Application: Evaluating Expressions

Evaluating Expressions

- Evaluating expressions is much trickier than it might seem due to issues of precedence.
  - 1 + 3 * 5 – 7 = 9
  - 4 / 2 + 2 = 2
  - 17 % 6 % 3 = 2
- How do we evaluate an expression?

Evaluating Expressions

- Two separate concerns in evaluating expressions:
  - **Scanning** the string and breaking it apart into its constituent components (*tokens*).
  - **Parsing** the tokens to determine what expression is being encoded.
- The scanning job is taken care of for us by the `TokenScanner` class.
- How do we handle parsing?

The Shunting-Yard Algorithm

- Maintain a stack of operators and operands.
- For each token:
  - If it's a number, push it onto the operand stack.
  - If it's an operator:
    - Keep evaluating operands until the current operator has higher precedence than the most recent operator.
    - Push the operator onto the operator stack.
- Once all input is done, keep evaluating operators until no operators remain.
- The value on the operand stack is the overall result.

The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

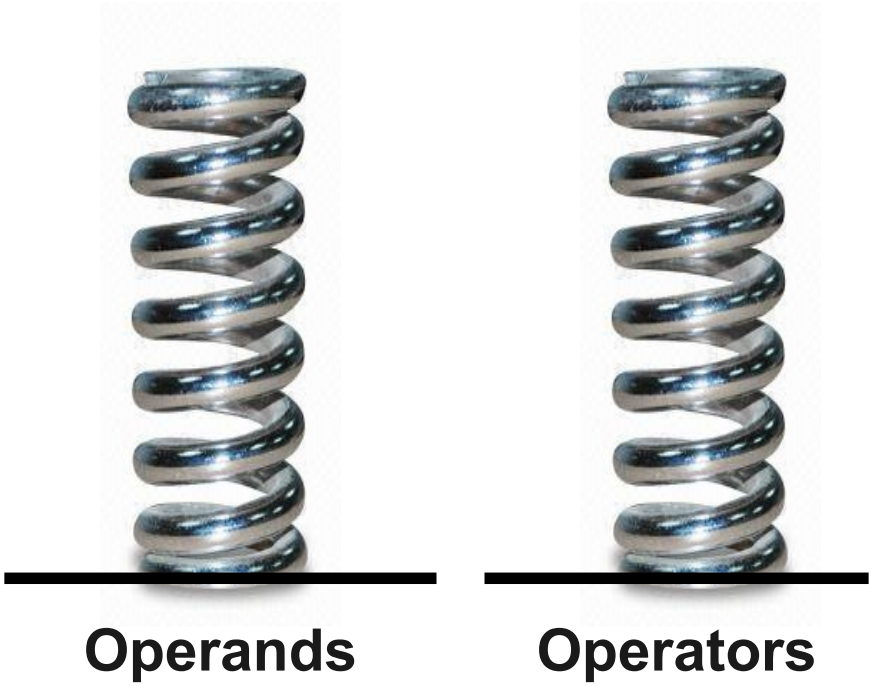
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

Operands

The Shunting-Yard Algorithm

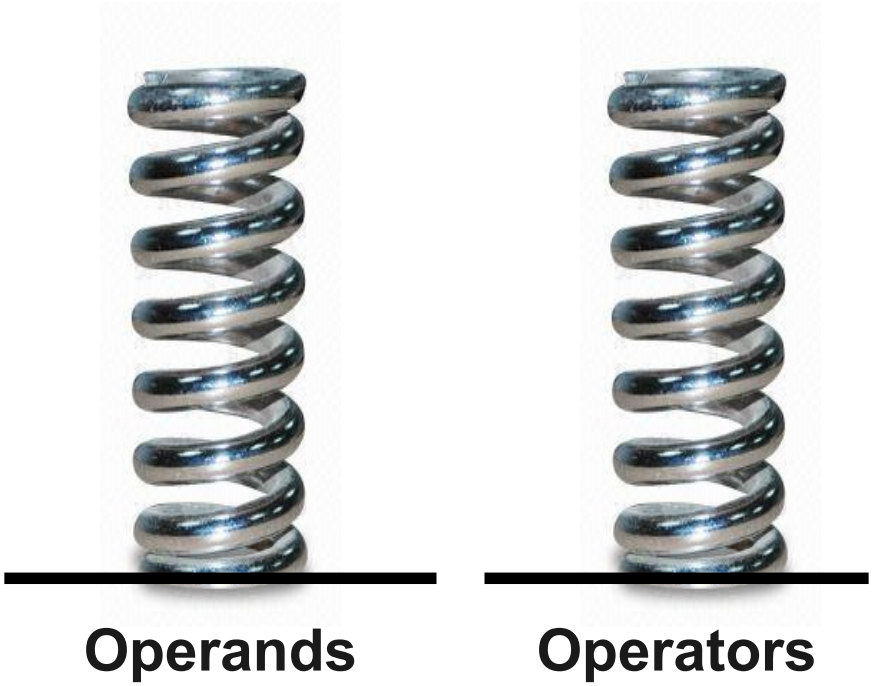
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

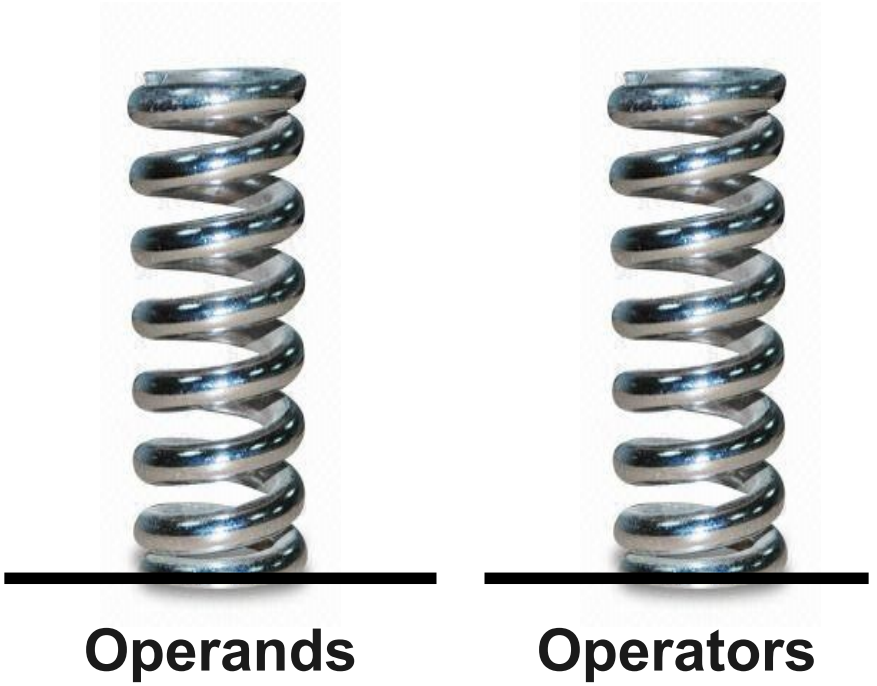
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

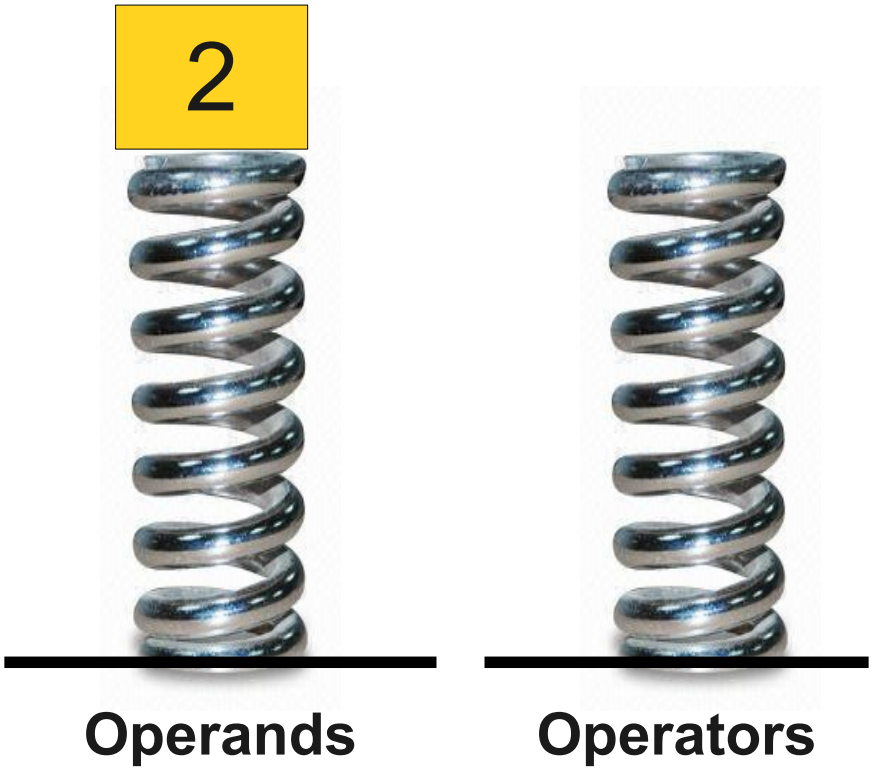
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

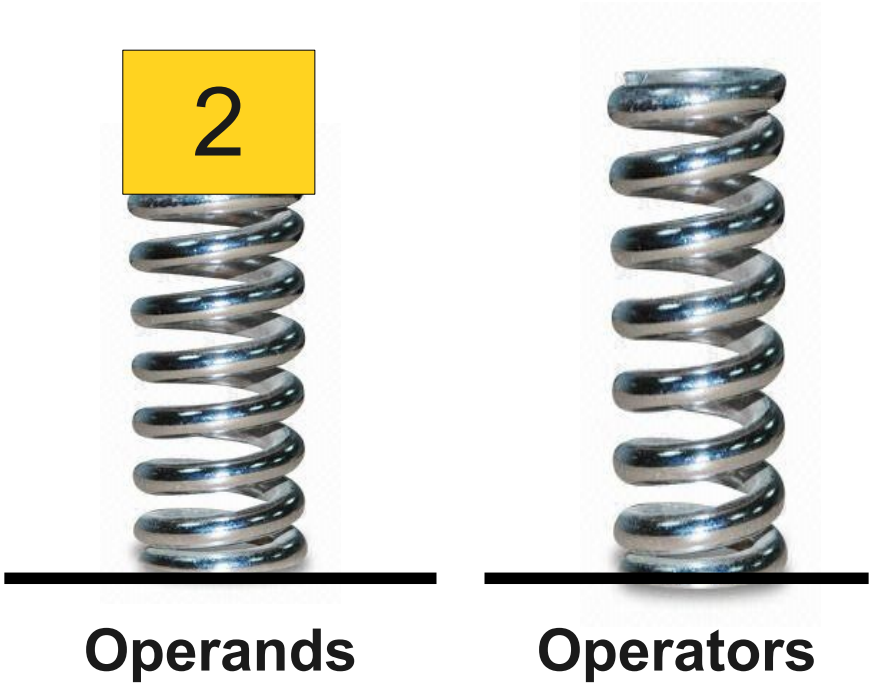
2	+	3	*	5	-	6	/	2

+	3	*	5	-	6	/	2



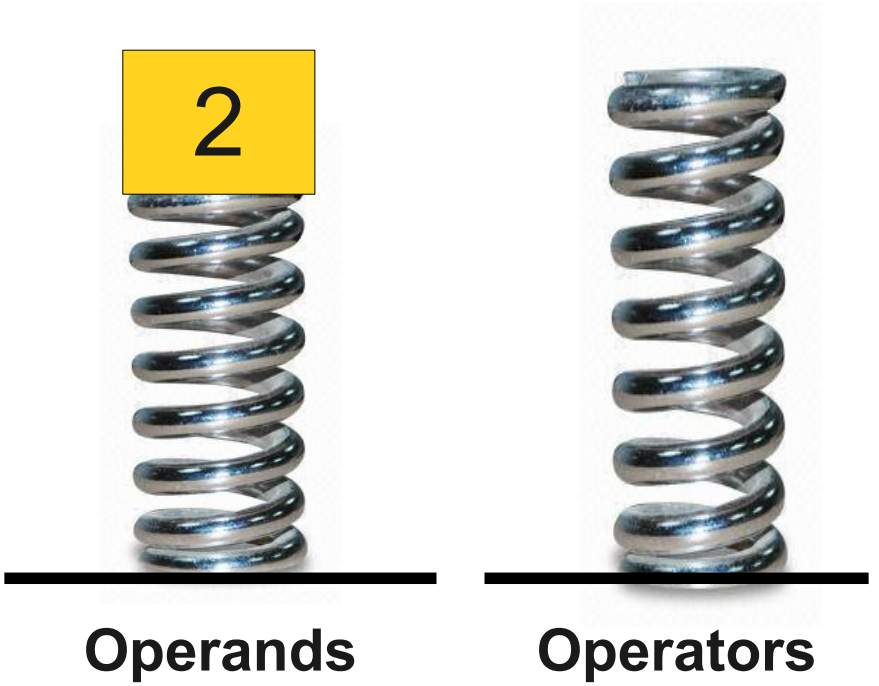
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2
	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

