

# 高级语言程序设计

张伯雷

[bolei.zhang@njupt.edu.cn](mailto:bolei.zhang@njupt.edu.cn)

bolei-zhang.github.io

计算机学院，软件教学中心

# 高级语言程序设计

## 第03章 运算符与表达式

## 1. 初识计算机、程序与C语言

- 计算机与存储程序
- C语言的特点
- 进制转化

## • 2.初识C源程序及其数据类型

- C语言源程序的组成结构
  - C程序的6种符号：关键字、标识符、运算符、分隔符、其他符号、数据
- 基本数据类型常量
- 基本数据类型变量

## C程序的6种符号

- 1 关键字：附录B
- 2 标识符：系统预定义、用户自定义
- 3 运算符：附录D
- 4 分隔符：空格、回车、逗号、分号
- 5 其它符号：{ }、/\* \*/、//
- 6 数据：常量

- 运算符、表达式的基本概念
- 常用运算符的运算规则、优先级和结合性
- C语言数据类型转换的方式

人们认为计算机科学是天才的艺术，但事实完全相反：只是很多人互相在对方的基础上做事，就像一面由小石头堆砌而成的墙。

——唐纳德·克努特（Donald Knuth），计算机科学家

# 3.1 什么是运算符与表达式

- 运算符 ( Operator )
  - 表示数据的运算，实现对数据的各种操作
- 运算对象 ( Operand )
  - 可以是常量、变量和**函数**
- 运算符的分类
  - 按所需运算对象 ( 操作数 ) 的个数分为三类：
    - 单目运算符 ( Unary Operator )
    - 双目运算符 ( Binary Operator )
    - 三目运算符 ( Ternary Operator )

# 3.1 什么是运算符与表达式

- 表达式 ( Expression )
  - 由运算符和运算对象组成
  - 任一个表达式都有一个确定的值，称为该表达式的值。
- 例如：
  - 3 /\*常量表达式，该表达式的值就是3\*/
  - a /\*变量表达式，该表达式的值是变量a当前的值\*/
  - $a+b*c$  /\*算术表达式，该表达式的值是算术运算的结果\*/
  - $a=10$  /\*赋值表达式，该表达式的值就是所赋的值10\*/
  - $\sin(1.2)$  /\*函数表达式，该表达式的值是弧度1.2的正弦函数值\*/

## 3.2 运算符的优先级与结合性

如何理解含有多个运算符的复杂表达式？

- 优先级 ( Precedence )
  - 级数越小，优先级越高 ( 详见附录D )
- 结合性 ( Associativity )
  - 左结合、右结合



## 3.2 运算符的优先级与结合性

- 表达式语义的理解（子表达式的划分）

- 规则1：根据运算符的优先级，优先级高的运算符先与相应的操作数构成子表达式，优先级低的运算符后与相应的操作数构成子表达式。（相当于打小括号“()”）

如： $x+y*z$  相当于  $x+(y*z)$

- 规则2：当两个运算符的优先级相同时，则根据运算符的结合性的结合方向进行计算：左结合的从左到右打“()”；右结合的从右到左打“()”。

如： $x+y-z$  相当于  $(x+y)-z$

**注意：运算符的优先级和结合性只与子表达式的划分有关，与计算次序无关！**

# 高级语言程序设计

张伯雷

[bolei.zhang@njupt.edu.cn](mailto:bolei.zhang@njupt.edu.cn)

