**CS340 P2 – Expression Tree**

In this assignment, you will complete the ExpTree class by adding the following functions:

1. Implement the makeExpTree functions that accepts an exp as a string argument and create expTree.
2. A bool function named eval that evaluates the expression tree and assign result to the reference argument. This evaluation should handle arithmetic expression of decimal numbers. Function heading:

bool eval (double & result);

function return the true if eval is successful, false otherwise. For this assignment, we will assume that the expression is grammatically correct. A straightforward implementation is to call a recursive version of the eval function that does all the work. It will have the following heading:

bool eval(node<string\* rt, double& result);

Obviously, you are free to define additional functions as necessary.

Sample I/O

Your program will continues to ask user for infix expression, create an exp tree, print the tree using indentation (see IP function), print the expression in postfix notation, and call eval to evaluate the expression and print the result.

# Required Data Structure

You MUST use an ExpTree class which derives from binary tree base class.

**Submission**

**This assignment is due Thursday (2/20) at 10 PM**. Name your expression tree class ExpTree.h and *ExpTree.cpp* and the driver file *main.cpp*, for grading simplicity. Submit the source code files on Moodle. Late submission will be accepted until 10 PM Saturday (2/22). Late penalties are 5 points per 12 hours until cutoff time.