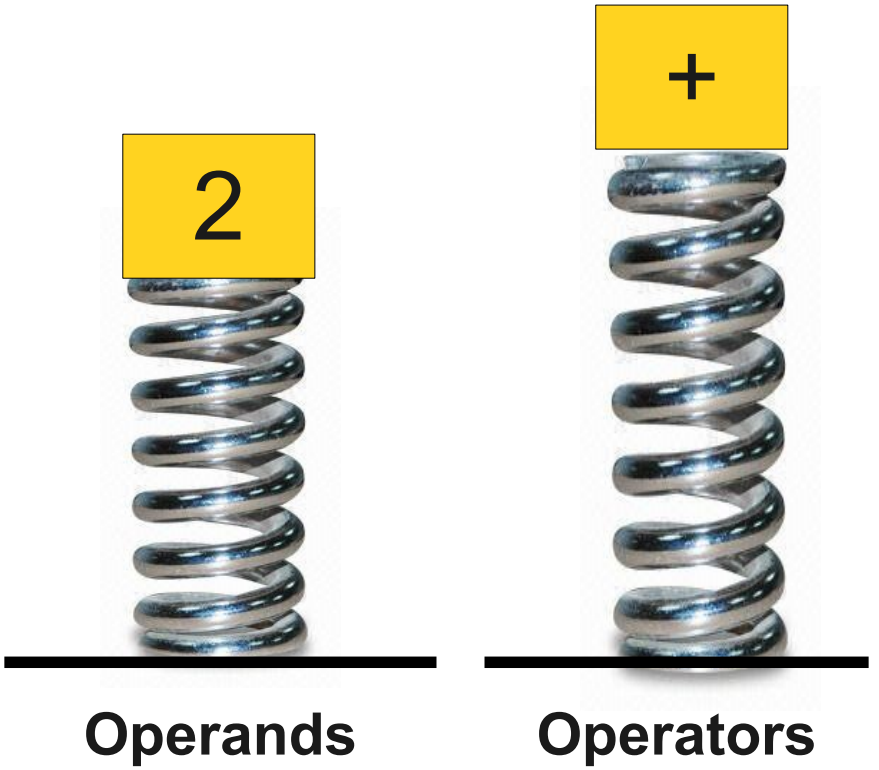
2	+	3	*	5	-	6	/	2
	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

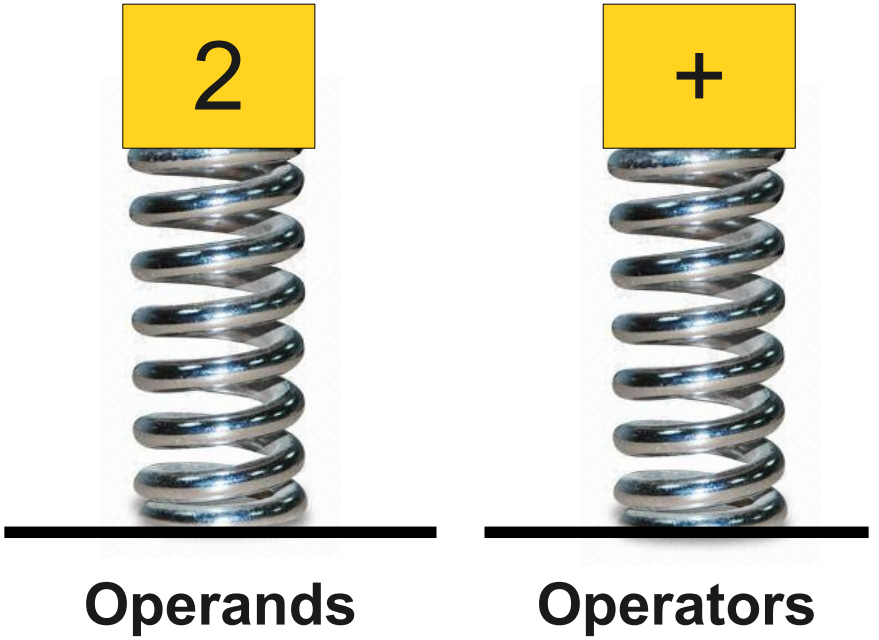
3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

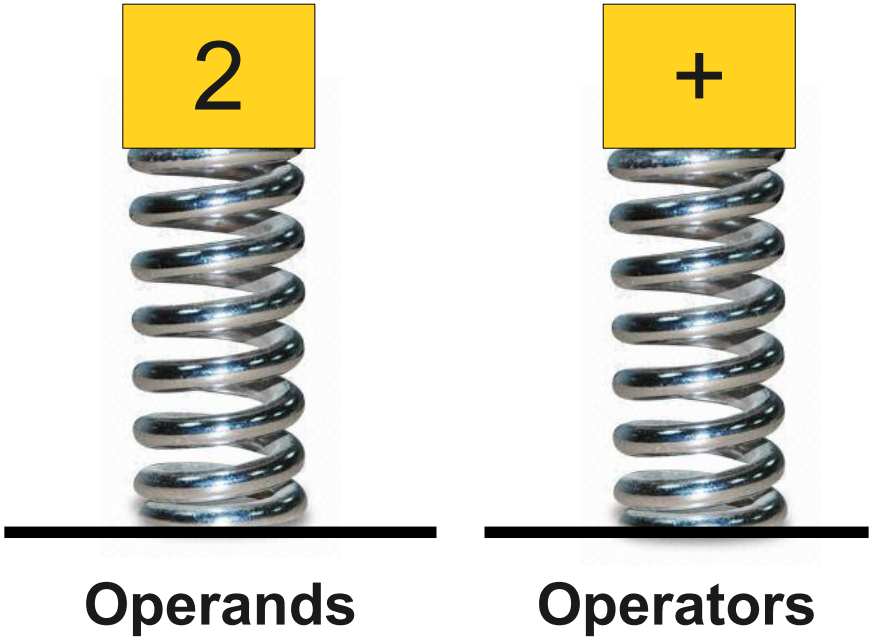
3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

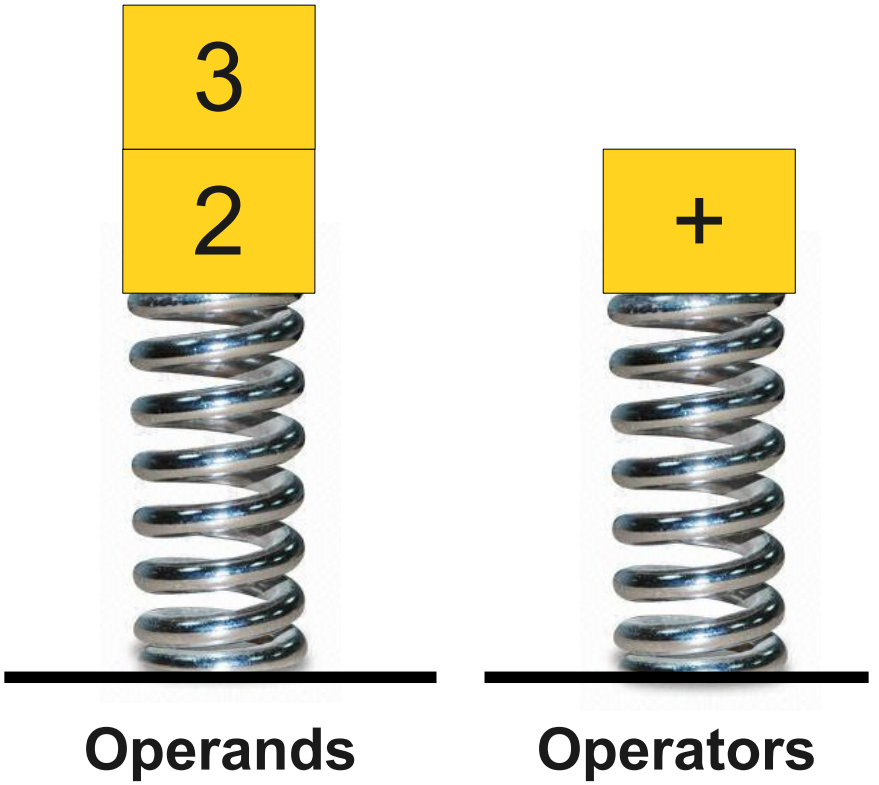
3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

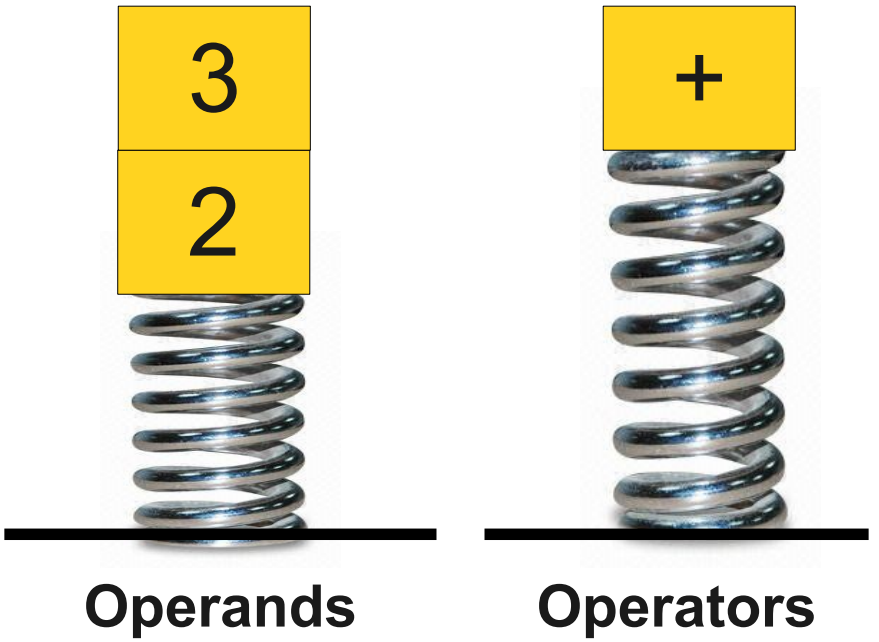
*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

