Peter M Piper

This program uses binary trees to evaluate arithmetic expressions. It prints a diagram of the tree, the corresponding prefix, postfix, and infix expressions, and its value.

Data Structures 2

Arithmetic Expression Tree

**Reason for Test Case \_\_\_\_\_\_\_\_ Input Values\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Expected Output**

Enter in invalid command enter A You entered an invalid command!

Enter a valid command enter E prompts user for expression

Empty input select enter expression, Expression is invalid!

enter no value

Single digit select enter expression, 2

enter 2 The value of the expression is: 2

Infix: 2

Prefix: 2

Postfix: 2

Invalid character select enter expression, Expression has an invalid character!

enter a

Expression w\o parenthesis select enter expression, +

enter 3+2 3 2

The value of the expression is: 5

Infix: 3 + 2

Prefix: + 3 2

Postfix: 3 2 +

Expression - unbalanced parenthesis select enter expression, Expression is unbalanced!

enter (3+2

**Reason for Test Case \_\_\_\_\_\_\_\_ Input Values\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_Expected Output**

Expression – parenthesis select enter expression, +

enter (3+2) 3 2

The value of the expression is: 5

Infix: 3 + 2

Prefix: + 3 2

Postfix: 3 2 +

Evaluation -Division by zero select enter expression, Cannot evaluate, division by zero!

enter 3/0

Quit program select quit exits the program