<http://bolei-zhang.github.io>

计算机学院，软件教学中心

# 高级语言程序设计

## 第03章 运算符与表达式

## 1. 初识计算机、程序与C语言

## 2. 初识C源程序及其数据类型

- C语言源程序的组成结构
  - C程序的6种符号：关键字、标识符、运算符、分隔符、其他符号、数据
- 基本数据类型常量
- 基本数据类型变量

## 3. 运算符与表达式

- 运算符、表达式的基本概念（表达式的值）
- 优先级和结合性

# 习题



- A. 1和'1'哪个占得空间大？
- B. C语言的unsigned short的取值范围是？
- C. 十进制数346对应的八进制数是多少？

# 3.3 常用运算符



- 算术运算符
- 关系运算符
- 逻辑运算符
- 条件运算符
- 赋值及复合赋值运算符
- 逗号运算符
- 自增、自减运算符

- 算术 ( **Arithmetic** ) 运算
  - 加、减、乘、除、求余、取负数等
  - **+**、**-**、**\***、**/**、**%**等 ( **运算符** )
- 优先级： $-$  ( 取负数 )  $\Rightarrow$   $*$ 、 $/$ 、 $\%$   $\Rightarrow$   $+$ 、 $-$  ( 减法 )
- 结合性：左结合 ( 单目运算符“取负数”为右结合 )
  - 算术表达式： **$3-1+5\%4$**
  - 利用**小括号()**划分子表达式： **$((3-1)+(5\%4))$**
  - 该表达式的值：**3**
- 可利用**( )**改变表达式的语义
  - 算术表达式： **$3-(1+5)\%4$**  的值为：**1**

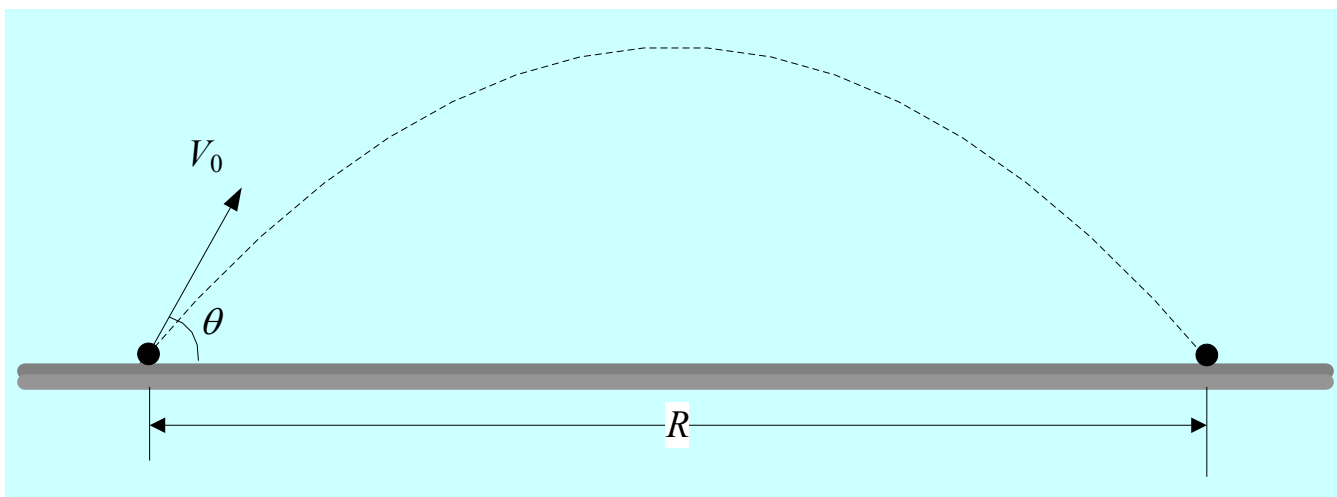
# 算术运算符的几点说明

- **整数除、实数除（务必牢记！）**
  - 如： $1/2$ 和 $1.0/2$ 的结果不同： $1/2=0$ ,  $1.0/2=0.5$
- **求余运算符%**
  - 两个操作数都必须为整型，如： $6.0\%4$ 是错误的
  - 余数的符号与被除数相同，如： $6\%(-4)=2$ ， $(-6)\%(4)=-2$ 。
- **数学函数的使用（参见表3\_2）**
  - 如： $\text{sqrt}(a*a+b*b)$ ，求a平方加b平方的算术平方根



# 算术运算符

## • 例3\_1 计算抛物运动的射程



$$R = \frac{V_0^2 \cdot \sin(2\theta)}{g}$$

### • 在VS2010环境下运行该程序并分析运行结果

思考题：若将sin函数中的参数  
“2\*theta/180.0\*pi” 改成 “2\*theta/180\*pi”  
对上例的运行结果有无影响？

# 关系运算符

- 关系 ( Relational ) 运算

- 小于、小于或等于、大于、大于或等于、等于、不等于
- $<$ 、 $<=$ 、 $>$ 、 $>=$ 、 $==$ 、 $!=$
- 运算结果为逻辑值：真 (1)、假 (0)

