# CS 400 Lab 5: Stack Applications: Convert Infix expression to Postfix expression.

**Note**: for CS 400’s lab works, you will need to

* work on the CS400Lab5\_spring2024\_name\_WSUID.cpp, and
* Submit your solution to the blackboard.

## Question:

Implement a function to convert infix expression (string input) to postfix expression (string output) by using the following algorithm:

You should formulate the conversion algorithm using the following six rules:

1. Scan the input string (infix notation) from left to right. One pass is sufficient.
2. If the next symbol scanned is an operand, it may be immediately appended to the postfix string.
3. If the next symbol is an operator,
   1. Pop and append to the postfix string every operator on the stack that
      1. is above the most recently scanned left parenthesis, and
      2. has precedence higher than or is a right-associative operator of equal precedence to that of the new operator symbol.
   2. Push the new operator onto the stack.
4. When a left parenthesis is seen, it must be pushed onto the stack.
5. When a right parenthesis is seen, all operators down to the most recently scanned left parenthesis must be popped and appended to the postfix string. Furthermore, this pair of parentheses must be discarded.
6. When the infix string is completely scanned, the stack may still contain some operators. [Why are there no parentheses on the stack at this point?] All the remaining operators should be popped and appended to the postfix string.