

S6: Lunes 6 enero

- Variables (basic types, declaration, initialization, scope rules, casting), constants, literal, keywords
- Standard library
- Arithmetic operators (+, -, *, /, %), precedence, and association
- Arithmetic expressions
- Math Library

Standard Library: No lo vamos a dar
Arithmetic Operators: + - * / %

Table 2-9 Fundamental Arithmetic Operators

Operator	Meaning	Example
+	Addition	<code>total = cost + tax;</code>
-	Subtraction	<code>cost = total - tax;</code>
*	Multiplication	<code>tax = cost * rate;</code>
/	Division	<code>salePrice = original / 2;</code>
%	Modulus	<code>remainder = value % 3;</code>

Precedencia:

Table 3-1 Precedence of Arithmetic Operators (Highest to Lowest)

()		Expressions within parentheses are evaluated first
-	unary	Negation of a value, e.g., -6
* / %	binary	Multiplication, division, and modulus
+ -	binary	Addition and subtraction

Asociación:

Table 3-3 Associativity of Arithmetic Operators

Operator	Associativity
(unary negation) -	Right to left
* / %	Left to right
+ -	Left to right

Arithmetic Expressions:

Table 3-4 More Arithmetic Expressions

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

Table 3-5 Algebraic and C++ Multiplication Expressions

Algebraic Expression	Operation	C++ Equivalent
6B	6 times B	$6 * B$
(3)(12)	3 times 12	$3 * 12$
4xy	4 times x times y	$4 * x * y$

Table 3-6 Algebraic and C++ Expressions

Algebraic Expression	C++ Expression
$y = 3\frac{x}{2}$	$y = x / 2 * 3;$
$z = 3bc + 4$	$z = 3 * b * c + 4;$
$a = \frac{3x + 2}{4a - 1}$	$a = (3 * x + 2) / (4 * a - 1)$

No Exponents Please!

Unlike many programming languages, C++ does not have an exponent operator. Raising a number to a power requires the use of a *library function*. The C++ library isn't a place where you check out books, but a collection of specialized functions. Think of a library function as a "routine" that performs a specific operation. One of the library functions is called `pow`, and its purpose is to raise a number to a power. Here is an example of how it's used:

```
area = pow(4.0, 2);
```

Math Library:

```
4 #include <cmath> // Needed for the pow function
```

Table 3-13 Selected Mathematical Library Functions

Function	Example	Description
<code>abs</code>	<code>y = abs(x);</code>	Returns the absolute value of the argument. The argument and the return value are integers.
<code>cos</code>	<code>y = cos(x);</code>	Returns the cosine of the argument. The argument should be an angle expressed in radians. The return type and the argument are doubles.
<code>exp</code>	<code>y = exp(x);</code>	Computes the exponential function of the argument, which is x . The return type and the argument are doubles.
<code>fmod</code>	<code>y = fmod(x, z);</code>	Returns, as a double, the remainder of the first argument divided by the second argument. Works like the modulus operator, but the arguments are doubles. (The modulus operator only works with integers.) Take care not to pass zero as the second argument. Doing so would cause division by zero.
<code>log</code>	<code>y = log(x);</code>	Returns the natural logarithm of the argument. The return type and the argument are doubles.
<code>log10</code>	<code>y = log10(x);</code>	Returns the base-10 logarithm of the argument. The return type and the argument are doubles.
<code>round</code>	<code>y = round(x);</code>	Returns the argument rounded to the nearest whole number. The return value is an integer.
<code>sin</code>	<code>y = sin(x);</code>	Returns the sine of the argument. The argument should be an angle expressed in radians. The return type and the argument are doubles.
<code>sqrt</code>	<code>y = sqrt(x);</code>	Returns the square root of the argument. The return type and argument are doubles. The argument must be zero or greater.
<code>tan</code>	<code>y = tan(x);</code>	Returns the tangent of the argument. The argument should be an angle expressed in radians. The return type and the argument are doubles.

Ejemplo: cmath library

Program 3-30

```
1 // This program inputs the lengths of the two sides of a right
2 // triangle, then calculates and displays the length of the hypotenuse.
3 #include <iostream>
4 #include <cmath>          // Needed to use the sqrt function
5 using namespace std;
6
7 int main()
8 {
9     double a, b, c;
10
11     // Get the length of the two sides
12     cout << "Enter the length of side a: ";
13     cin  >> a;
14     cout << "Enter the length of side b: ";
15     cin  >> b;
16
17     // Compute and display the length of the hypotenuse
18     c = sqrt(pow(a, 2.0) + pow(b, 2.0));
19
20     cout << "The length of the hypotenuse is ";
21     cout << c << endl;
22     return 0;
23 }
```