

[2.9 Constant variables](#)

Students: Section 2.10 is a part of 2 assignments: [CSC108 CH02.1-2.10 C2A](#) ▾ Includes: CA Due: 02/04/2025, 11:59 PM EST  
This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

## 2.10 Using math functions

### Basics

Some programs require math operations beyond +, -, \*, /, like computing a square root. A standard **math library** has about 20 math operations, known as functions. A programmer can include the library and then use those math functions.

A **function** is a list of statements executed by invoking the function's name, such as invoking `sqrt` is known as a **function call**. Any function input values, or **arguments**, appear within (), separated by commas if more than one. Below, function `sqrt` is called with one argument, `areaSquare`. The function call evaluates to a value, as in `sqrt(areaSquare)` below evaluating to 7.0, which is assigned to `sideSquare`.

PARTICIPATION ACTIVITY | 2.10.1: Using a math function.

Start □ 2x speed

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double sideSquare;
    double areaSquare = 49.0;

    sideSquare = sqrt(areaSquare);
    cout << "Square root of " << areaSquare
        << " is " << sideSquare << endl;
    return 0;
}
```

49.0 → sqrt → 7.0

Captions ^  
1. Some calculations require more than the +, -, \*, / operators. A programmer can include the cmath library, and then use various math functions like `sqrt` for square root.  
2. A function is like a black box. The `sqrt` function takes an input value, and produces that value's square root.  
3. Thus, `sqrt(49.0)` evaluates to 7.0.

Feedback?

Table 2.10.1: A few common math functions from the math library.

Function	Behavior	Example
<code>sqrt(x)</code>	Square root of x	<code>sqrt(9.0)</code> evaluates to 3.0.
<code>pow(x, y)</code>	Power:	<code>pow(6.0, 2.0)</code> evaluates to 36.0.
<code>fabs(x)</code>	Absolute value of x	<code>fabs(-99.5)</code> evaluates to 99.5.

Feedback?

Other available functions are `log` (natural log), `log2` (log base 2), `log10` (log base 10), `exp` (raising e to a power), `ceil` (rounding up), `floor` (rounding down), various trigonometric functions like `sin`, `cos`, `tan`, and more. See this [math functions](#) link for a comprehensive list of built-in math functions.

PARTICIPATION ACTIVITY | 2.10.2: Math functions.

1) `sqrt(36.0)` evaluates to \_\_\_\_\_.  
• 6.0  
• 36.0

2) What is `y`?  
`y = sqrt(81.0);`  
• 9.0  
• 81.0

3) What is `y`?  
`y = pow(2.0, 8.0);`  
• 64.0  
• 256.0

4) Is this a valid function call?  
`y = sqrt(2.0, 8.0);`  
• Yes  
• No

5) Is this a valid function call?  
`y = pow(8.0);`  
• Yes  
• No

6) If `w` and `x` are double variables, is this a valid function call?  
`y = pow(w, x);`  
• Yes  
• No

7) What is `y`?  
`w = 3.0;`  
`y = pow(w + 1.0, 2.0);`  
• 8.0  
• 16.0

Feedback?

### Example: Mass growth

The example below computes the growth of a biological mass, such as a tree. If the growth rate is 5% per year, the program computes 1.05 raised to the number of years. A similar program could calculate growth of money given an interest rate.

Figure 2.10.1: Math function example: Mass growth.

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double initMass; // Initial mass of a substance
    double growthRate; // Annual growth rate
    double yearsGrow; // Years of growth
    double finalMass; // Final mass after those years

    cout << "Enter initial mass: ";
    cin >> initMass;

    cout << "Enter growth rate (Ex: 0.05 is 5%/year): ";
    cin >> growthRate;

    cout << "Enter years of growth: ";
    cin >> yearsGrow;

    finalMass = initMass * pow(1.0 + growthRate, yearsGrow);
    // Ex: Rate of 0.05 yields initMass * 1.05^yearsGrow

    cout << "Final mass after " << yearsGrow
        << " years is: " << finalMass << endl;

    return 0;
}
```

Enter initial mass: 10000  
Enter growth rate (Ex: 0.05 is 5%/year): 0.05  
Enter years of growth: 20  
Final mass after 20 years is: 32071.4  
...  
Enter initial mass: 10000  
Enter growth rate (Ex: 0.05 is 5%/year): 0.40  
Enter years of growth: 10  
Final mass after 10 years is: 289255

Feedback?

PARTICIPATION ACTIVITY | 2.10.3: Growth rate.

1) If `initMass` is 10.0, `growthRate` is 1.0 (100%), and `yearsGrow` is 3, what is `finalMass`?  
`finalMass = initMass * pow(1.0 + growthRate, yearsGrow);`

Check Show answer

Feedback?

