

运算符与表达式

- 程序中，表达式是计算求值的基本单位，由运算符和运算量（也叫操作数）和括号组成。

例如: $a+b$, x/y

- 执行表达式所规定的运算，所得的值就是**表达式的值**。
- 一个常量或者标识符是一个最简单的表达式；
- 一个表达式可以是其他运算符的操作数，例如 $a+x/y$
- **二元运算符**需要两个操作数：操作数1 运算符 操作数2
- **一元运算符**需要一个运算符。

操作数 运算符 或者 运算符 操作数

C++的运算符十分丰富，例如把赋值号(=)也作为运算符处理，C++提供了以下运算符：

- (1) 算术运算符
- (2) 关系运算符
- (3) 逻辑运算符
- (4) 位运算符
- (5) 赋值运算符 (=及其扩展赋值运算符)
- (6) 条件运算符 (?:)
- (7) 逗号运算符 (,)
- (8) 指针运算符 (*)
- (9) 引用运算符和地址运算符 (&)
- (10) 求字节数运算符 (sizeof)
- (11) 强制类型转换运算符 ((类型)或类型())
- (12) 成员运算符 (.)
- (13) 指向成员的运算符 (->)
- (14) 下标运算符 ([])
- (15) 其他 (如函数调用运算符 ())

运算符的优先级和结合性

运算符具有**优先级**和**结合性**。

- **优先级**：当表达式中有**多个运算符**时，先执行优先级高的
 $a+b*c-d/e$ 的计算顺序？

- **结合性**：当一个**操作数**左右两边的**运算符优先级相同**时的
执行顺序

- $a+b-c+d$, b左边的 ‘+’和右边的 ‘-’具有相同的优先级。

1. 算术运算符与算术表达式

➤ 基本算术运算符

$+$ $-$ $*$ $/$ (若整数相除, 结果取整)

$\%$ (取余, 操作数为整数)

➤ 优先级与结合性

先乘除, 后加减, 同级自左至右

➤ 例如有表达式 $a - b * c$, b 的左侧为减号, 右侧为乘号, 而乘号优先于减号, 因此, 相当于 $a - (b * c)$ 。

说明：

1. 两个整数相除的结果为整数，舍去小数部分。如 $5/3$ 的结果值为1， $1/2=0$ 。

但是，如果除数或被除数中有一个为负值，则舍入的方向是不固定的。例如， $-5/3$ 在有的C++系统上得到结果-1，有的C++系统则给出结果-2。多数编译系统采取“**向零取整**”的方法，即 $5/3$ 的值等于1， $-5/3$ 的值等于-1，取整后向零靠拢。

如果参加 $+$ ， $-$ ， $*$ ， $/$ 运算的两个数中有一个数为**float**型数据，则运算的结果是**double**型，因为C++在运算时对所有**float**型数据都按**double**型数据处理。

自增和自减运算符

在C和C++中，常在表达式中使用自增(++)和自减(--)运算符，他们的作用是使变量的值增1或减1，如

➤ ++i (在使用i之前，先使i的值加1，如果i的原值为3，则执行j=++i后，j的值为4)

➤ --i (在使用i之前，先使i的值减1，如果i的原值为3，则执行j=--i后，j的值为2)

➤ i++ (在使用i之后，使i的值加1，如果i的原值为3，则执行j=i++后，j的值为3，然后i变为4)

➤ i-- (在使用i之后，使i的值减1，如果i的原值为3，则执行j=i--后，j的值为3，然后i变为2)

请注意：

(1) 自增运算符(++)和自减运算符(--)只能用于变量，而不能用于常量或表达式。

(2) ++和--的结合方向是“自右至左”。

(3) 自增运算符(++)和自减运算符(--)使用十分灵活，但在很多情况下可能出现歧义性。

(4) 自增（减）运算符在C++程序中是经常见到的，常用于循环语句中，使循环变量自动加1。也用于指针变量，使指针指向下一个地址。

```
a=30; B=a++; C=++a;
```

```
B=? C=? a=?
```

```
for(int i=0;i<100;i++); 执行后, n的值是多少?
```


以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/866030142123010142>