

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

Operators in arithmetic expressions have **precedence**, i.e., mathematically we define **precedence rules** that specify the order in which operations are performed in an expression involving multiple operators.

For the standard four arithmetic operators, the operator precedence rules are:

2. \* and / have higher precedence than + and - and are performed left to right.
1. + and - have lower precedence than \* and / and are performed left to right.

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

How would we evaluate  $3 - 4 * 5$ ? The steps are:

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 \* has higher precedence than the operator on the top of the operator stack, push \* 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* = 4.
  - c. Pop the top operator from the operator stack. Call this *op* = \*.
  - d. Evaluate the operator and push the result (20) onto the operand stack.
8. Since the operand stack is not empty, evaluate "the top".
  - a. Pop the top number from the operand stack. Call this *right* = 20.
  - 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 (-17) onto the operand stack.
9. Since the operand stack is empty, the result of evaluating the arithmetic expression is on top of the operand stack (-17).

## **4. Stacks and Queues :: Example Application :: Evaluating Arithmetic Expressions**