

第3章 C++运算符及表达式

本章说明：

C++运算符是对数据进行运算的符号,参与运算的数据称为操作数或运算对象,由操作数和运算符连接而成的式子称为表达式,通过本章的学习可以掌握运算符的使用,能为后面程序的编写建立准确的表达式。

本章主要内容：

- 基本运算符与表达式
- 逻辑运算符与表达式
- 位运算符
- 条件运算符与逗号表达式
- 赋值运算符及复合赋值运算符

本章拟解决的问题：

1. 两个整数相除如何得到正确的结果?
2. 如何求一个数除以另一个数的余数?
3. 如何进行数据类型转换?
4. 如何计算一个数据的宽度?
5. 比较运算符和逻辑运算符的返回值是什么?
6. 条件表达式中的条件用什么表示?
7. 如何进行位运算?
8. 逗号表达式的整个表达式的值是哪一个?
9. = 和 == 有什么区别?
10. 复合赋值运算符怎么使用?

3.1 基本运算符与表达式

C++运算符是对数据进行运算的符号,参与运算的数据称为操作数或运算对象,由操作数运算符连接而成的式子称为表达式。

按照运算符要求操作数个数的多少,C++运算符分为以下三种:

- 单目(或一元)运算符,一般位于操作数的前面,如对 a 取负为 $-a$ 。
- 双目(或二元)运算符,一般位于两个操作数之间,如两个数 a 和 b 相加表示为 $a+b$ 。

- 三目(或三元)运算符只有一个,即为条件运算符,它含有两个字符,分别把三个操作数分开。

3.1.1 算术运算符及表达式

这类运算符包括加、减、乘、除和取余 5 种,如表 3-1 所示。算术运算符的含义与数学上相同,该类运算的操作数可以为整数、实数、字符型等。用算术运算符连接起来的式子称为算术表达式。

表 3-1 算术运算符

序 号	运 算 符	含 义	举 例	结 果
1	+	加	$9+4$	13
2	-	减	$9-4$	5
3	*	乘	$9*4$	36
4	/	除	$9/4$ 或 $9/4.$	2/2.25
5	%	取余	$9\%4$	1

说明:

算术运算符的优先顺序如图 3-1 所示。

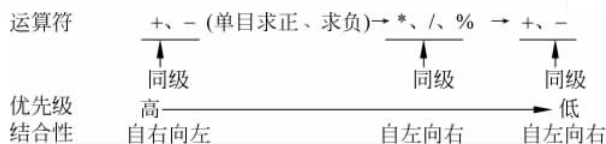


图 3-1 算术运算符的优先顺序

3.1.2 自增自减运算符

自增自减运算符主要是指 ++ 和 --, 如表 3-2 所示。

表 3-2 自增自减运算符

序 号	运 算 符	含 义	举 例	结 果
1	++	增 1	a 为 9 a++ 或 ++a	9 或 10
2	--	减 1	a 为 9 a-- 或 --a	9 或 8

说明:

- a++ 和 a-- 是先使用 a, 后加 1 或减 1。
- ++a 和 --a 是先加 1 或减 1, 后使用 a。
- a++ 等价于 $a = a + 1$ 也等价于 $a += 1$ 。
- a-- 等价于 $a = a - 1$ 也等价于 $a -= 1$ 。

3.1.3 pow 函数

除了算术运算符, 如果解决乘幂问题可以使用 pow 函数来实现, 具体格式为:

`pow(x,y)`

说明：

- `pow(x,y)`表示的是 x^y ，其中 `x,y` 可以是整数，也可以是实数。
- 该函数的返回值为双精度型。

3.1.4 强制类型转换函数

强制类型转换函数是将一个类型转换成另一个类型进行运算，具体格式为：

(类型名)(表达式)

说明：将整数 `9+4` 转换成单精度就可以写成 `(float)(9+4)`。

3.1.5 数据长度运算符

数据长度运算符也称数据宽度运算符，具体格式为：

`sizeof(数据或数据类型)`

说明：

返回值是一个数据所占用的内存空间，用字节来表示。比如字符型数据宽度为 1。

3.2 逻辑值运算符与表达式

3.2.1 关系运算符及表达式

关系运算符也称比较运算符，关系运算符共有 6 个，如表 3-3 所示，它们都是双目运算符，用来比较两个操作数的大小。由一个关系运算符连接的表达式称为关系表达式，当一个关系式成立时则计算结果为逻辑值真(1)，否则为逻辑值假(0)。

