Marlon Aquino

Professor Dominick Atanasio

CS 2400.03 – Data Structures and Advanced Programming

Project 1

September 28th 2018

For this project, I was required to create two classes, one of ArrayStack and one of LinkedStack, and implement the StackInterface in those classes. I then wrote a class called InfixToPostfixConveter that implements ExpressionConverterInterface which was provided to us. InfixToPostfixConverter converts an infix expression that is inputted by the user, to a postfix expression using an algorithm. I created a final class called InvalidExpressionException which was thrown in the ExpressionConverterInterface methods.

In order to test my code, I created a main method that imported the Scanner class in order to read input from the user. Since the project required us to use two constructors, I tested the default one first which initializes the stack to a new LinkedStack. After seeing that work, I tested it on the other constructor with parameters, which initializes it to a new ArrayStack. After seeing that work, I then tested to see any instances where the InvalidExpressionException is thrown such as no input and invalid characters.

I learned a lot from this project. I initially did not know that there was an order of precedence that was taken just like PEMDAS. I also learned more about how stacks work from this assignment. One of the most difficult aspects that I had to encounter was making sure that numbers were also included and not just characters. To solve this, I created a private method that checks to see if the next character is a number and if it is, add that to the postfix expression.