Vadym Kharchenko
CMSC 330
Prof. Richmond
October 3, 2024

Project 2

Approach

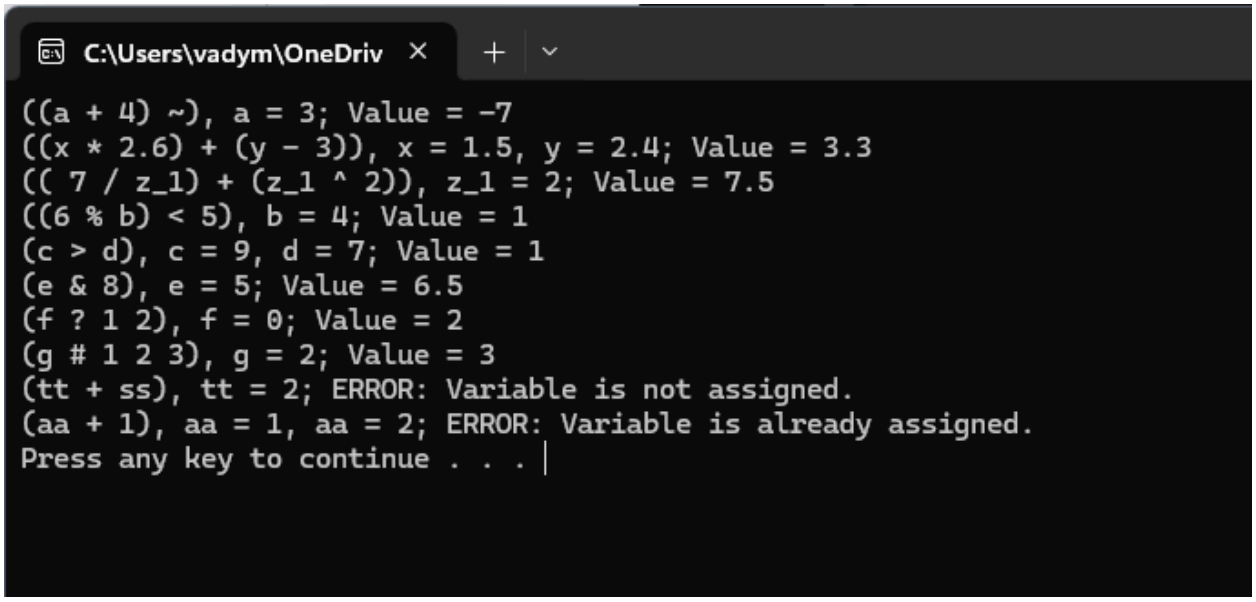
After starting this project and looking through the skeleton code and summary of grammar rules in this language of expression, it initially seemed like a manageable task. It wasn't as easy as I had assumed, as I soon discovered. It took longer than I had hoped to download Visual Studio and all of the C++ dependencies. Apart from setting up the environment, implementing the new functions was not too difficult; the proper formatting of the input presented a challenge. I had to pay close attention to each and every space, paren, and comma. My program would not display any errors if I had missed any of these. Aside from this little inconvenience, I liked working with conditional statements by handling the ternary and quaternary operators.

I was guided by examples, such as the plus and minus functions. Without them as a baseline for how the program was meant to handle various operations, I'm not sure I could write everything from scratch. I modified the parser to support additional operators, such as multiplication, negations (~), and some user-defined operators like average (&), maximum (>), and minimum (<). The program would handle more complex input scenarios if the variable token was modified to include underscores and the literals were changed to accept floating-point values.

Test Plan

Test Purpose	**Input**	**Expected Output**	**Actual Output**
Unary expression	((a + 4) ~), a = 3;	7	Correct
Binary expression	((x * 2.6) + (y - 3)), x = 1.5, y = 2.4;	3.3	Correct
Ternary expression	(f ? 1 2), f = 0;	2	Correct
Quaternary expression	(g # 1 2 3), g = 2;	3	Correct
Variable with underscore	((7 / z_1) + (z_1 ^ 2)), z_1 = 2;	7.5	Correct
Uninitialized variable	(tt + ss), tt = 2;	Error for ss	Error message

Duplicate assignment	(aa + 1), aa = 1, aa = 2;	Error for reassigning aa	Error message



Lessons Learned

After I struggled with the setup, it started to make a little more sense. Once I got past that, the logic just flowed. In this project, I have gained a lot about handling recursive parsers and symbol tables. Working with custom operators was enjoyable and somewhat challenging, especially since they weren't your typical relational operators. In this project, I would enhance the error-processing engine to generate more detailed feedback. Overall, this project was a successful learning experience, despite certain challenges encountered.