# 5. Stacks and Queues :: Example Application :: Evaluating Arithmetic Expressions

What would happen if the operators were reordered, i.e., how would we evaluate 3 * 4 - 5?

1. Create the operand and operator stacks. Both are empty at the beginning.
2. Scan the first operand (3) and push it onto the operand stack.
3. Scan the first operator (*) and push it onto the operator stack.
4. Scan the next operand (4) and push it onto the operand stack.
5. Scan the next operator (-). Since the operator on top of the operator stack (*) has higher precedence than -, evaluate the top.
   a. Pop the top number from the operand stack. Call this $right = 4$.
   b. Pop the top number from the operand stack. Call this $left = 3$.
   c. Pop the top operator from the operator stack. Call this $op = *$.
   d. Evaluate the operator and push the result (12) onto the operand stack.
   e. Now push the operator - onto the operator stack.
6. Scan the next operand (5) and push it onto the operand stack.
7. The end of the expression has been reached. Evaluate "the top".
   a. Pop the top number from the operand stack. Call this $right = 5$.
   b. Pop the top number from the operand stack. Call this $left = 12$.
   c. Pop the top operator from the operator stack. Call this $op = -$.
   d. Evaluate the operator and push the result (7) onto the operand stack.
8. Since the operand stack is empty, the result of evaluating the arithmetic expression is on top of the operand stack.

# 5. Stacks and Queues :: Example Application :: Evaluating Arithmetic Expressions