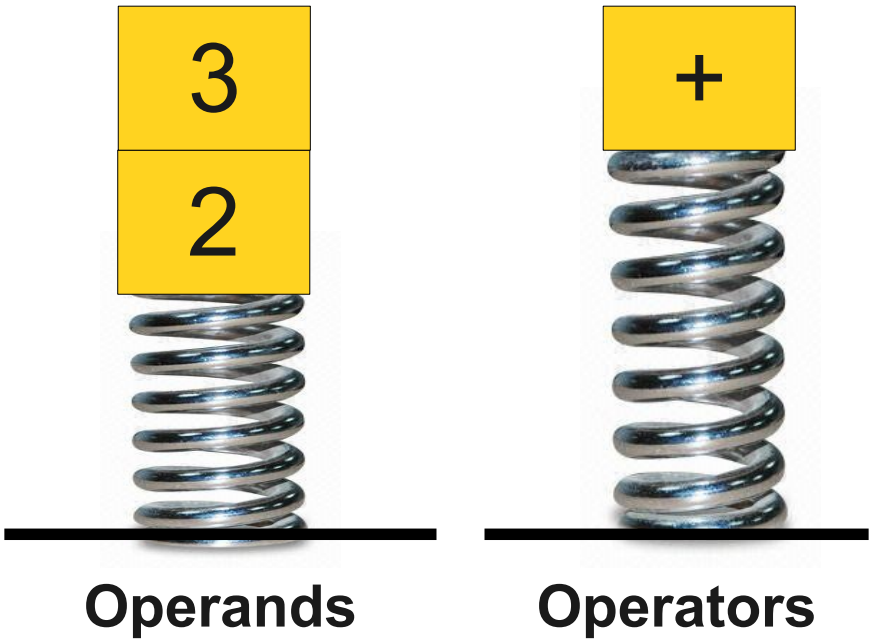
*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

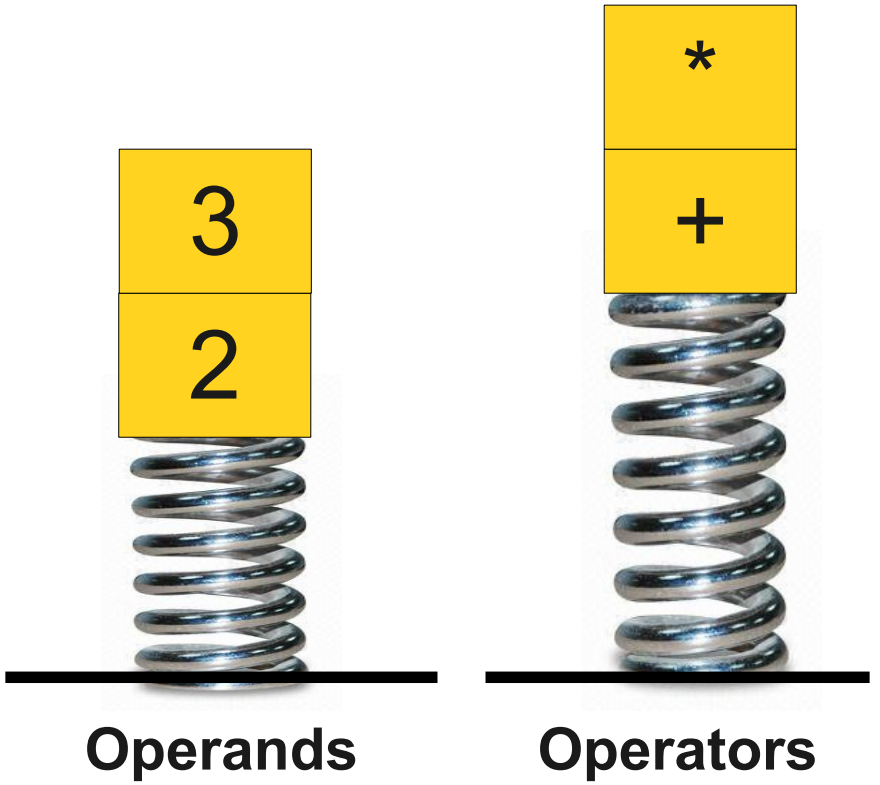
*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

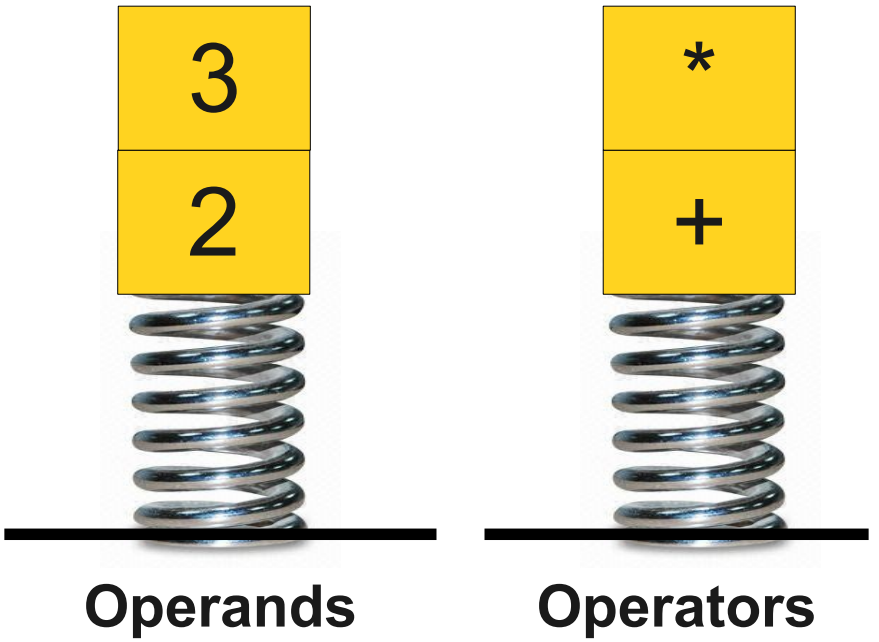
5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

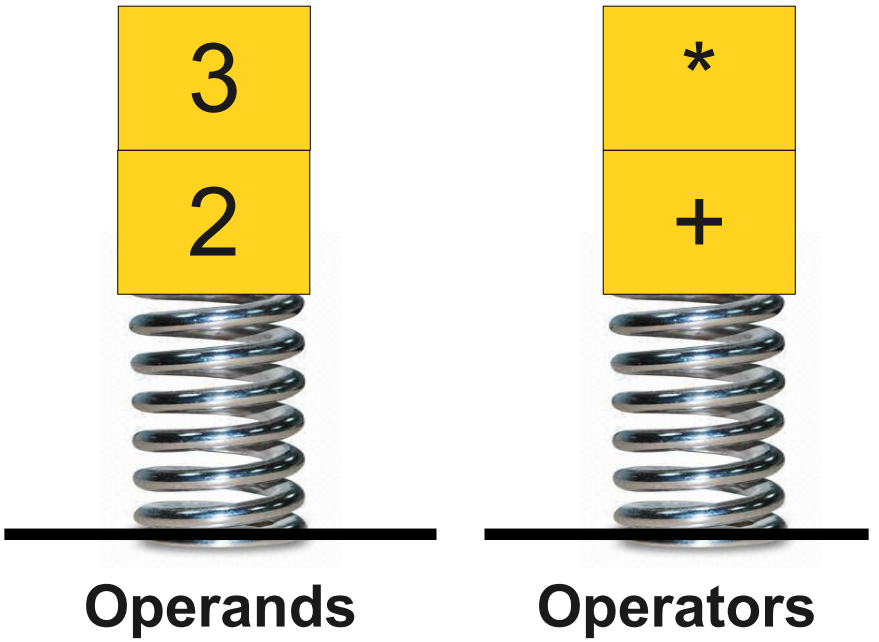
5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

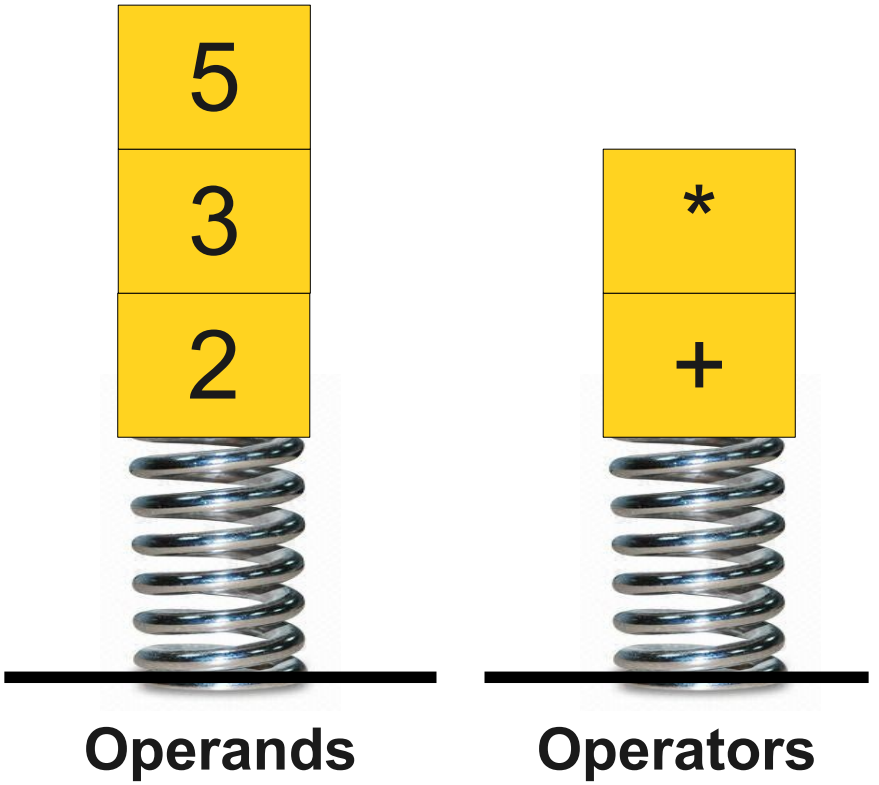
5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

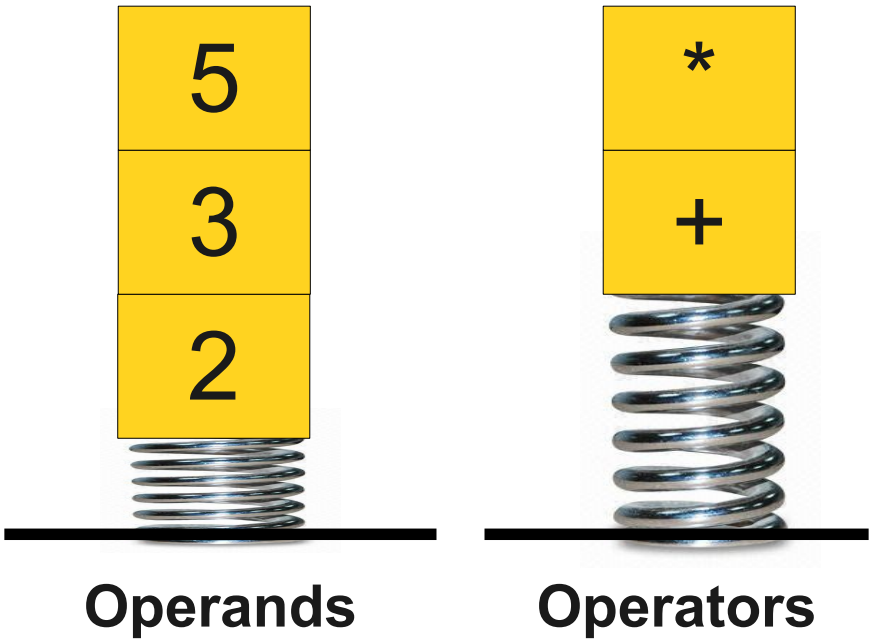
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

