Christopher Ennis

James Vo

Keagan Haar

CSC 300

Language Design and Implementation

100

Read Me

YouTube: <https://www.youtube.com/watch?v=IryiMrNo1rQ>

Constants, Position, Tokens, Lexer, and ReadMe: Keagan Haar

The parser and interpreter put the project together and fixed any errors that arrived from meshing our parts together. Then, did the EBNF: James Vo.

BankAccount, Interpreter, Errors, and Tests: Christopher Ennis

Running the code:

To run our code, you first need to download our file. You can access our test file (Specification\_tests.py) from there and simply run it in your chosen IDE. This will go through the given specification tests we put in place and output whether they pass or fail and which tests were specifically running or not. This will give insight into our program and show that our code is running and works as intended.

Addendum

Grammar: We forgot to go through the attached grammar. It is basically defining the main digits, numbers, tokens, and keywords. The statements were, unfortunately, unused.

References:

Ho, T. (2020, July 18). *Make your own language 1: The lexer*. YouTube. <https://www.youtube.com/watch?v=Xu4RtLlm42I>

Klikovits, S. (2019, June 25). *I created a DSL in python and I love it!*. Modeling Languages. <https://modeling-languages.com/dsl-in-python-cyber-physical-systems/>

Schafer, C. (2017, August 16). *Python tutorial: Unit testing your code with the UNITTEST module*. YouTube. <https://youtu.be/6tNS--WetLI?si=0eCENgOvtmuhUGVP>

YouTube. (n.d.). *Make your own programming language in Python*. YouTube. <https://youtube.com/playlist?list=PLZQftyCk7_SdoVexSmwy_tBgs7P0b97yD&si=3ghq1NQJ9ZOiRtF6>