Daniel Frey
CS 3160-001
Optional Project
12/7/18

Problem Description: Evaluate a mathematical expression given in postfix notation. Postfix notation is when the operators follow the operands. This notation does not need any parentheses as the operators have a fixed number of operands.

Solution Approach: I approached this problem in steps. I knew I wanted to use a stack-like data structure to solve a postfix notation expression. A stack is generally used to implement a solution to this problem. Since Python does not have a stack data structure, I found that a list data structure would perform similarly.

The main idea for the solution was to correctly process an expression string. The input string is trimmed of excess whitespace so that there is only a single space separating each item in the string. The input string is then split into tokens according to the spacing. The program scans each token and checks if it is an operand or an operator. Operands are pushed onto the stack. When an operator is encountered, the top two values on the stack are popped and the operator action is applied to the values. Different cases were built to handle different operators, which are matched using a regular expression. It is possible to add in support for more operators.

Once the main skeleton was built, I wanted to add error handling, such as non-digits, invalid operators, invalid expressions, etc. I wanted to allow floating-point numbers, so I decided to make my own isDigit() function because the built-in isdigit() would only match integer numbers. To allow cases where decimal numbers were used, I treated every number as a floating-point number instead of as integers.

Language Chosen: I chose Python to solve this problem. Python was easily understood, which is one of the main reasons for choosing it. Python has a structure that is very easy to read as well. Furthermore, Python has language similarities to those that I already know, like C++. The built-in structures and function in Python also helped to solve the problem. I am comfortable using object-oriented programming languages.

Difficulties and Lessons Learned: The main issue I had with using Python was that I was new to Python and was not too familiar with the exact syntax of the language. After looking up the correct syntax, the rest of the solution was not too difficult.

Another issue I faced was with the PyCharm IDE. When trying to run the program for testing and debugging, the output had some strange characters in the output. Also, some of the user prompts would repeat twice. Initially, I thought the double output was an issue with how my code was working. In reality, it seems to be a common issue with the IDE, so I just used the console where everything went smoothly.