表 3-3 关系运算符

序 号	运 算 符	功 能	举 例	结 果
1	<	小于	<code>9 < 4</code>	0
2	>	大于	<code>9 > 4</code>	1
3	<=	小于等于	<code>9 <= 4</code>	0
4	>=	大于等于	<code>9 >= 4</code>	1
5	==	等于	<code>9 == 4</code>	0
6	!=	不等于	<code>9 != 4</code>	1

说明：

- 前 4 种关系运算符(>,>=,<,<=)的优先级别相同，后两种也相同。前四种高于后两种。
- 关系运算符的优先级低于算术运算符。
- 关系运算符的优先级高于赋值运算符。

- 关系运算符用 1 表示真值,用 0 表示假值。

3.2.2 逻辑运算符

逻辑运算符有三个,如表 3-4 所示,其中! 为单目运算符,&& 和|| 为双目运算符。逻辑运算的对象是逻辑值 0 或 1,若它不是一个逻辑值,则对于非 0 值首先转换为逻辑值 1,对于 0 值转换为逻辑值 0。逻辑运算的结果是一个逻辑值 1 或 0。

表 3-4 逻辑运算符

运 算 符	功 能	举 例	结 果
!	逻辑非	! 9	0
&&	逻辑与	9&&2	1
	逻辑或	9 4	1

说明:

- 逻辑运算符优先级顺序! (非)→&&(与)→|| (或),即“!”最优先。
- 逻辑运算符中的“&&”和“||”低于关系运算符,“!”高于算术运算符。
- 多个“&&”运算符,只有前一个为真,才判断下一个。
- 多个“||”运算符,只要有一个为真,就不判断下一个。

3.3 位运算符

位运算符要求操作数必须是整型、字符型和逻辑型数据,如表 3-5 所示。

表 3-5 位运算符

序 号	运 算 符	功 能	举 例	结 果
1	&	位“与”	9&4	0
2	^	位“异或”	9^4	13
3		位“或”	9 4	13
4	~	位“取反”	~9	-10
5	>>	右移	9<<4	144
6	<<	左移	9>>2	2

说明:

- 一个数按位左移(<<)多少位将通常使结果比操作数扩大了 2 的多少次幂。
- 按位右移(>>)多少位将通常使结果比操作数缩小了 2 的多少次幂。
- 按位取反(~)使结果为操作数的按位反,即 0 变 1 和 1 变 0。
- 按位与(&)使结果为两个操作数的对应二进制位的与,1 和 1 的与得 1,否则为 0。
- 按位或(|)使结果为两个操作数的对应二进制位的或,0 和 0 的或得 0,否则为 1。
- 按位异或(^)使结果为两个操作数的对应二进制位的异或,0 和 1 及 1 和 0 的异或得 1,否则为 0。

3.4 条件运算符与逗号表达式

3.4.1 条件运算符

条件运算符是 C++ 中唯一的三目运算符,其使用格式为:

`<条件表达式 1>?<表达式 2> : <表达式 3>`

说明:

- 当计算<条件表达式 1>时,值非 0(真)则计算出<表达式 2>的值,这个值就是整个表达式的值。
- 若<条件表达式 1>的值为 0(假),则计算出<表达式 3>的值,它就是整个表达式的值。

3.4.2 逗号运算符

逗号运算符是一种顺序运算符,对于分别用逗号分开的若干个表达式,每个逗号都称为逗号运算符,合起来称为逗号表达式,具体格式为:

`表达式 1, 表达式 2, …… , 表达式 n`

说明:

- `x++ , y += x, z-3` 就是一个逗号表达式,它首先计算 `x++` 的值,该计算使 `x` 增 1;接着计算 `y += x` 的值,该计算使 `y` 增加了 `x` 的值;最后计算 `z-3` 的值。
- `z-3` 的值则成为整个表达式的值,也就是后一个表达式为整个表达式的值。

3.5 赋值运算符及复合赋值运算符

3.5.1 赋值运算符

赋值运算符主要是指 = 号,具体格式为:

`变量名 = 表达式`

说明:

- 使用 = 号进行赋值, == 表示的是比较。
- 将实型数据(包括单精度、双精度)赋给整型变量时,舍弃小数部分。
- 将整型数据赋给单双精度变量,数值不变,以浮点数形式存储到变量中。
- 整型赋给字符型,把整型的 ASCII 值的字符赋给字符型。
- 把字符赋给整型,是把字符的 ASCII 值赋给整型。

