Combining Operations

- Given this statement, what will the computer do?

```
cout << 10 + 11 + 12 - 3;
```

Combining Operations

- Assume that the computer can only do one operation at a time
  - What is the sequence of operations the computer will perform?
  - How many reasonable sequences can you come up with?

```
cout << 10 + 11 + 12 - 3;
```

Combining Operations

- Assume that the computer can only do one operation at a time
  - What is the sequence of operations the computer will perform?
  - How many reasonable sequences can you come up with?

```
cout << 10 + 11 + 12 - 3;
```

1. Add 10 and 11
2. Then add 12
3. Then subtract 3
4. Then print it to the screen

- Why does it have to be in that order?

Expressions

- An operator with its arguments (also called *operands)* is called an *expression*

```
5 + 6
17 - 8
```

- Every expression *evaluates* to a value, a piece of data

`5 + 6` evaluates to `11`

`17 - 8` evaluates to `9`

- Syntax rule: an operand can be any expression that evaluates to the right type of data
- Necessary rule: an operator *must* evaluate it's operands before it can execute

Using Expressions

- I want to print to the screen
- Start with the right operator
  - Insertion (`<<`)
- Then give it the correct arguments
  - LHS is `cout`, meaning print to the screen (we'll get back to this)
  - RHS can be any piece of data that I want to print
    - E.g. `cout << 5;`
- *Before* the operator can print, it has to evaluate the RHS to see what it is supposed to print
    - It could be a number: `cout << 5;`
    - Or, an expression that evaluates to a number: `cout << 2 + 3;`