

**Due**

The due date is specified in the OAKS dropbox.

**OBJECTIVES**

- To implement a binary search tree using dynamic memory allocation.
- To use a Node class that has a parent pointer.

**PROGRAM REQUIREMENTS**

Write an expression evaluator using a binary expression tree.

**PROGRAM SPECIFICATION**

Write a program that will accept a simple mathematical expression in **prefix notation** from the keyboard. Parse the string into tokens that are either single digit operands or single digit operators. All tokens are separated by one comma. As the parsing is occurring, build the expression tree to hold the expression. Once the expression is in the expression tree, print the tree on the console in preorder, inorder and postorder forms, then output the value of the expression. In the inorder printout, make sure that the expression is fully parenthesized.

**Sample I/O:**

```
Enter an expression: +,10,*,2,4
Preorder: + 10 * 2 4
Inorder: (10 + (2 * 4))
Postorder: 10 2 4 * +
Evaluates to 18
```

**Simplifications:**

binary operators: +, -, \*, /

operands: a positive literal integer

The program is not expected to detect an error in the input. Assume that the expression in prefix notation is correct.

**Abstract Data Type:**

Binary Tree ADT implemented on the Node data structure that implements the BinaryTreeNode interface.

**REFERENCES**

None

**GUI:**

No GUI. Use the console for I/O.

**PROGRAM DOCUMENTATION**

Provide internal documentation only as required in the program documentation standard in OAKS.

Updates and clarifications to this assignment, if needed, will be done on Discussions.

**PROGRAM SUBMISSION**

Use the corresponding dropbox in OAKS using the same naming conventions as given in Program 1.