3.5.2 复合赋值运算符

复合赋值运算符如表 3-6 所示。

表 3-6 复合赋值运算符

序 号	运 算 符	含 义	表 达 式	等价表达式
1	<code>+=</code>	复合赋值加	<code>n += 1</code>	<code>n = n + 1</code>
2	<code>-=</code>	复合赋值减	<code>n -= 1</code>	<code>n = n - 1</code>
3	<code>* =</code>	复合赋值乘	<code>n * = 1</code>	<code>n = n * 1</code>
4	<code>/ =</code>	复合赋值除	<code>n / = 1</code>	<code>n = n / 1</code>
5	<code>% =</code>	复合赋值取余	<code>n % = 1</code>	<code>n = n % 1</code>
6	<code><< =</code>	复合赋值左移	<code>n << = 1</code>	<code>n = n << 1</code>
7	<code>>> =</code>	复合赋值右移	<code>n >> = 1</code>	<code>n = n >> 1</code>
8	<code>&. =</code>	复合赋值按位与	<code>n &. = 1</code>	<code>n = n &. 1</code>
9	<code> =</code>	复合赋值按位或	<code>n = 1</code>	<code>n = n 1</code>
10	<code>^ =</code>	复合赋值按位异或	<code>n ^ = 1</code>	<code>n = n ^ 1</code>

说明:

- `x += y + 1` 实际上就是 `x = x + (y + 1)`。
- 使用复合运算符可以提高 C++ 程序的执行效率。

3.6 本章教学案例

3.6.1 除法运算

案例描述

输入两个整数,对两个整数进行除法运算,运行时输入 9 和 4 作为调试数据,保存程序文件名为 C++ 03-01.CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a, b;
    float c, d, e;
    cout << "请输入两个整数";
    cin >> a >> b;
    c = a / b;
    d = 9 / 4.;
    e = (float) a / b;
    cout << "c = " << c << endl;
    cout << "d = " << d << endl;
    cout << "e = " << e << endl;
}
```

程序运行结果(图 3-2)



图 3-2 C++ 03-01.CPP 运行结果

知识要点分析

- 如果两个整数相除,结果只能得到整数,要想得到小数,其中的一个数必须转换成小数(单双精度)。
- 如果是整型常量直接加小数点,如果是整型变量必须进行强制类型转换,转换成单双精度数。

3.6.2 取余运算

案例描述

输入两个整数,对两个整数进行取余运算,运行时输入 9 和 4 作为调试数据,保存程序文件名为 C++ 03-02.CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a,b,c;
    cout <<"请输入两个整数";
    cin >> a >> b;
    c = a%b;
    cout <<"两个整数相除的余数: "<<c<<endl;
}
```

程序运行结果(图 3-3)



图 3-3 C++ 03-02.CPP 运行结果

知识要点分析

- 如果两个整数直接相除,得到的是商。
- 如果得到余数就用%,运算结果是整数而不是小数。

3.6.3 求商运算

案例描述

输入两个数,对两个数进行除法运算并求商,运行时输入 9 和 4 作为调试数据,保存程序文件名为 C++ 03-03. CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a, b, c;
    cout << "请输入两个数";
    cin >> a >> b;
    c = a/b;
    cout << "两个数的商为: " << c << endl;
}
```

程序运行结果(图 3-4)



图 3-4 C++ 03-03. CPP 运行结果

知识要点分析

两个整数相除,求商,只需两个整数相除即可,切记不能转换为单双精度。

3.6.4 数位分解运算

案例描述

输入一个三位数,然后把个位、十位、百位上的数输出,运行时输入 789 作为调试数据,保存程序文件名为 C++ 03-04. CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a, gw, sw, bw;
    cout << "请输入一个三位数:";
    cin >> a;
    gw = a%10;
```

```

sw = a/10%10;
bw = a/100%10;
cout <<"个位上的数为: "<< gw << endl;
cout <<"十位上的数为: "<< sw << endl;
cout <<"百位上的数为: "<< bw << endl;
}

```

程序运行结果(图 3-5)

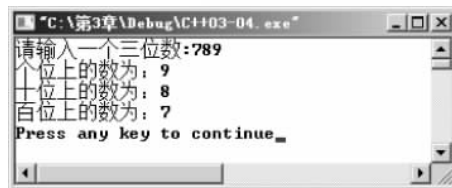


图 3-5 C++ 03-04.CPP 运行结果

知识要点分析

- 任意一个数除以 10 的余数都是个位数。
- 一个整数除以 10(不能转换为单双精度)都是舍弃个位数。

3.6.5 自增自减运算

案例描述

$i = 1$, 输出三次 $i++$ 的结果与 $++i$ 的结果, 比较有何区别, 保存程序文件名为 C++ 03-05. CPP。

案例实现

```

#include <iostream>
using namespace std;
void main()
{
    int i;
    i = 1;
    cout << i++ << endl;           //1
    cout << i++ << endl;           //2
    cout << i++ << endl;           //3
    cout << "i = " << i << endl;
    i = 1;
    cout << ++i << endl;           //2
    cout << ++i << endl;           //3
    cout << ++i << endl;           //4
    cout << "i = " << i << endl;
}

