Project #3
Stacks
CS 240

Philip Sloan
July 27th, 2014
prsloan@csupomona.edu

Project Specifications:

This project is designed to implement the stack abstract data structure. Using stacks, a program had to be constructed that could take an infix expression and convert it both to a postfix and a prefix expression following the appropriate rules of those formats. In addition, the program needed to have error checking to prevent empty stack exceptions, or format exceptions that may arise due to erroneous user input.

Testing Methodology:

I tested the program with a variety of cases. I made sure that there was at least 3 incorrect input cases. One with a missing right parentheses, one with a missing left parentheses, and another with a right parentheses as the first term in the expression. The program successfully caught the exceptions and prompted the user for a new input. I also tested several different expressions with a variety of terms in order to test the robustness of the algorithms used in the conversion methods. As long as I formatted the expression correctly, each input I gave evaluated exactly the same as the demonstration applet.

Lessons Learned:

The algorithm for the postfix conversion was simpler to wrap my head around for some reason. The prefix algorithm as written on the assignment gave me problems, and after some research I discovered an alternative algorithm that

worked much like the postfix algorithm, but iterating through the input in reverse. I still had kinks to work out in terms of the switched right and left parentheses and modified error checking, but eventually this worked.