PARTICIPATION ACTIVITY | 2.10.4: Calculate Pythagorean theorem using math functions.

Select the three statements needed to calculate the value of `x` in the following:  
For this exercise, calculate before.

1) First statement is:  
• `temp1 = pow(x, 2.0);`  
• `temp1 = pow(z, 3.0);`  
• `temp1 = pow(y, 2.0);`  
• `temp1 = sqrt(y);`

2) Second statement is:  
• `temp2 = sqrt(x, 2.0);`  
• `temp2 = pow(z, 2.0);`  
• `temp2 = pow(z);`  
• `temp2 = x + sqrt(temp1 + temp2);`

3) Third statement is:  
• `temp2 = sqrt(temp1 + temp2);`  
• `x = pow(temp1 + temp2, 2.0);`  
• `x = sqrt(temp1 + temp2);`  
• `x = sqrt(temp1 + temp2);`

Feedback?

### Calls in arguments

Commonly a function call's argument itself includes a function call. Below, `z` is computed via `pow(x, y)`. The result is used in an expression that is an argument to another call, in this case to `pow()` again: `pow(2.0, pow(x, y) + 1)`.

PARTICIPATION ACTIVITY | 2.10.5: Function call in an argument.

Start □ 2x speed

$z = 2^{(x^y + 1)}$

$x^y \rightarrow \text{pow}(x, y)$

$2^{(x^y + 1)} \rightarrow \text{pow}(2.0, \text{pow}(x, y) + 1)$

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double x;
    x = sqrt(9.0);

    // Note: Trailing zeros not output
    // Ex: 9.0 is output as 99 (no .0)
    cout << x << endl;
    return 0;
}
```

3

Captions ^  
1. can be computed using `pow(x, y)`.  
2. A function's argument can be an expression, including a call to another function. can be computed as `pow(2.0, pow(x, y) + 1)`.  
3. Upon execution, if `x = 3.0` and `y = 2.0`, then `pow(x, y)` is called and evaluates to 9.0. Next, `pow(2.0, 9.0+1)` is called, yielding 1024.0.

Feedback?

PARTICIPATION ACTIVITY | 2.10.6: Function calls in arguments.

Type the ending value of `z`:

1) `z = pow(2.0, pow(2.0, 3.0));`

Check Show answer

2) `x = 9.0;`  
`z = pow(sqrt(x) + sqrt(x), 2.0);`

Check Show answer

3) `x = -9.0;`  
`z = sqrt(fabs(x));`

Check Show answer

Feedback?

chmath and cstdlib

The 'c' in chmath indicates that the library comes from a C language library.

Some math functions for integers are in a library named `cstdlib`, requiring `#include <cstdlib>`. `Ex: abs()` computes the absolute value of an integer.

CHALLENGE ACTIVITY | 2.10.1: Math functions.

Start

Type the program's output

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double x;
    x = sqrt(9.0);

    // Note: Trailing zeros not output
    // Ex: 9.0 is output as 99 (no .0)
    cout << x << endl;
    return 0;
}
```

3

1 2 3 4 5

Check Next

Feedback?

CHALLENGE ACTIVITY | 2.10.2: Using math functions to calculate the distance between two points.

Organize the code statements to find the distance between point (`x1, y1`) and point (`x2, y2`), and assign the result to `pointsDistance`. The calculation is:

Click here for example ▾

Note: Not all code statements on the left will be used in the final solution.

How to use this tool ▾

Used

```
yDist = pow((y2 - y1), 2);
yDist = pow((x2 - x1), 2);
xDist = pow((x2 - x1), 2);
pointsDistance = sqrt(xDist + yDist);
```

main.cpp Load default template...

3

1 2 3 4 5

Check Next

Feedback?

CHALLENGE ACTIVITY | 2.10.3: Writing math calculations.

Start

Compute:

If the input is 4.00 9.00, then the output is:

`d = 6.00`

```
1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 using namespace std;
5
6 int main() {
7     double b;
8     double c;
9     double d;
10
11    cin >> b;
12    cin >> c;
13
14    /* Your code goes here */
15
16    cout << "d = ";
17    cout << fixed << setprecision(2); // setprecision(2) outputs d with 2 decimal places
```

1 2 3

Check Next level

Feedback?

How was this section? Provide section feedback

Activity summary for assignment: [CSC108 CH02.1-2.10 C2A](#) ▾ 41 / 41 points  
Due: 02/04/2025, 11:59 PM EST  
This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See [this article](#) for more info.

Completion details ▾

41 / 41 points submitted to BlackboardLearn

41 / 41 points

41 / 41 points submitted to BlackboardLearn

41 / 41 points

41 / 41 points