

# Manipulation of Data Part 2

By Kenneth Leroy Busbee

# Expression

A valid sequence of operand(s) and operator(s) that reduces (or *evaluates*) to a single value.

# Operator

A language-specific syntactical token (usually a symbol) that causes an action to be taken on one or more operands.

# Operand

A value that receives the operator's action.

# Precedence

Determines the order in which the operators are allowed to manipulate the operands. Higher precedence goes first.

# Associativity

Determines the order in which the operators of the same precedence are allowed to manipulate the operands.

# Statement

Within “C++” a line of code ending in a semicolon.

# Consider the Statement

`answer = 2 < 3 >= 4;`

Mark the Operator(s)  
(using an exclamation point or line)

answer = 2 < 3 >= 4;

! ! !

# Identify the Precedence

(using numbers with 1 being highest)

answer = 2 < 3 >= 4;

!   !   !  
2   1   1

# Identify Associativity

(using letters with 'a' being first)

answer = 2 < 3 >= 4;

!   !   !  
2   1a   1b

# Evaluation

The process of letting the operator(s) do their action to the operand(s).

# Evaluation – Step 1

(this changes to the next slide)

answer = 2 < 3 >= 4;

!   !   !  
2   1a   1b

# Evaluation – Step 2

(this changes to the next slide)

answer = 1 >= 4;

! !

2 1b

# Evaluation – Step 3

(this changes to the next slide)

answer = 0;

!

2

# Evaluation – Done

(the variable answer is assigned the value of '0' meaning false)

```
answer = 0;
```

**The End**