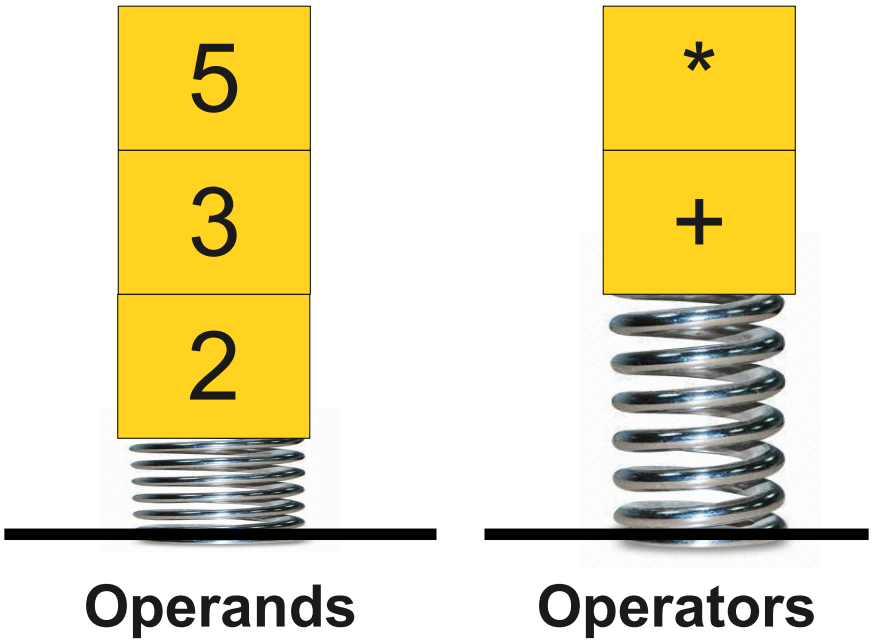
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

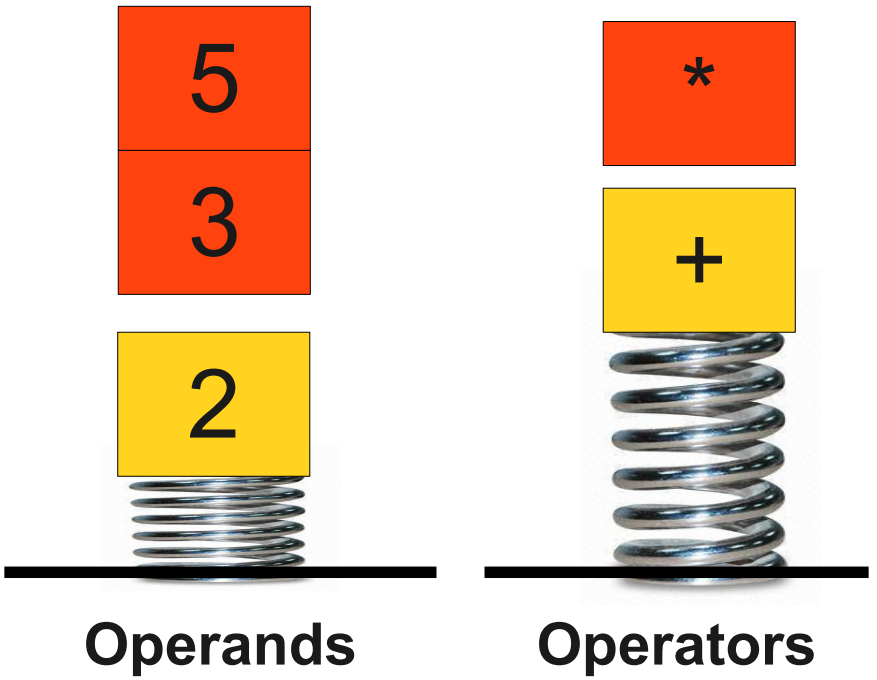
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

-	6	/	2

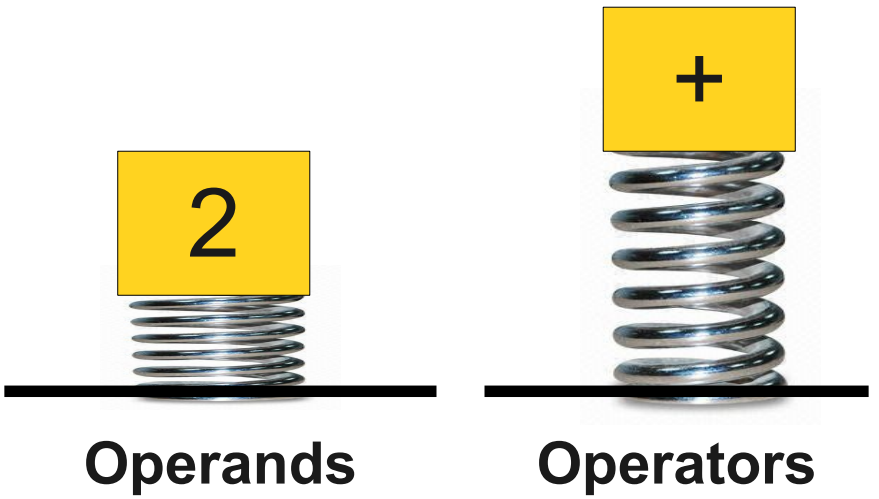


The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

-	6	/	2

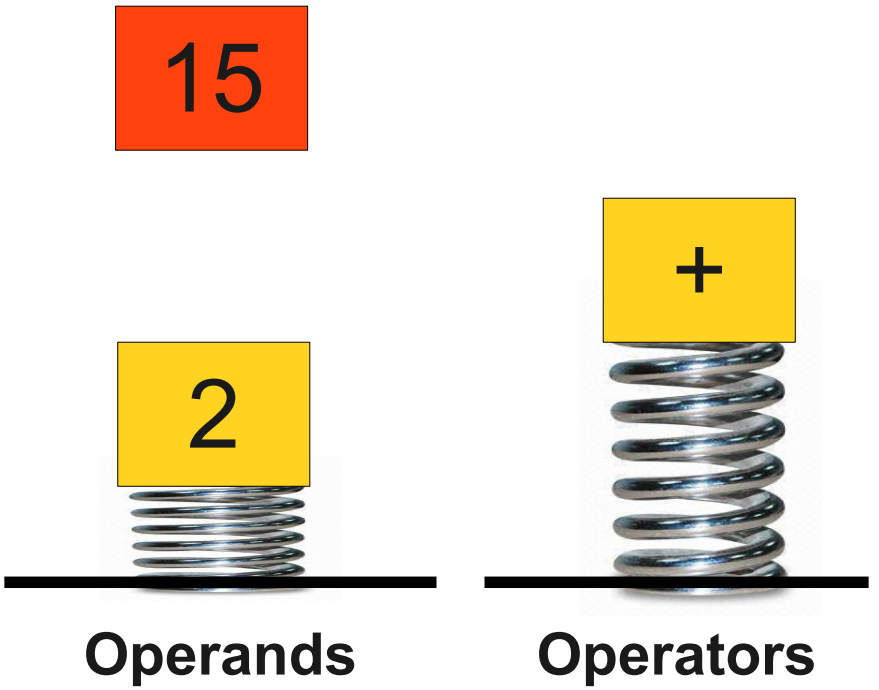
3	*	5



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

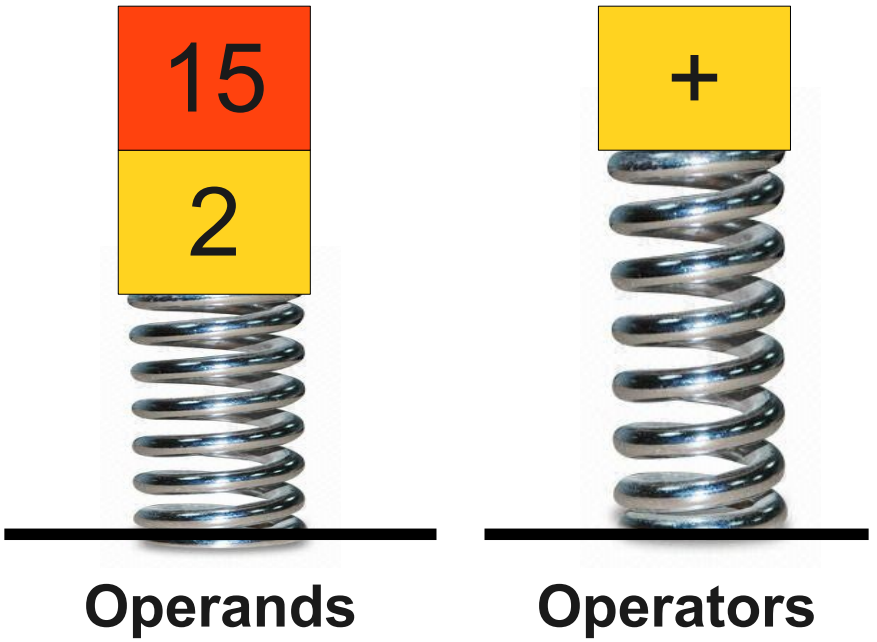
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

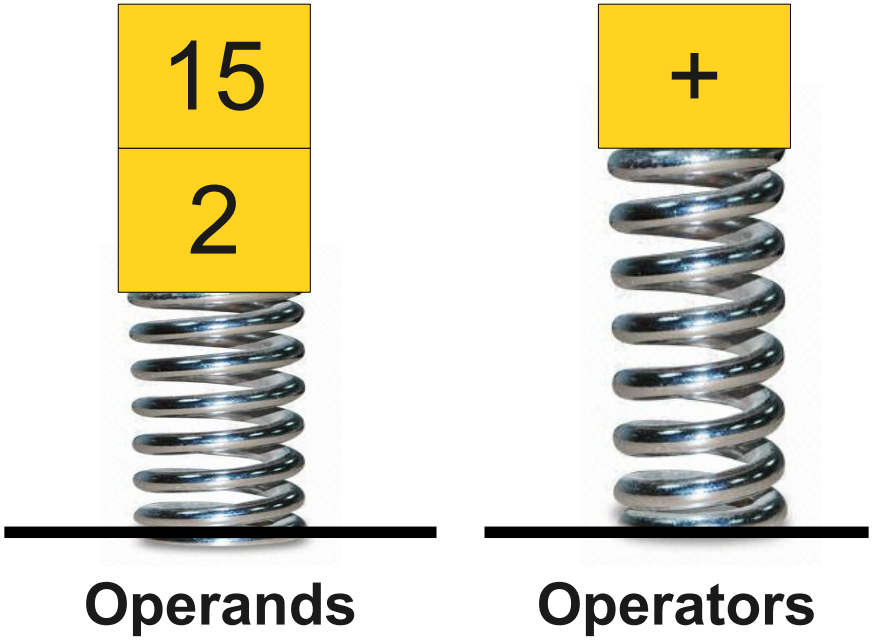
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

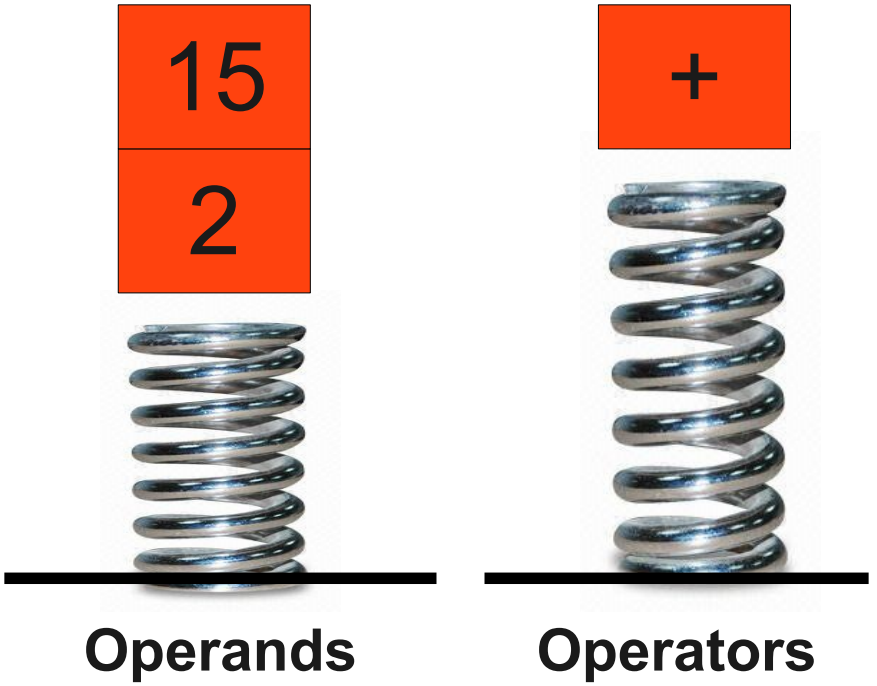
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

