

C++ Division

C++ Division Arithmetic Operation

In C++, Division is performed using arithmetic operator `/`. The operator takes two operands and returns the division of left operand by the right operand.

In this tutorial, we shall learn how to use Arithmetic Division Operator with values of different datatypes using example programs.

Syntax of C++ Division Operator

Following is the syntax of Arithmetic Division Operator in C++.

```
result = operand_1 / operand_2
```

operand_2 tries to divide operand_1 into equal parts. operand_1 is the dividend and operand_2 is the divisor.

Based on the datatype of operands and result, the result would be able to store precision or not.

C++ Division with Two Integers

You can divide two integers using division operator. The datatype of the operands and returned value is given in the following code snippet.

```
int = int / int
```

As both the operands are integers, if dividend is not exactly divisible by divisor, the division operator returns only quotient and the remainder is discarded.

In the following program, we initialize two integer variables and pass them as operands to division operator.

C++ Program

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

```
int main() {  
    int a = 16;  
    int b = 7;  
  
    int div = a / b;  
  
    cout << div << endl;  
}
```

Output

2

C++ Division with Two Floating Point Numbers

You can divide two floating point numbers using division operator. The datatype of the operands and returned value is given in the following code snippet.

```
float = float / float
```

As both dividend and divisor are floating point numbers, the division operator divides dividend by divisor until the full precision of floating point number.

C++ Program

```
#include <iostream>  
using namespace std;  
  
int main() {  
    float a = 21.3;  
    float b = 4.1;  
  
    float div = a / b;  
  
    cout << div << endl;  
}
```

Output

5.19512

C++ Division with Integer and Floating Point Number

You can divide a floating point number with integer. The datatype of the operands and returned value is given in the following code snippet.

```
float = int / float
```

In the following program, we initialize an integer variable and a floating point variable, divide them and store in a float variable. The precision is preserved. But, if you store the result in an integer, the precision is lost.

C++ Program

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

int main() {
    int a = 16;
    float b = 7.1;

    float div = a / b;

    cout << div << endl;
}
```

Output

```
2.25352
```

Chaining of Division Operator

You can chain Division Operator and perform the division of more than two operands in a single statement. The pseudo code is given below.

```
result = operand_1 / operand_2 / operand_3 / ... / operand_n
```

In the following example, we shall take four integer variables and divide them altogether in a single statement using arithmetic division operator. The calculation happens from left to right. So firstly `operand_1 / operand_2` is performed, then the result is divided by `operand_3` and so on.

C++ Program

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

int main() {
    int a = 100;
    int b = 2;
    int c = 10;

    int div = a / b / c;

    cout << div << endl;
}
```

Output

5

Conclusion

In this [C++ Tutorial](#), we learned how to use C++ Division Operator on values of different datatypes, etc.

C++ Tutorials

- ◆ [C++ Tutorial](#)
- ◆ [C++ Hello World Program](#)
- ◆ [C++ If Else](#)
- ◆ [C++ Switch](#)
- ◆ [C++ Ternary Operator](#)
- ◆ [C++ Logical Operations](#)
- ◆ [C++ Arithmetic Operations](#)
- ◆ [C++ While Loop](#)
- ◆ [C++ Do-While Loop](#)
- ◆ [C++ For Loop](#)
- ◆ [C++ ForEach](#)
- ◆ [C++ Continue](#)
- ◆ [C++ Break](#)
- ◆ [C++ Comments](#)
- ◆ [C++ Recursion](#)
- ◆ [C++ Try Catch](#)
- ◆ [C++ String Operations](#)
- ◆ [C++ Array Operations](#)
- ◆ [C++ Vector Operations](#)
- ◆ [C++ Input Output Operations](#)
- ◆ [C++ Class](#)
- ◆ [C++ Programs](#)

