Over the course of the Spring 2019 semester, I have learned increasingly about programming languages. There are different classifications of programming languages and these languages grow and expand for many reasons. Some expansion and evolution are due to hardware architecture and the software. Therefore, with so many languages to choose from while programming it is important to choose the correct tools for each task at hand. All of which was applied to the term project of creating a speech-to-text programming language.

For our classes term project, with needing to create a speech-to-text programming language the classes decided to create SIMPL. SIMPL is to be a speech-to-text programming language that can be used by users speaking and giving preset vocal commands to perform the art of programming in the SIMPL language. SIMPL is to give focus and allow for people who are visually impaired or visually disabled and allow for them to be able to program easily. The language is also to give focus on the aspect of programming while mobile. Therefore, SIMPL could be used when a laptop is inconvenient or unavailable like while traveling by train, bus, or other form of transportation.

SIMPL was developed with Python, using CMUSphinx4 as the voice recognition back end. Python was chosen to be the developing language for SIMPL for multiple reasons. Python is considered to be a Hybrid programming language, between linear and procedural programming languages and event-driven or object-oriented languages. The class decided that an object oriented language would be beneficial for our project due to our experience and the manipulation of the data and program that we desired. In the process of deciding the back end language for SIMPL, the options were narrowed to Java, C++, and Python. Java and C++ are classified as object-oriented languages strictly. Python follows the basic principles of object-oriented programming through inheritance, encapsulation, and polymorphism. Inheritance is a process of using details from a new class without modifying an existing class (Programiz, *Python Object Oriented Programming*). Encapsulation is the hiding of the private details of a class form other objects and polymorphism being the concept of using common operations in different ways for different data input (Programiz, *Python Object Oriented Programming*). Python programs are also typically 3 to 5 times shorter than Java programs, the difference is attributed to Python’s built-in high-level data types and dynamic typing (Rossum, 1997). Python programs are also typically 5 to 10 times shorter than C++ programs (Rossum, 1997). Therefore, with the classes, knowledge, and practice with object-oriented languages along with the semester time constraint, as well as the classes collective curiosity to learn more about the Python Language, the class decided that Python would be the best language to use for the back-end development of SIMPL.

To begin the SIMPL language the class decided that string manipulation was to be our starting point. There are multiple reasons for this starting point. One being the fact that computers and humans do not speak the same language and do not understand one another. Therefore, the computer needs to process the user's speech into strings in order to then translate and provide input for the SIMPL language. Therefore, we decided that we would need three differing parts working as one to create the SIMPL language. The first being that we would need some way to translate what the user was speaking into a text file, that could then be used and broken apart to be translated in a way that the computer could understand. The second part being the parser that would take the dissected text strings from voice recognition. The parser would look for specific words or commands. The commands were generated by the entire class prior to the beginning of the programming process. Once a recognized command was encountered the parser would call the third component. The third component being the processor. The processor contained the actual commands that perform the action specified by the user. Thus, string manipulation was our starting point.

I personally faced some challenges with this project. I was in charge of creating and working on the parser for the SIMPL language. Before, starting this project I did not know a lot about Python or have much experience with using the Python language. At first, I struggled with a starting point because this parser was not just a simple conversion of a text file to strings or vice-versa. It took some instruction for me to understand how to start the parser, but after I received instruction I understood and continued on. I struggled with the semantics for a while like remembering to put colons in the correct places as well as the correct use of indentation. However, that was easily corrected. Another issue that I faced with the parser was the fact that some of the commands were not created in the processor when I was trying to call them in the parser. Therefore, I had to make up command names and make comments to ensure that the team members in charge of the processor knew what to name those functions when they did create them. I also struggled with some of the parameters and when to include them within calling certain functions. It took some instruction on multiple occasions to understand this concept and what parameters were needed when a certain function was called. Even with these difficulties I still enjoyed and succeeded.

I learned many things through this course as well as the SIMPL project. I learned more about the aspects and advantages that Object-Oriented programming has and brings into the programming realm. Also, before this course and the SIMPL project I had no idea all of the components that go into making a programming language, let alone the components that go into making a programming language that converts speech-to-text. I also learned more about the Python language and how to use it properly, as well as gaining more experience with using Python. I also realize how long it would take to perfect the SIMPL language and how working with a team is definitely a benefit. Working with a team on different components also helped to teach the importance of communication and the factor of accountability between the team members and their work. The initial decisions that the team made were also important and it would have been hard to complete the project without one another and our work put together collectively. I really appreciate the work put in by my teammates and all the work and learning they did. I know they had frustrations in their areas and their components, but they worked through them. Overall, this course helped me to learn a lot about Object-oriented programming languages, the Python language, and the aspects that go into creating a new programming language.

I have taken away a large amount of new knowledge and skill from the programming languages course this semester. There are no decisions that I feel that if I started over again I would change, other than maybe a different speech-to-text aspect for the front end of SIMPL, due to the frustrations that I saw my teammate go through with trying to get it to be operational. I believed the decisions made were good and worked well other than that one aspect. I am glad to have participated and completed the programming languages course this semester.

Work Cited

Programiz. “Python Object Oriented Programming.” *Python Object Oriented Programming*, www.programiz.com/python-programming/object-oriented-programming.

Rossum, Guido. “Comparing Python to Other Languages.” *Python.org*, 1997, www.python.org/doc/essays/comparisons/.