-	6	/	2

2	+	15

Operands **Operators**

The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

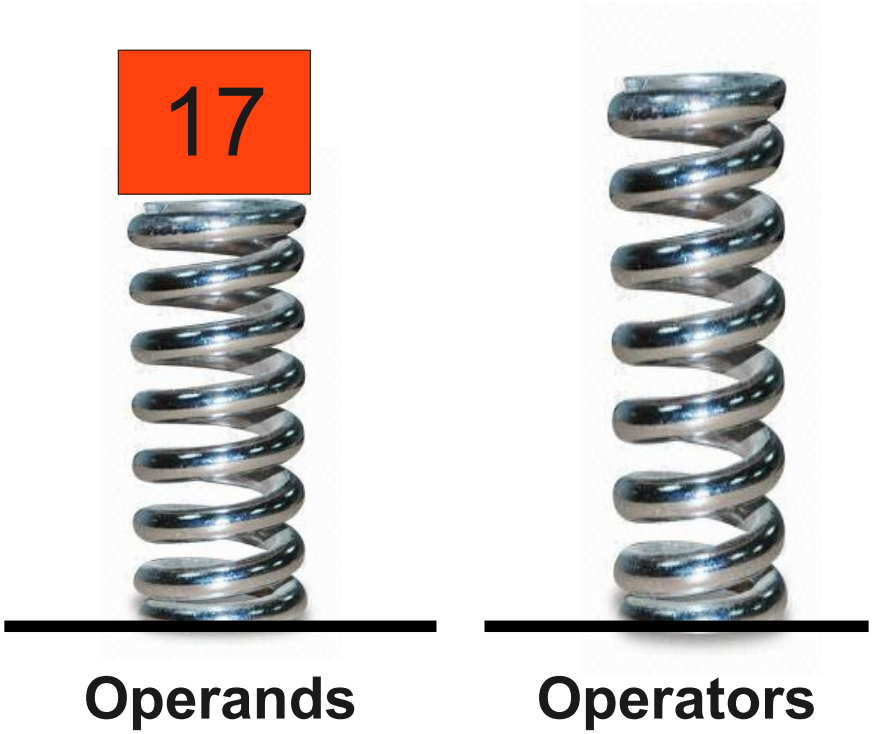
-	6	/	2

17

The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

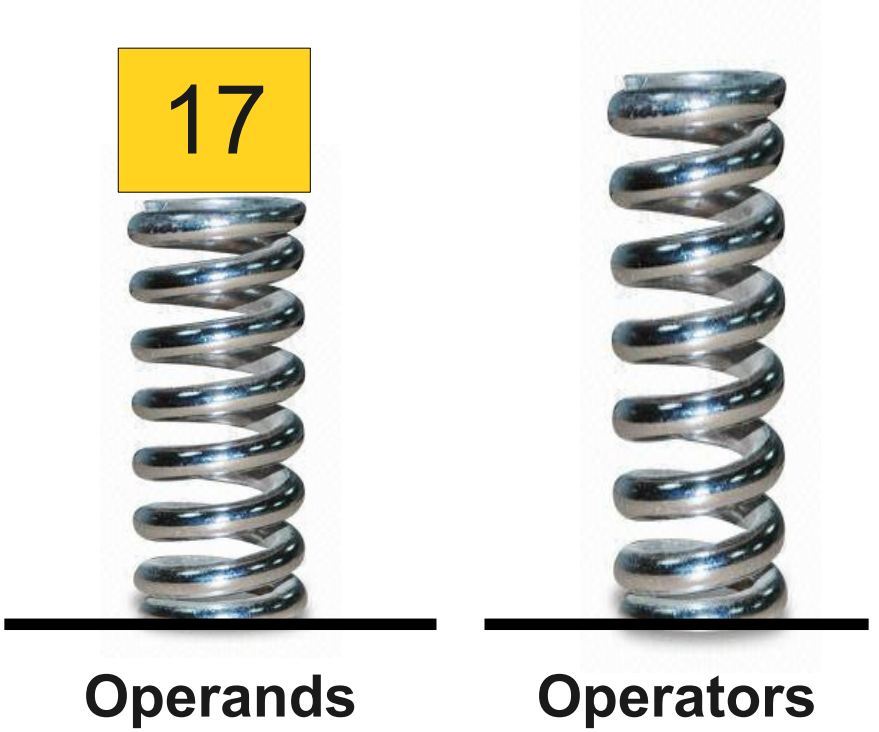
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

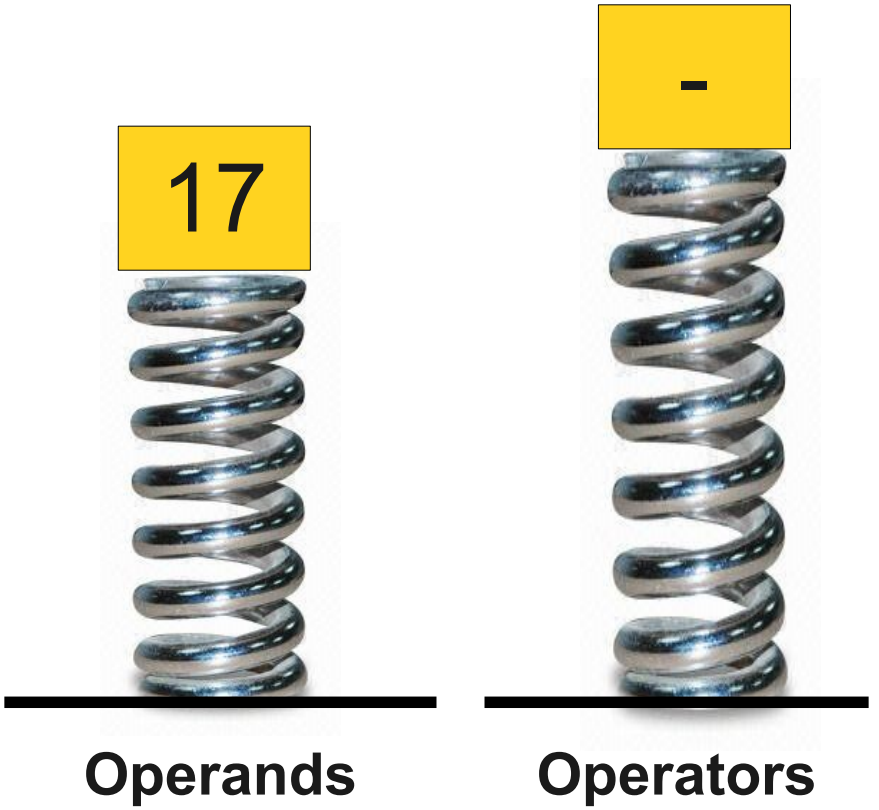
-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

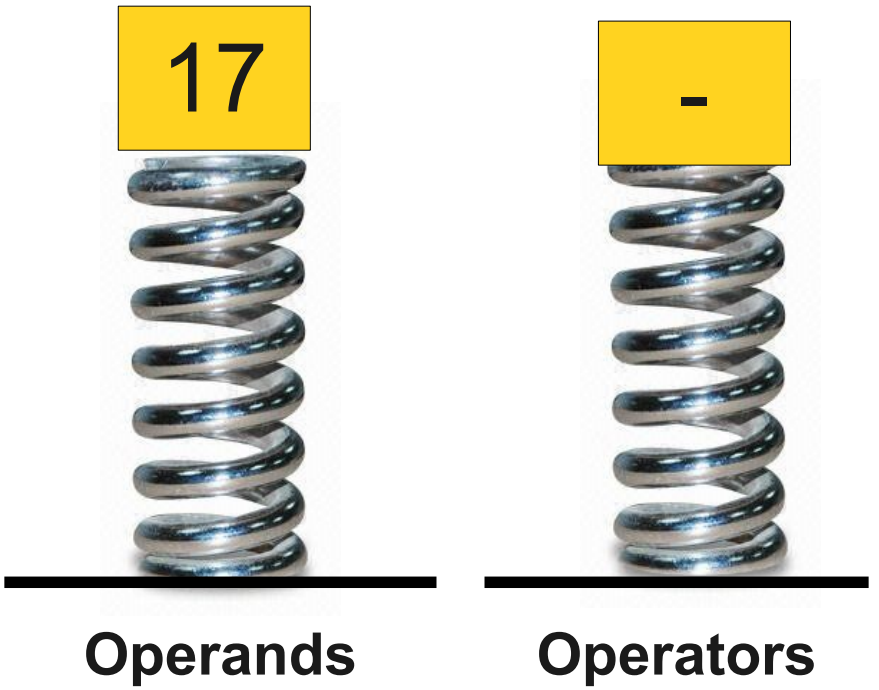
6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

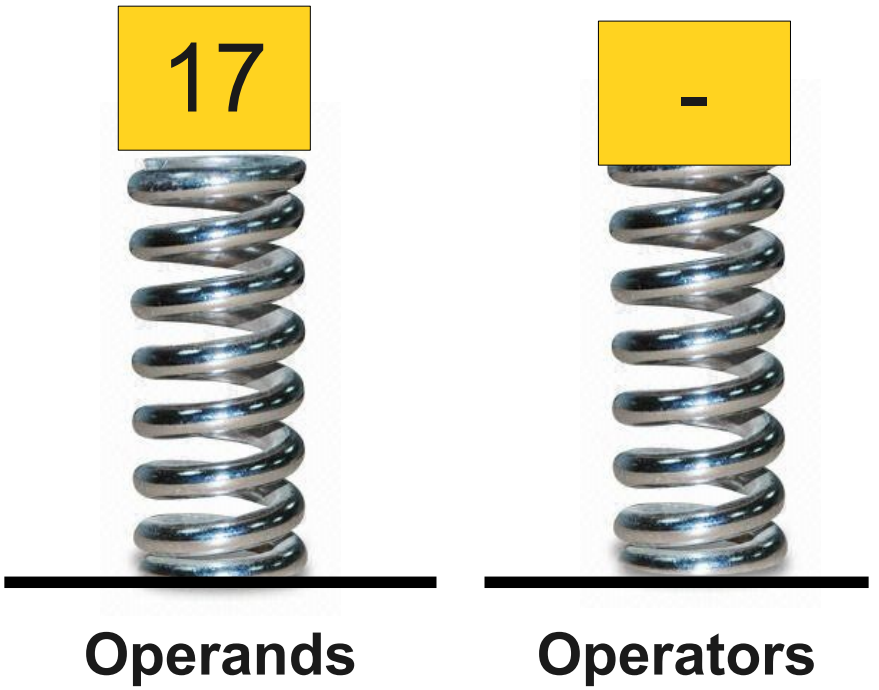
6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

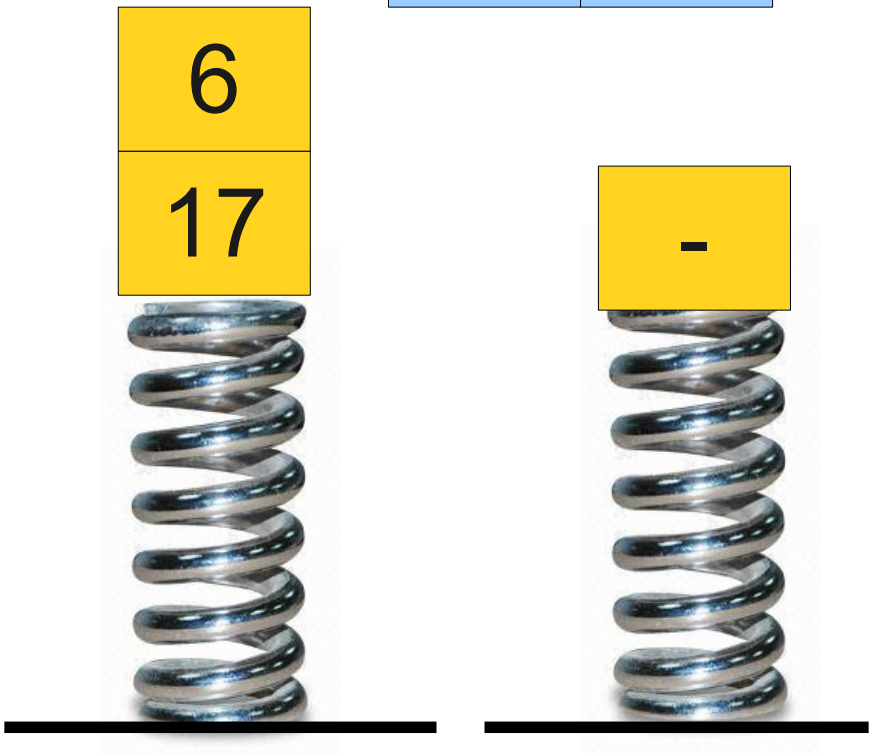
6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

