

Week# 3

Arithmetic Operators & Mathematical Expressions

Lecture# 7

Arithmetic Expression

Mathematical Expressions

- ❑ Can create complex expressions using multiple mathematical operators
- ❑ An expression can be a literal, a variable, or a mathematical combination of constants and variables
- ❑ Can be used in assignment, `cout`

```
area = 2 * PI * radius;
```

```
cout << "border is: " << 2*(1+w) ;
```

Mathematical Expressions in C++

□ Example of algebraic expressions

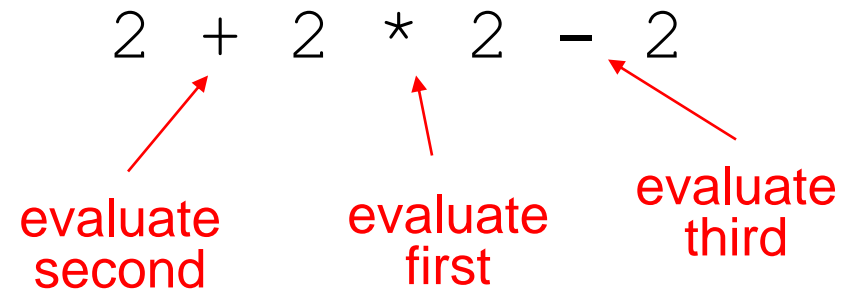
Algebraic expression	C++ expression
$a + 2x$	<code>a + 2*x</code>
$\sqrt{a^2 + b^2}$	<code>sqrt(a*a + b*b)</code>
$\sin\left(\frac{b}{\sqrt{a+b}}\right)$	<code>sin(b/sqrt(a+b))</code>

Precedence & Associativity

- ❑ *Precedence* defines the order of operations in expressions
- ❑ *Associativity* defines if an operator is grouped with right or left operand, in an expression

Operators	Precedence	Associativity
()	Highest	Left to right
* / %		Left to right
+ -	Lowest	Left to right

Precedence & Associativity



Expression	Value
$5 + 2 * 4$	13
$10 / 2 - 3$	2
$8 + 12 * 2 - 4$	28
$4 + 17 \% 2 - 1$	4
$6 - 3 * 2 + 7 - 1$	6

Precedence & Associativity

□ parentheses () can be used to override the order of operations:

$$2 + 2 * 2 - 2 = 4$$

$$(2 + 2) * 2 - 2 = 6$$

$$2 + 2 * (2 - 2) = 2$$

$$(2 + 2) * (2 - 2) = 0$$

Expression	Value
$(5 + 2) * 4$	28
$10 / (5 - 3)$	5
$8 + 12 * (6 - 2)$	56
$(4 + 17) \% 2 - 1$	0
$(6 - 3) * (2 + 7) / 3$	9

Practice Examples

- ❑ Write a program that returns the Celsius value for a given temperature measured in Fahrenheit. For example, input 68 gives 20. Use the conversion formula $5(F-32) = 9C$
- ❑ Write a program that converts a given number of inches to centimeters (1 inch = 2.54 cm)
- ❑ Write a program that inputs a number of hours and outputs the equivalent number of weeks, days and hours. For example, an input of 4000 would output 23 weeks, 5 days and 16 hours.

Relational, Logical, Assignment Operators

Relational Operators

Operator	Description	Example
<code>==</code>	Checks if the values of two operands are equal or not, if yes then condition becomes true.	<code>A == B</code>
<code>!=</code>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	<code>A != B</code>
<code>></code>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	<code>A > B</code>
<code><</code>	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	<code>A < B</code>
<code>>=</code>	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	<code>A >= B</code>
<code><=</code>	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	<code>A <= B</code>

Logic Operators

Operator	Description	Example
&&	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	If A = 1, B = 0 (A && B) is false.
	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	If A = 1, B = 0 (A B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	If A = 1, B = 0 !(A && B) is true.

Assignment & Combined Assignment Operators

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand.	<code>C = A + B</code> will assign value of <code>A + B</code> into <code>C</code>
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand.	<code>C += A</code> is equivalent to <code>C = C + A</code>
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand.	<code>C -= A</code> is equivalent to <code>C = C - A</code>
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand.	<code>C *= A</code> is equivalent to <code>C = C * A</code>
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand.	<code>C /= A</code> is equivalent to <code>C = C / A</code>
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand.	<code>C %= A</code> is equivalent to <code>C = C % A</code>