

C++ Modulus Operator

C++ Modulus Arithmetic Operation

In C++, Modulus is performed using arithmetic operator `%`. Modulus is also called modular division or modulo. The operator takes two operands and returns the remainder after performing division of dividend by divisor.

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

Syntax of C++ Modulus Operator

Following is the syntax of Arithmetic Modulus 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. But

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

Datatypes of Operands to Modulus Operator

C++ Modulus Operator takes only signed or unsigned integer datatypes. Following is the complete list of allowed datatypes.

- signed char
- short
- int
- long
- long long
- unsigned char
- unsigned short
- unsigned int
- unsigned long
- unsigned long long

C++ Modulus with Two Integers

You can find remainder using modular division operator, when two integers are divided. The datatype of the operands and returned value is given in the following code snippet.

```
int = int % int
```

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

C++ Program

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

int main() {
    int a = 13;
    int b = 5;

    int modulo = a%b;

    cout << modulo;
}
```

Output

```
3
```

C++ Modulus with Long and Unsigned Char

In this example, we shall take two operands of datatypes long and unsigned char. And perform modulus division with these values.

C++ Program

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

int main() {
    long a = 13;
    unsigned char b = 5;

    int modulo = a%b;

    cout << modulo;
}
```

Output

3

C++ Modulus with values of Not Allowed Types

We know the list of allowed datatypes for operands of modulus operator. We shall try with a not allowed datatype for one of the operands and check the output.

C++ Program

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

int main() {
    float a = 13;
    unsigned char b = 5;

    int modulo = a%b;

    cout << modulo;
}
```

Output

You will get compilation error similar to the following output.

```
d:\workspace\cpp\main.cpp: In function 'int main()':
d:\workspace\cpp\main.cpp:8:18: error: invalid operands of types 'float' and 'unsigned c
  8 |     int modulo = a%b;
    |                   ~^~
    |                   | |
    |                   | unsigned char
    |                   float
The terminal process terminated with exit code: 1
```

Chaining of Modulus Operator

You can chain Modulus Operator and perform the modular 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
```

C++ Program

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

int main() {
    int a = 13;
    int b = 5;
    int c = 2;

    int modulo = a % b % c; // a % b is 3, 3 % c is 1

    cout << modulo;
}
```

Output

```
1
```

Conclusion

In this [C++ Tutorial](#), we learned how to use C++ Modulus Operator, and the list of allowed datatypes for modulus operator in C++, with example C++ programs.

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