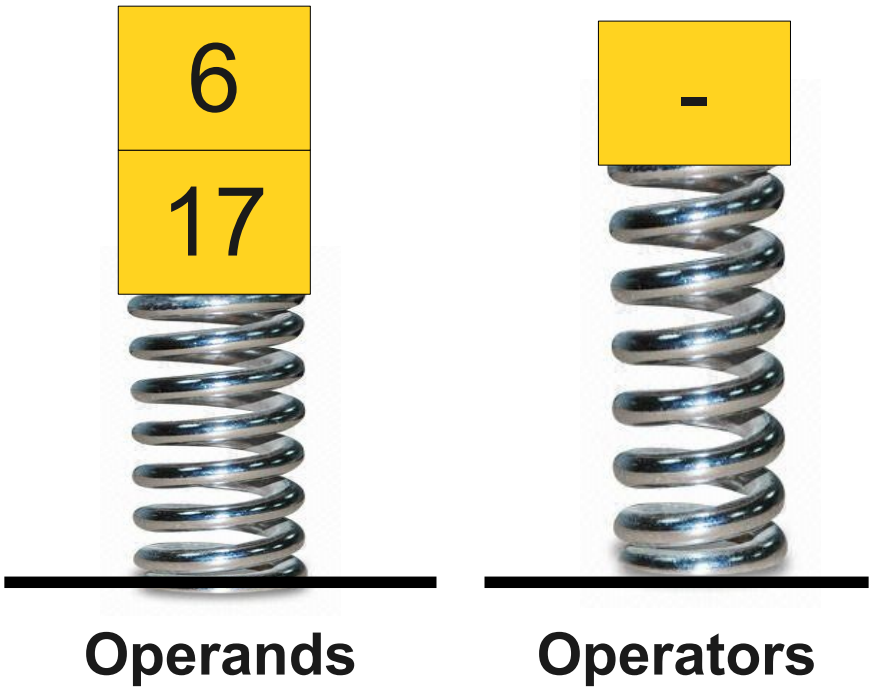
/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

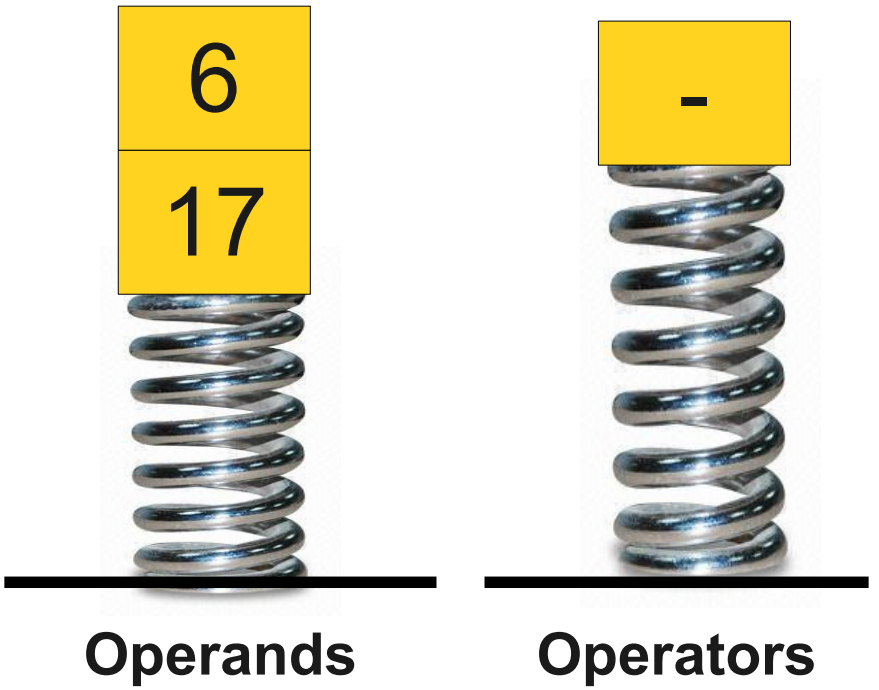
/	2



The Shunting-Yard Algorithm

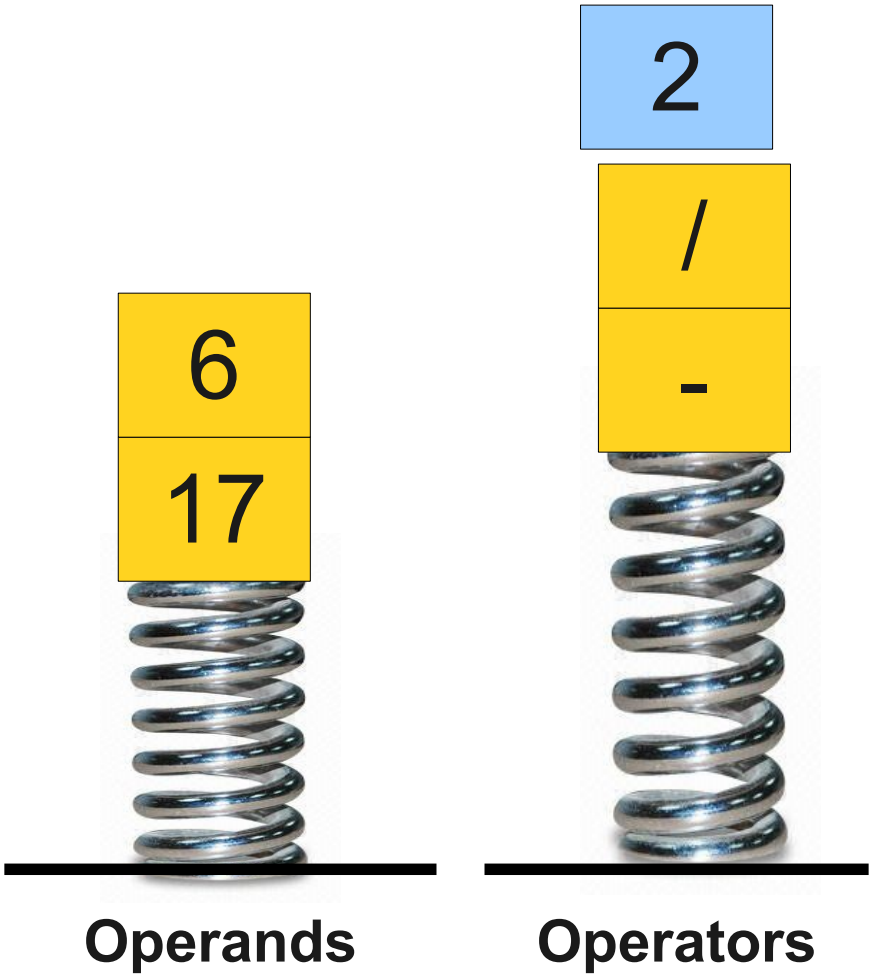
2	+	3	*	5	-	6	/	2

/	2



The Shunting-Yard Algorithm

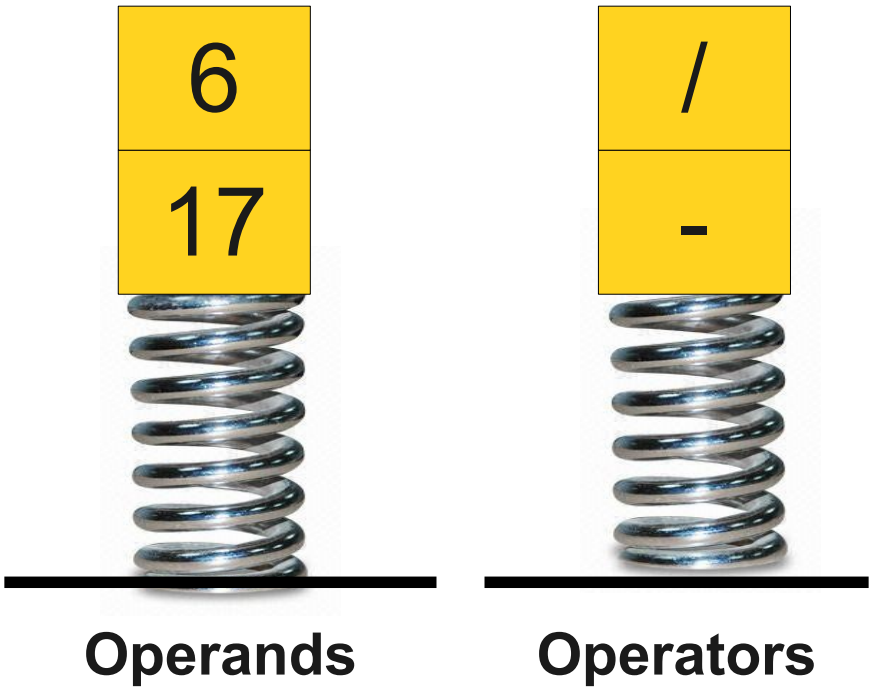
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

