CS2313 Data Structures
Assignment 2: Stacks and Queues

1. Evaluating Postfix Notation (20 points)

Sketch of algorithm for evaluating postfix strings that you are implementing:

(1) Create stack s.
(2) For each token, x, in the postfix expression (these tokens will be passed to your function in a queue):
  1 If x is "true" or "false" push it into the stack s.
  2 Else if x is a unary operator
    i If you do not have at least one operand in s, you should return an error in the boolean.
    ii pop an operand, $op1$, from s
    iii compute x $op1$ (see unary table)
    iv push the result into s
  3 Else if x is a binary operator
    i If you do not have at least two operands in s, you should return an error in the boolean.
    ii pop an operand, $op2$, from s
    iii pop an operand, $op1$, from s
    iv compute $op1$ $op2$ x (see binary table)
    v push the result into s
(3) If s contains more than one operand after all of the tokens are evaluated you should return an error in the boolean.
(4) Otherwise pop and return the only value in s.

Operator Type	**Usage**	**Calculation**
unary operator	$op1$ NOT	$!op1$
binary operator	$op1$ $op2$ AND	$op1$ && $op2$
binary operator	$op1$ $op2$ NAND	$!(op1$ && $op2)$
binary operator	$op1$ $op2$ OR	$op1$ || $op2$
binary operator	$op1$ $op2$ NOR	$!(op1$ || $op2)$
binary operator	$op1$ $op2$ XOR	$op1$!= $op2$
binary operator	$op1$ $op2$ COND	$!op1$ || $op2$
binary operator	$op1$ $op2$ BICOND	$op1$ == $op2$

HINT: You can use *Boolean.valueOf(string s)* to convert a string to a boolean and *Boolean.toString(boolean b)* to convert a boolean to a String.

Deliverables: Your solution should be submitted in the provided file "ComputePostfixFormula.java". Also attach any additional files you create to solve this problem. Upload these files to Canvas under Assignment 2. **Do not zip your files**. To receive full credit, your code must compile and execute.