Lab 8

Objectives:

- Sign up for the lab
- Learn how to evaluate a binary expression tree
- Use Connex to submit two files: *ExprTree.java* and *Lab8.java* at the end of the lab

Expression trees are binary trees used to represent algebraic expressions formed with binary operators. For example, 14 – 30 / 5 can be represented by the following expression tree (figure 8.1). In this lab, only four operations – "+", "-", "*", "/" are considered.
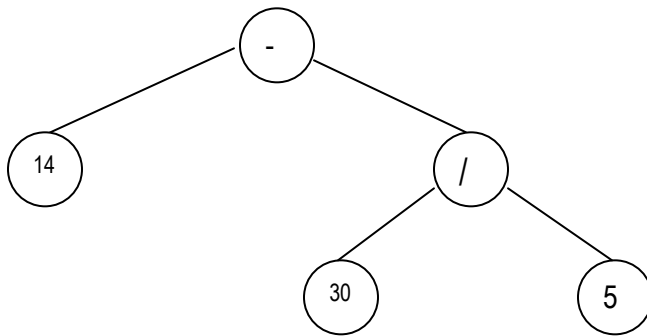
figure 8.1

Download InvalidExprException.java, ExprTree.java, and Lab8.java
An expression tree is built based on an array of strings (in post order), an exception is thrown if an invalid expression is encountered.

The tree data structure is not a linear structure. There are several ways to traverse a tree based on our needs. In this lab, we are going to revisit the concept of recursion to traverse a tree (inorder).

Implement the evaluate method in the **ExprTree.java** and in **Lab8.java**, create a tree and generate output like this "3 + 8 = 11.0".

Sample inputs and outputs:
>Java Lab 8 2 +
8 + 2 = 10.0

>Java Lab 8 2 "*"
8 * 2 = 16.0

>Java Lab 8 2 /
8 / 2 = 4.0

The algorithm for the evaluate method is:

      If (node is null or the item of the node is empty)

          Throw InvalidExprException

      If (node.item is a number)

          Convert it to a double and return it

      If (node.item is an operator)

      {

          If (either of the children of node is null)

             Throw InvalidExprException

          Get the value of the left sub-tree of node, set it to left

          Get the value of the right sub-tree of node, set it to right

          Depends on the value of node.item, do one of the following and return the result:

            Left + right or

            Left - right or

            Left * right or

            Left / right (need to check if right is 0, Throw InvalidExprException

               if it is 0)

          Throw InvalidExprException if none of the operators apply

      }

      Throw InvalidExprException if none of the operators apply

Task: implement ExprTree.java and Lab8.java. Test them thoroughly.

Use Connex to submit two files: *ExprTree.java* **and** *Lab8.java* **at the end of the lab.**