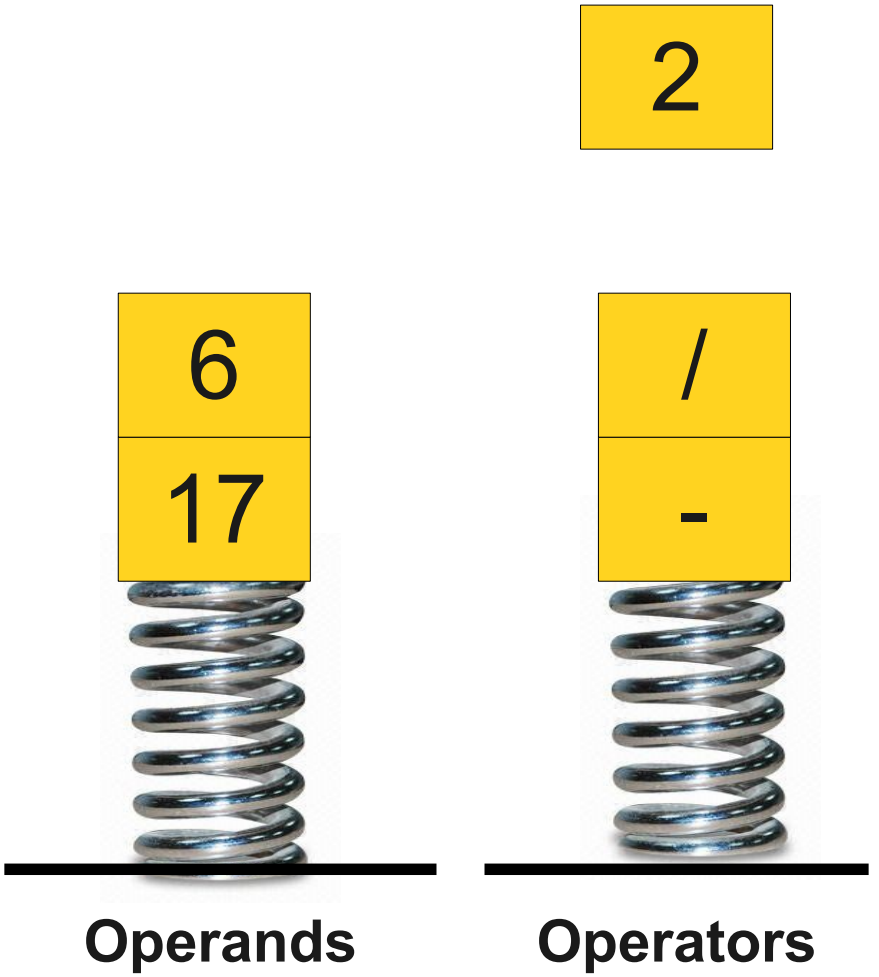
2	+	3	*	5	-	6	/	2

2



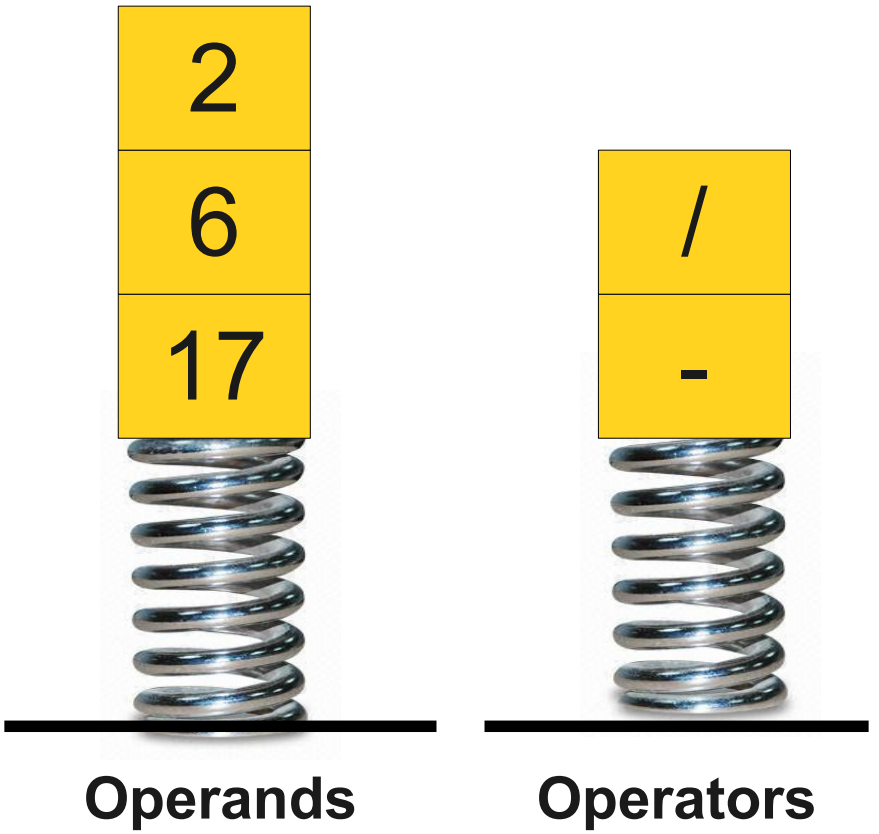
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



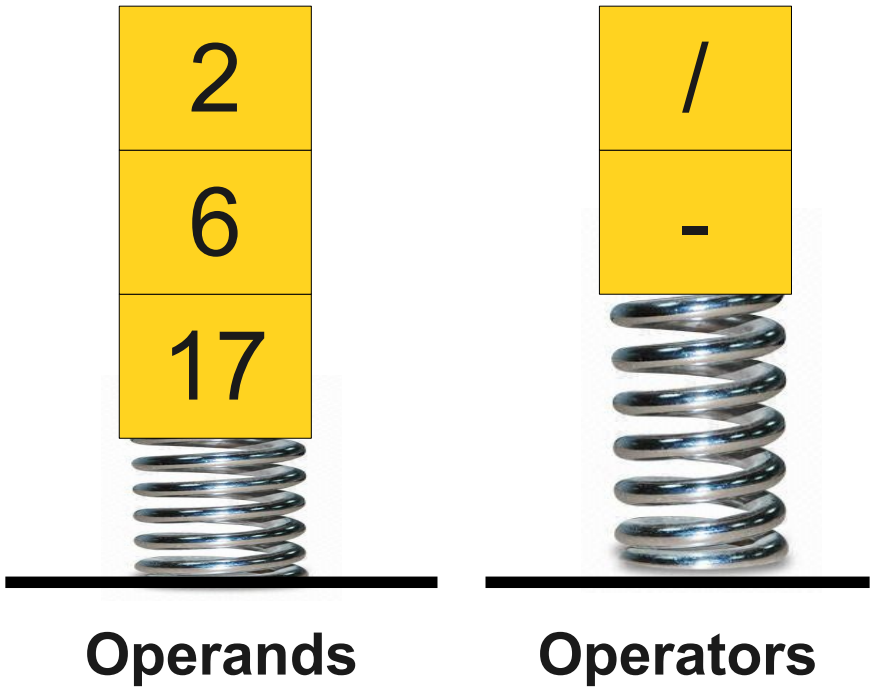
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



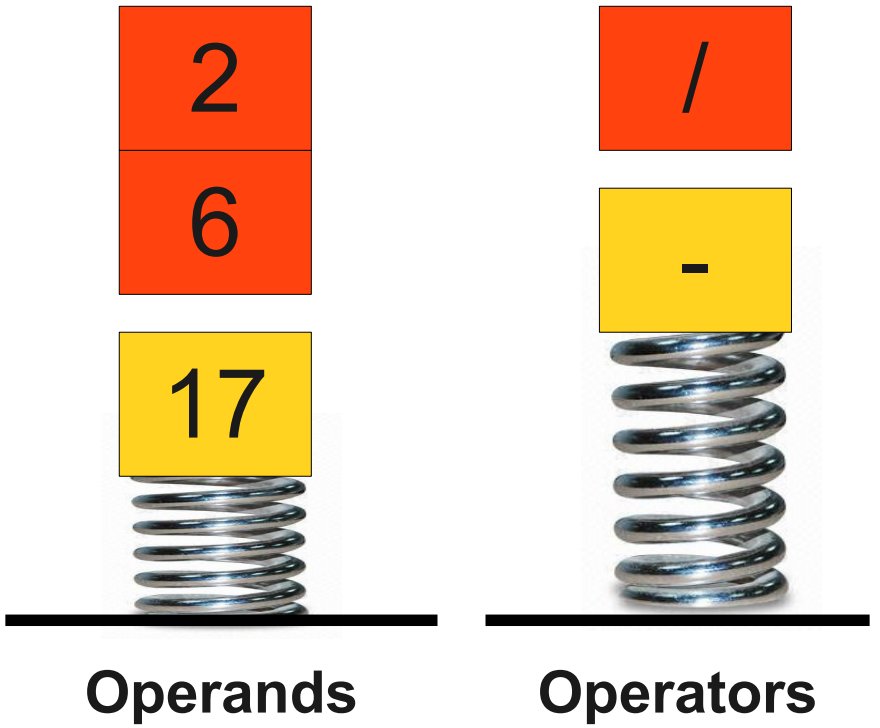
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

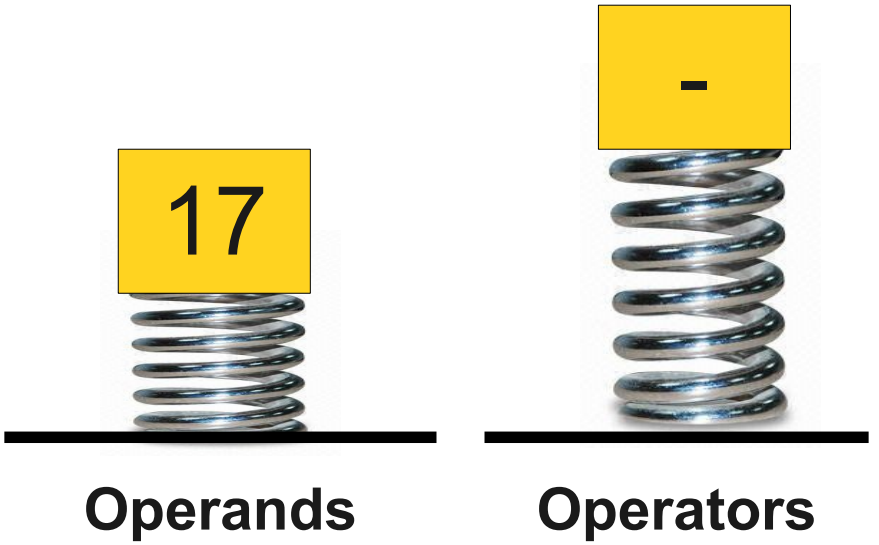
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

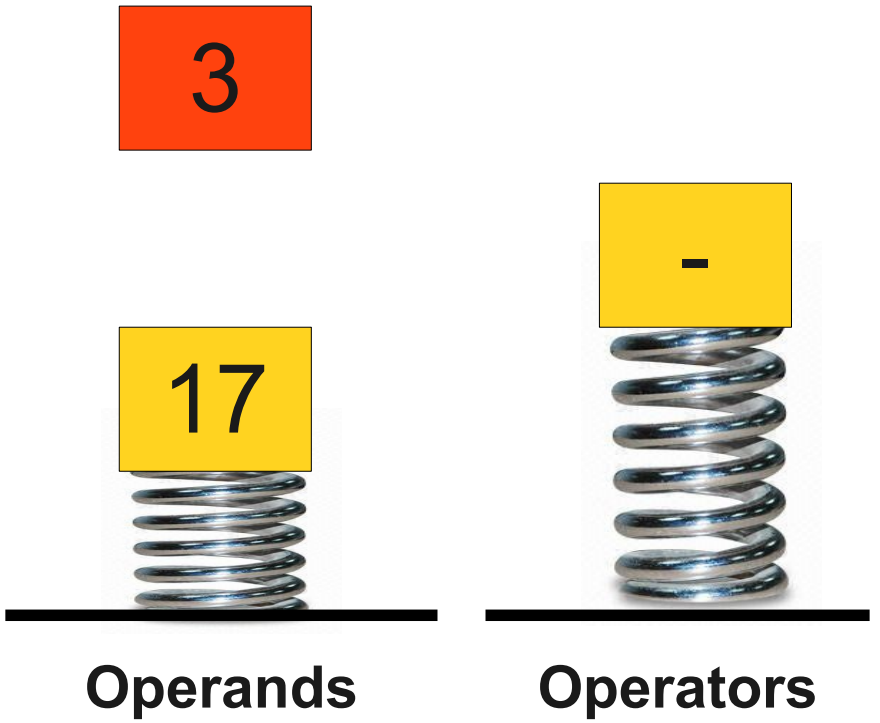
2	+	3	*	5	-	6	/	2

6	/	2



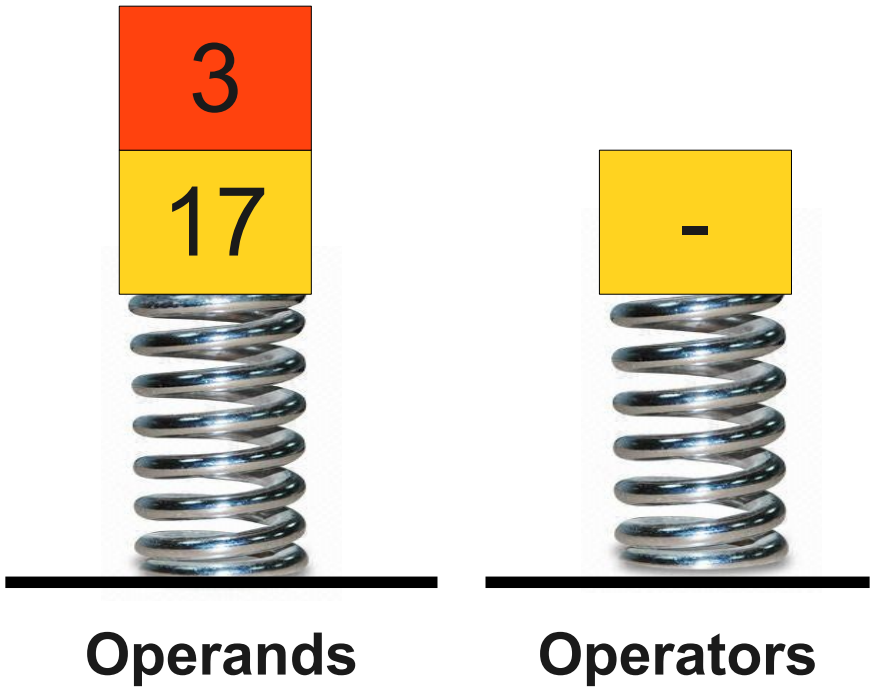
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