```

程序运行结果(图 3-6)



图 3-6 C++ 03-05.CPP 运行结果

知识要点分析

- $i++$ 是先用 i 然后再加 1。
- $++i$, 是先加 1 后用 i ,
- $--$ 同 $++$ 用法一样。

3.6.6 用 pow 函数计算数的次方

案例描述

编写一个程序求 2 的 5 次方, 保存程序文件名为 C++ 03-06.CPP。

案例实现

```
#include <iostream>
using namespace std;
#include <cmath>
void main()
{
    int a = 2, pf;                                //定义成单双精度可以求任意次方
    pf = pow(a, 5);
    cout << "a = " << pf << endl;
}
```

程序运行结果(图 3-7)



图 3-7 C++ 03-06.CPP 运行结果

知识要点分析

如果求 2 的平方根, 可以使用 $\text{pow}(2, 1/2.)$ 。

3.6.7 左移与右移

案例描述

将十进制 44 左移两位和右移两位的值,保存程序文件名为 C++ 03-07. CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a = 44, b, c;
    b = a >> 2;
    c = a << 2;
    cout << "b = " << b << endl;
    cout << "c = " << c << endl;
}
```

程序运行结果(图 3-8)

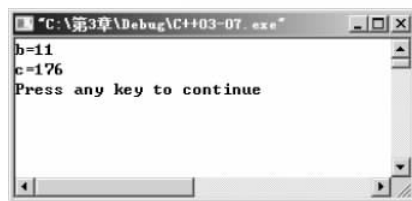


图 3-8 C++ 03-07. CPP 运行结果

知识要点分析

位移是指二进制的位移,包括左移和右移两种。

3.6.8 用条件运算符求最大值

案例描述

用条件运算符求两个数中较大的一个数,运行时输入 7 和 9 作为调试数据,保存程序文件名为 C++ 03-08. CPP。

案例实现

```
#include <iostream>
using namespace std;
void main()
{
    int a, b, max;
    cout << "输入 a, b 分别为: ";
    cin >> a >> b;
    max = a > b ? a : b;
    cout << "两个数最大的数是: " << max << endl;
}
```

程序运行结果(图 3-9)



图 3-9 C++ 03-08. CPP 运行结果

知识要点分析

- b 值为真,则得到 a 的值,否则得到 b 的值。
- 同类问题也可以通过 if 语句实现。

3.7 本章课外实验

1. 通过效果图中的数据进行位运算,保存程序文件名为 C++ 03-KS01. CPP,最终效果如图 3-10 所示。



图 3-10 C++ 03-KS01. CPP 运行结果

2. 设 $a=1, b=1, c=3$, 依次求出下列表达式 a 的值:

- $a += b + 4;$
- $a <= c - 2;$
- $a * = 3;$
- $a += b += c;$
- $a -= b = ++c + 2;$

保存程序文件名为 C++ 03-KS02. CPP,最终效果如图 3-11 所示。

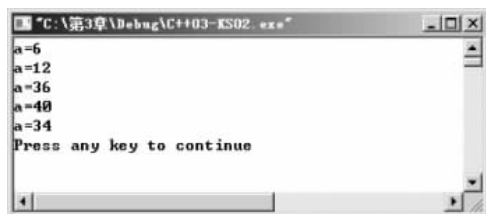


图 3-11 C++ 03-KS02. CPP 运行结果

3. 用条件运算符求 x 的绝对值,并将结果赋给 y ,程序运行时输入 -5 调试,保存程序文件名为 C++ 03-KS03.CPP,最终效果如图 3-12 所示。



图 3-12 C++ 03-KS03.CPP 运行结果

4. 设 $a=2, b=4, c=6$ 。求 $y = x = a + b, b + c$ 的值,保存程序文件名为 C++ 03-KS04.CPP,最终效果如图 3-13 所示。



图 3-13 C++ 03-KS04.CPP 运行结果

5. 利用条件运算符判断输入的一个成绩,如果成绩 ≥ 90 分的用 A 表示,60~89 分之间的用 B 表示,60 分以下的用 C 表示,然后输出这个学生的成绩和等级,保存程序文件名为 C++ 03-KS05.CPP,最终效果如图 3-14 所示。

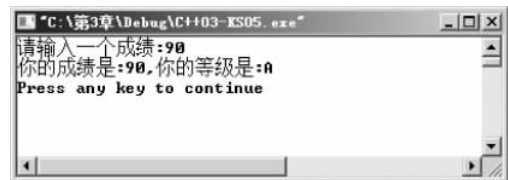


图 3-14 C++ 03-KS05.CPP 运行结果