C++ Arithmetic

Instructor: Chi Tse (Ricky)
When we write programs, we often have to do calculations in order to turn some input data into meaningful output information.

It could be something as simple as counting number of attempts a user has entered his or her password.

Or it could be data analysis of experiments performed in a research labs.

We need to be able to perform calculations!
Note: C++ does not directly support Exponent operation.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>%</td>
<td>Modulus (remainder)</td>
</tr>
<tr>
<td>( )</td>
<td>Parentheses</td>
</tr>
</tbody>
</table>
# Order of Operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Name</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>Parentheses</td>
<td>Operations in the parentheses have the highest precedence. Should be evaluated first.</td>
</tr>
<tr>
<td>^</td>
<td>Exponent</td>
<td>This operation is <strong>not</strong> supported in C++.</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>Have equal precedence. Evaluated from left to right.</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>Modulus</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>Addition</td>
<td>Have equal precedence. Evaluated from left to right.</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
<td></td>
</tr>
</tbody>
</table>
Finding the remainder of \( 5 / 2 \) \( 5 \mod 2 = \)

Find the remainder of the sum of two numbers divided by 2

A) \( \text{number1} + \text{number2} \mod 2 = \)

B) \( 2 \mod (\text{number1} + \text{number2}) = \)

C) \( (\text{number1} + \text{number2}) \mod 2 = \)

Is the answer A, B or C?
• Integer division in C++ will truncate any decimal value, for example:
  – $5 / 2 = 2$ (not 2.5) for an integer division
  – $4 / 2 = 2$
  – $10 / 3 = 3$
  – $11 / 2 = 5$

• The resulting type is an integer.
• What makes this an integer division?
  – **Both** the dividend and divisor are integers (not decimal).
Double division in C++ will retain the appropriate decimal value, for example:

- \( 5 / 2.0 = 2.5 \)
  - \( 4 / 2.0 = 2.0 \) // on the screen you see 2 printed
- \( 10.0 / 3 = 3.333333... \)
- \( 11.0 / 2.0 = 5.5 \)

- The resulting type is a **double**.

- What makes this a decimal division?
  - Either the dividend or divisor must be a decimal.
• Same rules apply to Addition, Subtraction and Multiplication.

• If both values are of type int, the result is int.

• If either is of type double, the result is double.
Look at the example below:

• 5.0 + 5 / 2
  — What is the resulting data type of the first operation?
  — What is the resulting data type of the second operation?

• 5 + 5 / 2.0
  — What is the resulting data type of the first operation?
  — What is the resulting data type of the second operation?
Look at the example below:

- **5.0 + 5 / 2**
  - What is the resulting data type of the first operation?
    - int
  - What is the resulting data type of the second operation?
    - double

- **5 + 5 / 2.0**
  - What is the resulting data type of the first operation?
    - double
  - What is the resulting data type of the second operation?
    - double
If we want to convert from an int to a double for a more precise result we can do so with type casting.

Type casting is a temporary change from one type to another.

To type cast from int to double we can do the following:

```java
double value = (double) 5 / 2;
//value = 2.5
```
• We can also type cast from double to an int, this will truncate the decimal value.

• Example:
  double total = 100.5;

  int value = (int) total;

  //value = 100