OPERATORS AND OPERANDS

- Operators are symbols or words that denote some processing that takes place on one, two, or three expressions.

- Operands are the expressions on which operators work; generally they can be:
  - constants
  - variables
  - functions that return values
  - combinations of the above

- Operators produce a new expression.
Operators can be characterized by the number of required operands:

- **Unary**: a single operand
- **Binary**: two operands
- **Ternary**: three operands

Examples:
- `-N`  
- `new Person`  
- `sizeof(int)`
- `a + b`  
- `y * sqrt(2)`  
- `x / 2`
- `(x < y) ? x : y`
ORDER OF OPERATOR EVALUATION

• When an expression contains multiple operators, two characteristics govern the order in which the operators are evaluated
  • Precedence
    • Arbitrary but generally follows algebraic conventions
    • Built into the compiler
  • Associativity
    • Arbitrary but generally makes good sense
    • Built into the compiler
**PRECEDENCE**

- *, /, and % all have the same precedence
- + and – have the same precedence, which is lower than the above
- = has a very low precedence
- \(a = 4 + 2 \times 3\)
  - \(2 \times 3\) is evaluated first
  - \(4 + 6\) is evaluated next
  - \(a = 10\) is the last operation
- Precedence can be overridden with parentheses
  - \(a = (4 + 2) \times 3\)
ASSOCIATIVITY

- Associativity is the direction of evaluation (left to right or right to left)
- *, /, %, +, and − are all left associative (evaluated left to right)
- = is right associative (evaluated right to left)
- \( a = 4 + 2 + 3 \)
  - 4 + 2 is evaluated first
  - 6 + 3 is evaluated next
  - \( a = 9 \) is evaluated next
- \( a = b = c = 0 \); is evaluated as \( a = (b = (c = 0)) \);
<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>Grouping</td>
<td>Right</td>
</tr>
<tr>
<td>!</td>
<td>Logical negation / not</td>
<td>Right</td>
</tr>
<tr>
<td>+, -</td>
<td>Unary + and -</td>
<td>Right</td>
</tr>
<tr>
<td>*, /, %</td>
<td>Multiplication, division, modular</td>
<td>Left</td>
</tr>
<tr>
<td>+, -</td>
<td>Addition, subtraction</td>
<td>Left</td>
</tr>
<tr>
<td>&lt;, &gt;, &lt;=, &gt;=</td>
<td>Less/greater than, less/greater than or equal to</td>
<td>Left</td>
</tr>
<tr>
<td>==, !=</td>
<td>Equal to, not equal to</td>
<td>Left</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Logical AND</td>
<td>Left</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>=</td>
<td>Assignment</td>
<td>Right</td>
</tr>
</tbody>
</table>