Arithmetic Operators
Today

- Arithmetic Operators & Expressions
  - sections 2.15 & 3.2
  - Computation
  - Precedence
  - Algebra vs C++
  - Exponents
Assigning **floats** to **ints**

- Look at the following situation.

```cpp
int intVariable;
intVariable = 42.7;
cout << intVariable;
```

- What do you think is the output?
Assigning *floats* to *ints*

- What is the output here?

```cpp
int intVariable;
double doubleVariable 78.9;
intVariable = doubleVariable;
cout << intVariable;
```
Arithmetic Operators

- Operators allow us to manipulate data
  - Unary: `operator operand`
  - Binary: `operand operator operand` (left hand side) (right hand side)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Negation</td>
<td>Unary</td>
<td>- 5</td>
</tr>
<tr>
<td>=</td>
<td>Assignment</td>
<td>Binary</td>
<td>rate = 0.05</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>Binary</td>
<td>cost * rate</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td>Binary</td>
<td>cost / 2</td>
</tr>
<tr>
<td>%</td>
<td>Modulus</td>
<td>Binary</td>
<td>cost % 2</td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td>Binary</td>
<td>cost + tax</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td>Binary</td>
<td>total - tax</td>
</tr>
</tbody>
</table>
Integer Division

- What is the output?
  - `int grade;`  
    `grade = 100 / 20;`  
    `cout << grade;`

- `int grade;`  
  `grade = 100 / 30;`  
  `cout << grade;`
Division

• \( \text{grade} = 100 \div 40; \) \text{ grade is 2}
  
  o If both operands of the division operator are integers, then integer division is performed.
    - the data type of grade is not considered, why?
  
  o We say the integer is \textit{truncated}. Everything after the decimal point is dropped. No rounding.

• \( \text{grade} = 100.0 \div 40; \)
  
  o \text{ grade} is 2.5
  
  o What data type should grade be declared as?
Modulus

- Modulus is the remainder after integer division

- \texttt{grade} = 100 \mod 20;
  - \texttt{grade} = \, ?

- \texttt{grade} = 100 \mod 30;
  - \texttt{grade} = \, ?

- \texttt{rem} = x \mod n;
  - What are the possible values for \texttt{rem}?
Q.1. What value is assigned to x?

a.  \( x = 8 + 3; \)
b.  \( x = 8 - 3; \)
c.  \( x = 8 \times 3; \)
d.  \( x = 8 \% 3; \)
e.  \( x = 8 / 3; \)
Mathematical Expressions

- Complex mathematical expressions are created by using multiple operators and grouping symbols
  - expression: programming statement that has value
    - `sum = 21 + 3;`
    - In these two examples, we assign the value of an expression to a variable
    - `number = 3;`
Examples

- \texttt{result = x;}
- \texttt{result = 4 + result;}
- \texttt{result = 15 / 3;}
- \texttt{result = 22 * number;}
- \texttt{result = a + b \% c;}
- \texttt{result = a + b + d / c - q + 42;}
- \texttt{cout \ll \"The value: \" \ll (sum / 2) \ll \endl;}

Operator Precedence

• `result = a + b + d;`

• `result = 12 + 6 / 3;`
  o `result = ?`

• Rules on how to evaluate an arithmetic expression
  o arithmetic expressions are evaluated left to right
  o when there are two operators, do them in order of precedence
Operator Precedence

<table>
<thead>
<tr>
<th>Precedence of Arithmetic Operators</th>
<th>(Highest to Lowest)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unary negation) -</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>%</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

If two operators have the same precedence, evaluate them from left to right as they appear in the expression
Q.2. Practice

a. $5 + 2 \times 3$

b. $10 / 2 - 1$

c. $3 + 12 \times 2 - 3$

d. $4 + 17 \% 3 + 9$

e. $6 - 2 \times 9 / 3 \times 4 - 9$
Summary

• Today we have looked at:
  o Arithmetic Operators & Expressions

• Next time we will:
  o Continue looking at mathematic operators

• Completed section 2.15 & started on section 3.2