جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني
Department of Cyber Security

Subject:

Programming Fundamentals

Class:

1st stage

Lecturer:

Dr. Abdulkadhem A. Abdulkadhem

Lecture: (9)

Mastering the <cmath> Library in C++

Introduction

The `<cmath>` library in C++ provides essential mathematical functions to perform complex calculations. These functions are widely used in scientific computations, engineering applications, and problem-solving.

In this lecture, we will cover:

1. Overview of `<cmath>` library functions.
2. Translation of mathematical equations into C++ expressions.
3. Understanding and visualizing the order of operations.

1. Overview of `<cmath>` Functions

Below is a table summarizing key functions in the `<cmath>` library:

Functions

Function	Description	Example	Result
`sqrt(x)`	Computes the square root of `x`.	`sqrt(16)`	`4.0`
`pow(x, y)`	Computes `x` raised to the power `y` (`x^y`).	`pow(2, 3)`	`8.0`
`abs(x)`	Returns the absolute value of an integer `x`.	`abs(-5)`	`5`
`fabs(x)`	Returns the absolute value of a floating-point number `x`.	`fabs(-5.6)`	`5.6`
`round(x)`	Rounds `x` to the nearest integer.	`round(4.5)`	`5`
`ceil(x)`	Rounds `x` up to the smallest integer greater than or equal to `x`.	`ceil(4.2)`	`5`
`floor(x)`	Rounds `x` down to the largest integer less than or equal to `x`.	`floor(4.8)`	`4`
`log(x)`	Computes the natural logarithm (base `e`) of `x`.	`log(10)`	`2.3026`
`log10(x)`	Computes the base-10 logarithm of `x`.	`log10(100)`	`2`
`sin(x)`	Computes the sine of `x` (in radians).	`sin(3.14 / 2)`	`1.0`
`cos(x)`	Computes the cosine of `x` (in radians).	`cos(3.14)`	`-1.0`
`tan(x)`	Computes the tangent of `x` (in radians).	`tan(3.14 / 4)`	`1.0`
`exp(x)`	Computes the exponential value of x ($e^{\wedge x}$, where e ≈ 2.71828).	`exp(1)`	`2.71828`

Example:

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {


    // Testing basic cmath functions
    cout << "ceil(5.8) = " << ceil(5.8) << endl;
    cout << "floor(5.8) = " << floor(5.8) << endl;
    cout << "abs(-4.7) = " << abs(-4.7) << endl;
    cout << "pow(3, 3) = " << pow(3, 3) << endl;
    cout << "sqrt(16) = " << sqrt(16) << endl;
    cout << "log(2.71828) = " << log(2.71828) << endl;  // Approximation of e
    cout << "log10(1000) = " << log10(1000) << endl;

    // Trigonometric functions (angles in radians)
    cout << "sin(0.7854) = " << sin(0.7854) << endl;  // Approx. π/4 radians
    cout << "cos(0.7854) = " << cos(0.7854) << endl;
    cout << "tan(0.7854) = " << tan(0.7854) << endl;


    return 0;
}
```

```
Output:
ceil(5.8) = 6
floor(5.8) = 5
abs(-4.7) = 4.7
pow(3, 3) = 27
sqrt(16) = 4
log(2.71828) = 1
log10(1000) = 3
sin(0.7854) = 0.707107
cos(0.7854) = 0.707107
tan(0.7854) = 1
sinh(1) = 1.1752
cosh(1) = 1.54308
tanh(1) = 0.761594
```

2. Translating Equations into C++

To use `<cmath>`, we often translate mathematical equations into equivalent C++ expressions. Let's explore examples that combine functions.

Example 1: Simple Equation

Write the following equation in C++ and determine its order of evaluation:

$$f = \sqrt{\frac{\sin(x) + x^3}{\log(x) - x/2}}$$

Solution: C++ Expression:

```
f = sqrt((sin(x) + pow(x, 3)) / (log(x) - x / 2));
```

Order of Evaluation:

1. Compute `sin(x)`.
2. Compute `pow(x, 3)`.
3. Add `sin(x) + pow(x, 3)`.
4. Compute `log(x)`.
5. Compute `x / 2`.
6. Subtract `log(x) - x / 2`.
7. Divide the numerator by the denominator.
8. Take the square root using `sqrt`.

Example 2: Trigonometric Equation

Write the following equation in C++:

$$g = \sqrt{\frac{\tan(x) - e^x}{\sin(x) + \frac{x^2}{3}}}$$

Solution: C++ Expression:

```
g = sqrt((tan(x) - exp(x)) / (sin(x) + pow(x, 2) / 3));
```

Order of Evaluation:

1. Compute `tan(x)`.
2. Compute `exp(x)`.
3. Subtract `tan(x) - exp(x)`.
4. Compute `sin(x)`.
5. Compute `pow(x, 2)`.
6. Divide `pow(x, 2) / 3`.
7. Add `sin(x) + (pow(x, 2) / 3)`.
8. Divide the numerator by the denominator.
9. Take the square root using `sqrt`.

Example 3: Nested Equation

Write the following equation in C++:

$$h = \sqrt{\frac{\ln(x) + \sqrt{x}}{\cos(x) \cdot x^4}}$$

Solution: C++ Expression:

```
h = sqrt((log(x) + sqrt(x)) / (cos(x) * pow(x, 4)));
```

Order of Evaluation:

1. Compute `log(x)`.
2. Compute `sqrt(x)`.
3. Add `log(x) + sqrt(x)`.
4. Compute `cos(x)`.
5. Compute `pow(x, 4)`.
6. Multiply `cos(x) * pow(x, 4)`.
7. Divide the numerator by the denominator.
8. Take the square root using `sqrt`.

3. Understanding Order of Operations

C++ follows **precedence rules** for operators, but `<cmath>` functions have their own evaluation order based on the equation.

1. Parentheses `()` override normal precedence.
2. Function calls (e.g., `sqrt`, `pow`) are evaluated from **innermost to outermost**.
3. Multiplication, Division, Addition, and Subtraction follow the standard **operator precedence**.

For complex expressions, **breaking them into smaller steps** improves readability and ensures correctness.

4. Practice Problems

Problem 1: Write the following equation in C++ and determine the order of evaluation:

$$k = \sqrt{\frac{\cos(x) \cdot x^2}{\ln(x) - \tan(x)}}$$

Problem 2: Write a C++ program that calculates the value of:

$$f = \sqrt{\sin(x) \cdot \cos(x)} + \log(x^2)$$

Conclusion

The `<cmath>` library simplifies mathematical computations in C++. By understanding how to translate equations and evaluate operations step-by-step, you can efficiently handle complex calculations in your programs.