

Computer Science I: Lab 4

due Wednesday, March 16, 11:59pm – right before midnight

This document presents our first lab on functions.

For this lab, we will build a 5-function calculator, performing calculations only on **integers**. The user should be prompted to continually enter an equation in the form of $a + b$ or $a \times b$, etc. The calculator should support the following functions:

- Addition ($a + b$)
- Subtraction ($a - b$)
- Multiplication ($a \times b$)
- Division (a / b)
- Run sum calculation ($a \text{ R } b$) – *this calculation will be defined below!*

Each time a **valid** calculation has been entered, the results of the operation should be printed out also in the form of an equation (e.g., $3 \times 7 = 21$). The program should check for three invalid types of operations:

- Divide-by-zero – any number divided by 0 is invalid
- $a > b$ in the $a \text{ R } b$ operation – this operation is only defined if $a \leq b$

The program should stop execution and exit when a special sentinel calculation is entered - I suggest you use 0×0 .

In writing your program, you should define **at least** the following functions:

- `inputError(...)`
Takes in the two numbers and the operation character. Determines if the calculation is valid. Returns a number or letter indicating the type of error if the equation is invalid, otherwise returns another number or letter indicating there is no error. For example, `inputError` could return 'Z' for divide-by-zero error, and 'V' if the calculation is valid.
- `Add(...)`
Calculates the sum of the two inputs
- `Subtract(...)`
Calculates the difference of the two inputs
- `Multiply(...)`
Calculates the product of the two inputs
- `Divide(...)`
Calculates the integer quotient of the two inputs
- `RunSum(...)`
Calculates run sum between a and b for the two inputs – *see below!!*

For example, if you have declared and defined `Multiply` correctly, you can call it as follows:

```
int x=Multiply(40,12);
```

at which point `x` would have the value 52.

If you have declared and defined `Subtract` correctly, you can call it as follows:

```
int a=6, b=2, c;  
c = Subtract(a,b);
```

at which point `c` would have the value 4.

Note: It is your job to determine the input and output types for each function you write, making sure to fulfill the specifications written above.

An example execution of the calculator program is given at the end of this document.

Start early and look over my programming advice online!

Submitting your file:

Submit the final C++ code as `calculator.cpp` using `submit1600` (and verify proper submission using `verify1600`).

What is the Run Sum operation?

Run sum calculates the sum of the numbers between `a` and `b`:

$$a \text{ R } b = a + (a+1) + \dots + (b-1) + b$$

For example: **3 R 7** = 3+4+5+6+7 = 25

Example execution:

```
> ./calculator.out
```

```
Enter an equation or 0 x 0 to exit: 12 / 4  
12 / 4 = 3
```

```
Enter an equation or 0 x 0 to exit: 4 x 2  
4 x 2 = 8
```

```
Enter an equation or 0 x 0 to exit: 5 R 3  
Error -- 5 > 3 in 5 R 3
```

```
Enter an equation or 0 x 0 to exit: 3 R 6  
3 R 6 = 18
```

```
Enter an equation or 0 x 0 to exit: 13 / 0  
Error -- cannot divide by 0
```

```
Enter an equation or 0 x 0 to exit: 0 x 0
```

```
Thanks for using the calculator. Goodbye.
```

```
>
```