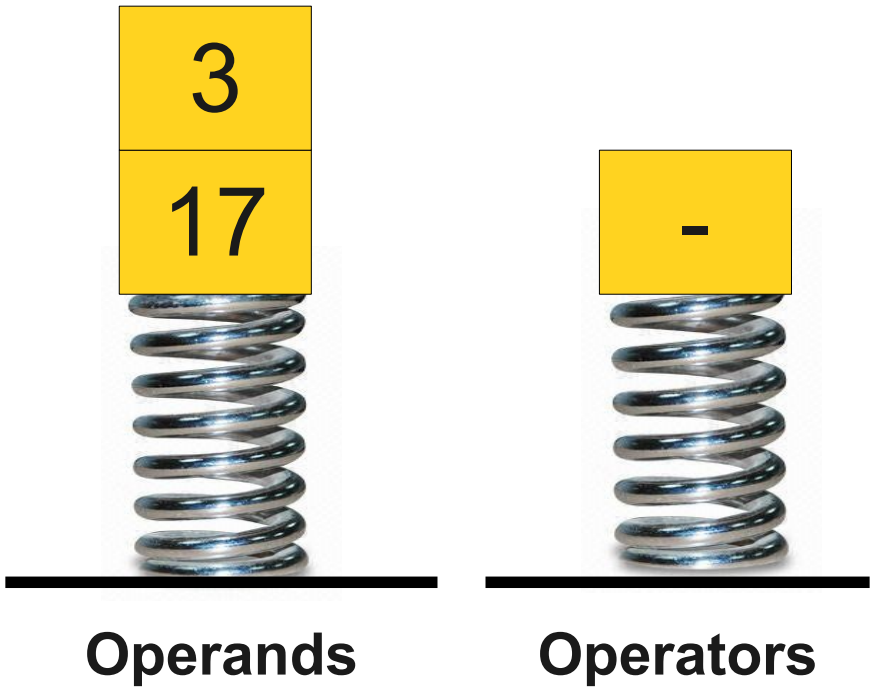
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



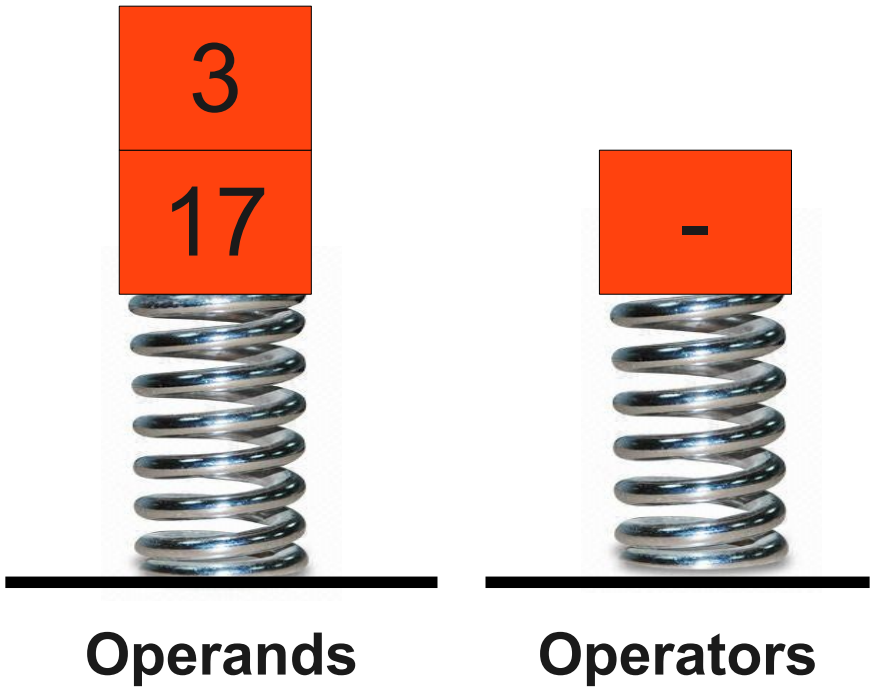
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

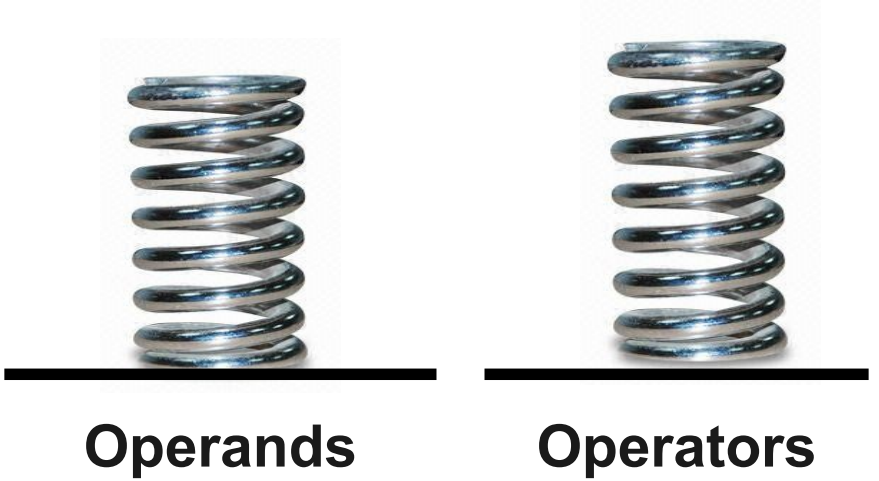
2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2

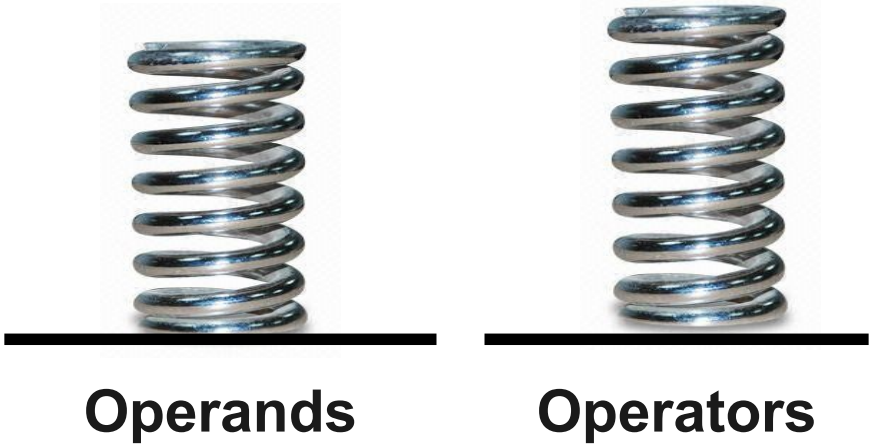
17	-	3



The Shunting-Yard Algorithm

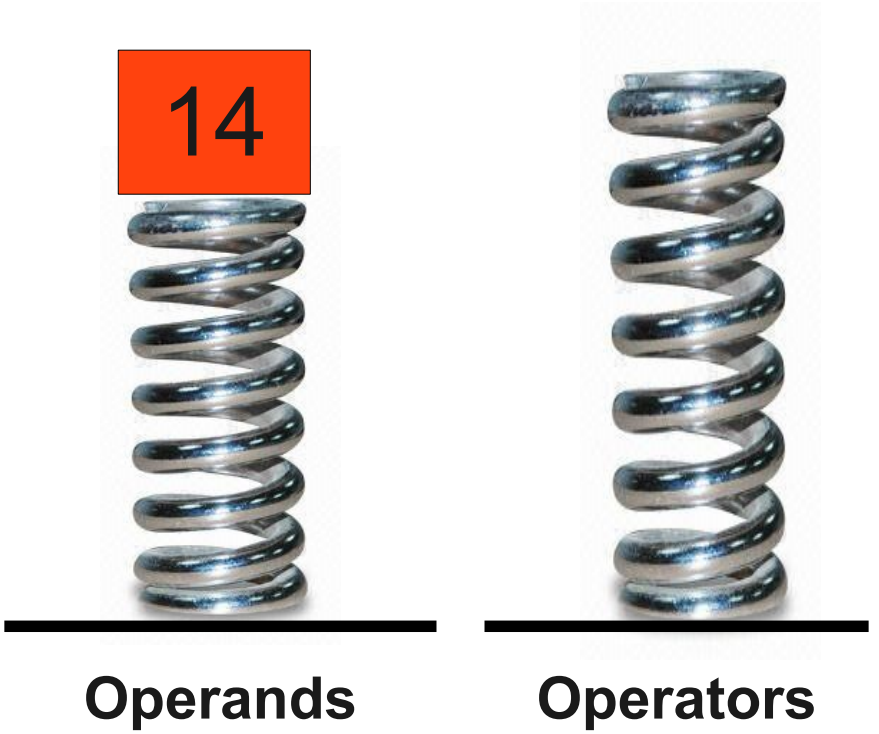
2	+	3	*	5	-	6	/	2

14



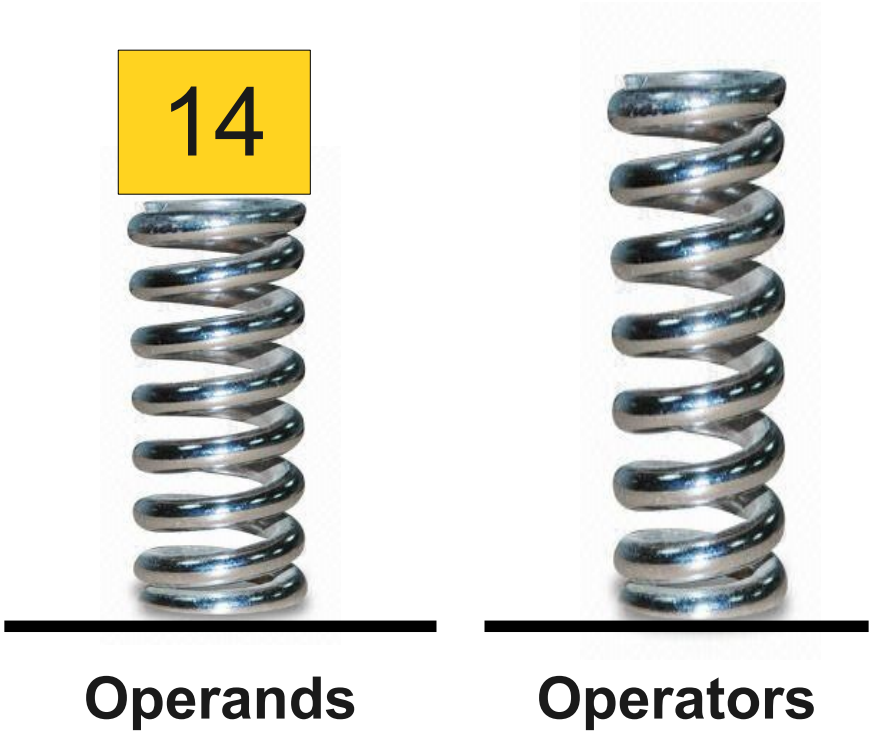
The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



The Shunting-Yard Algorithm

2	+	3	*	5	-	6	/	2



Extensions to Shunting-Yard

- How might you handle/report syntax errors in the input?
- How might you handle parentheses?
- What about functions like sin, cos, and tan?
- Could you add support for variables?