Python Evaluation

Riley Fiske – September 11th, 2023

***Preface***

My previous exposure to Python is quite limited, having only used it for about 3 days in a previous internship where I had nothing better to do than teach it to myself for a couple days for personal projects. Thus, I had some prior experience in syntax and how certain data structures worked.

***Readability***

Python’s minimalistic use of keywords and lack of data type defining makes the lines quite human readable. The special words used are all quite English based, such as import, def, for, while, and, or, etc. The syntax of Python is quite picky, meaning the indents are always indicative of where a statement lies in relation to another line, which makes it very readable at quick glance. The built in functions also have logical names, so there is very little guesswork required to figure out what methods such as print, copy, input, or clear mean.

***Writability***

The forced indentation of Python is a double-edged sword between readability and writability. While it makes it much easier to read and see what is going on, one extra space or improperly placed tab by the programmer or the IDE can throw incredibly frequent errors. Besides the specific syntax required, it’s a quite simply written language since it uses English words for many method names or other keywords as opposed to symbols or abbreviations that are not as intuitive in our languages. The one slightly annoying thing is the use of the += operator instead of ++ for simple variable incrementation, but that is a small nit-picky thing.

***Reliability***

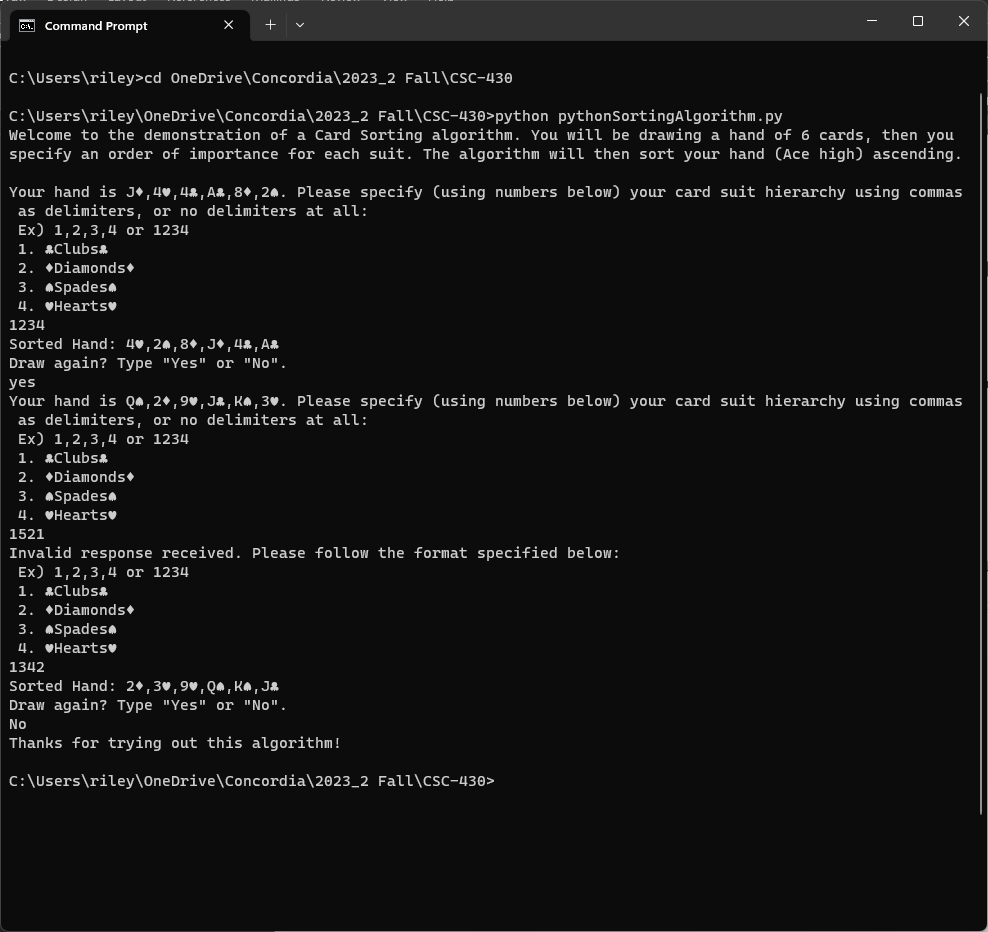
When I was writing my program and encountered errors, it always pointed me to the correct line and gave a good explanation as to what error halted the execution. I did not have to use any exception handling in my program, so I cannot speak to that. Since Python doesn’t require you to specify variable types, there can sometimes be errors in methods that are not caught by the IDE given the ambiguity of variable type being execution, but these errors are caught and explained well when they occur for fixing.

***Cost***

Many young programmers today begin or at least know Python given its use of spoken-language words, minimal use of brackets and elements that make the program less readable, and ease of execution. While I don’t believe Python is the end-all-be-all of programming languages, it does have very good use in professional statistical analysis and other small object-oriented programs, so it has utility and purpose for learning it.

***Quick Comparison to Java***

The biggest difference between Python and Java, in my mind, is the typed syntax. Java is all about inclosing everything in the main method and with the correct set of brackets to specify the end of loop, conditional, method declaration, or class. Python uses indentation to specify this, meaning there is much less flexibility on how you write things in Python. You could theoretically write a Java program in one line with effective bracket usage, but it would look horrible. Python forces this syntax to make it understand how to compile and execute your code. Another big difference is variable type defining. Python allows flexible variable typing while Java forces you to declare it and stick with it. This can lead to sloppier programming in Python and errors for using the wrong type in the wrong method, whereas Java sidesteps this problem by knowing the variable type before execution time and warning the programmer that a variable is used incorrectly. I am obviously much more comfortable coding in Java, but I did enjoy my time getting to explore Python more and get more comfortable with methods working with arrays and the syntax required for the extensive conditionals used in my program that making the syntaxing a strenuous challenge.

***Screenshots of Python Sorting Algorithm Execution:***