around and attempt to get my script to create folders and implement in-memory processing.

**Frustrations:**

Regarding frustrations experienced using Python and the ArcPy Module, I decided to create a list of my Top Ten frustrations.

**Top Ten List of Frustrations:**

**10. Forgetting to place + "\\" at the end of workspace path.**

Numerous times I would get erroneous results, or my script would not run because I forgot to end my paths correctly.

**9. Not saving my script to a new version after I was able to get some code to function properly.**

Many times, while coding I would spend considerable time figuring something out. Then instead of saving a new copy of the script, I would keep working. Inevitably, I would break something as I continued on. Then it became difficult to figure out where I broke the script. Needless to say, I eventually started to save my scripts every time something went right.

**8. Building paths and the 000732 Error.**

Many times, while I was coding, I would receive an exception with the 000732 Error. What I eventually figured out is that there was something wrong with the path I built, and the script could not find the correct data. Once I realized what the error meant, I was much more careful building paths or better able to figure out what the problem was.

**7. Not taking breaks and then quickly solving the problem after a break**

In many instances, while I was working I couldn’t quite figure out how to fix an exception or solve my next challenge, so I would just keep working through the problem. Many times, it was because I did not want to lose the thought process I was working on. Almost without fail, I was able to relatively quickly solve the problem after a break away from the code window.

**6. At times I did not take good notes and could not figure out what I was doing when I returned to the code.**

There were a few times that I was able to solve a problem but did not take any notes on how I accomplished the task. Then when I returned to the code later, I couldn’t quite figure out what I did to solve the problem. Then I would have to search the code or try to reread how I found the solution. Eventually, I began to take better notes and started verbosely commenting on my script.

**5. Forgetting to switch to the proper workspace for the operation I was performing**

There were numerous times when I forgot to change the environment in my script and would either process no data, the wrong data or get the 000732 Error because the proper data was not able to be located. I can’t count how many times I did this mistake

**4. The syntax for certain tools was confusing, especially when dealing with joins and SQL statements**

For some of the more complex operations, the syntax for setting the tool parameters was confusing. This was especially the case for SQL statements and join operations. To solve these problems and better understand the proper syntax, I would make a simple version in ModelBuilder and study the output python script. Going back a forth between ModelBuilder and python became annoying but ultimately proved to be the solution.

**3. The leading zeros for the New England State ZIP Codes**

The leading zeros for the New England State ZIP Codes have always been problematic for anyone using these ZIP Codes in a database or table. Getting the leading zero back onto a ZIP Code is an incredible nescience. Thank you, New England, for having leading zeros in your ZIP Codes!

**2. Typos obviously!**

Uggggh! Typos and case sensitivity in Python. It is incredibly frustrating attempting to debug code over one little typo. And it happened over and over again. As my code grew, cut and paste began to minimize the syntax and exception errors that were the result of typos. However, typos are just part of the game when writing code.

**1. An improper logic that takes hours to work though**

Just knowing how to approach a particular problem and develop the proper workflow of a tool was frustrating and took up a large amount of time. This was made even more difficult while attempting to learn python function like SearchCursors and GetParametersFromText(). However slow and methodical reasoning won the day. It burned up a lot of time though.

**Realizations:**

Due to the effort and work, I put into creating my final project script, I have come to the realization that I am a bonafide “ArcPy” Python programmer/scripter now. At the beginning of this class, I would never have thought I would feel that I have the skills and knowledge to make that statement. With the fundamental skills I have obtained as a result of taking this class, I now have a renewed interest in programming and plan to experiment much more with Python and Visual Studio.net. I have to say that I am really glad I took this course and the knowledge I gained will be invaluable in the future. I feel the programming and python scripting skills will be adventitious and beneficial for my career for the foreseeable future.

**Conclusion:**

Overall this was an excellent class and I enjoyed my time here. At first, I was apprehensive about taking this class due to my lack of programming skills. Those fears, however, were unfounded. My final project ended up becoming a useful script that allows me to be more productive at work as well as be able to be ported to other areas of interest that I have. I feel I have taken a great deal of knowledge way from this class and plan on continue becoming a better programmer an scripter in the future.

For instructions on running my script, please reference the **Postal Data Geoprocessing Script Tool Instructions** that I have provided as a separate document.