- 优先级： $<$ 、 $<=$ 、 $>$ 、 $>=$   $\Rightarrow$   $==$ 、 $!=$

- 结合性：左结合

- 若变量  $a=1$ ， $b=2$ ， $c=3$ ，则：

$a\%2!=0$  相当于  $(a\%2)!=0$  运行结果为：1

$a+b > b+c$  相当于  $(a+b) > (b+c)$  运行结果为：0

$a<b==b<c$  相当于  $(a<b)==(b<c)$  运行结果为：1

- $'A' < 'a'$  的结果为1，因为比较的是字符的ASCII码

注意： $==$ 和 $=$ 的区别！

# 逻辑运算符

- 逻辑 ( Logic ) 运算

- 逻辑与、逻辑或、逻辑非
- &&、 ||、 !

- 优先级&结合性 ( 见表3\_4 )

- ! 优先级高(2) , 单目运算符 , 右结合
- &&(11)优先级高于||(12) , 但低于关系和算术运算符 ; 双目运算符 , 左结合

- 运算对象

- 逻辑值 : 事实上可以是任何类型的值
- 在C语言中 : 非0相当于真 ; 0相当于假

# 逻辑运算符

## • 逻辑运算的真假值表

A 的取值↵	B 的取值↵	A&&B↵	A    B↵	!A↵
真 (非 0) ↵	真 (非 0) ↵	真 (1) ↵	真 (1) ↵	假 (0) ↵
真 (非 0) ↵	假 (0) ↵	假 (0) ↵	真 (1) ↵	假 (0) ↵
假 (0) ↵	真 (非 0) ↵	假 (0) ↵	真 (1) ↵	真 (1) ↵
假 (0) ↵	假 (0) ↵	假 (0) ↵	假 (0) ↵	真 (1) ↵

- $a < b \ \&\& \ b < c$  相当于  $(a < b) \ \&\& \ (b < c)$
- $a == b \ || \ c > d \ \&\& \ x < y$  相当于  $(a == b) \ || \ ((c > d) \ \&\& \ (x < y))$

# 逻辑运算符

- 例如：判断一个字符ch是否为小写字符的逻辑表达式为：'ch>='a' && ch<='z'
- 判断某年y是否为闰年的条件是：y能被4整除，但不能被100整除；或者y能被400整除。其逻辑表达式为：

不能写成：'a'<=ch<='z'

$((y \% 4 == 0) \&\& (y \% 100 != 0)) \parallel (y \% 400 == 0)$

## 逻辑短路现象

- 当仅通过第一操作数就能确定含逻辑运算符的表达式的结果时，第二操作数就不再计算。
- 若变量a=1，b=2，c=3，下列表达式的运算结果为：  
(a>b)&&(c=c\*2) 因a>b的结果为0，整个“&&”表达式的值就为0，发生逻辑短路，c=c\*2没运行，c值仍为3  
(a<b)|| (c=c\*2) 可同理分析

# 条件运算符

- 条件 ( **Conditional** ) 运算符

- 由两个符号 “?” 和 “:” 组成

- $\langle \text{表达式1} \rangle ? \langle \text{表达式2} \rangle : \langle \text{表达式3} \rangle$

- 优先级：低于算术、关系以及逻辑等运算符

- 结合性：右结合，C语言中**唯一的三目运算符**

- 若有两个整数变量a和b，通过条件表达式求出a、b之中较大的数的值：

$a > b ? a : b$

当 $a=2, b=1$ 时，上式的结果是2；当 $a=1, b=2$ 时，上式的结果是2。

# 赋值及复合赋值运算符

- 赋值 ( Assignment ) 运算

- $\langle \text{变量} \rangle = \langle \text{表达式} \rangle$

- 优先级：低于前面所有运算符，仅高于逗号运算符

- 结合性：右结合

- $a = a + 1$  , 相当于  $a = (a + 1)$  , 注意与数学代数式的区别。

- $a = b = c = 1$  相当于  $a = (b = (c = 1))$  , 右结合

# 赋值及复合赋值运算符

- 复合赋值 ( Combined Assignment ) 运算

- $\langle \text{变量} \rangle \langle \text{双目运算符} \rangle = \langle \text{表达式} \rangle$

- 等价于： $\langle \text{变量} \rangle = \langle \text{变量} \rangle \langle \text{双目运算符} \rangle \langle \text{表达式} \rangle$

- 算术复合赋值运算符： $+=$ 、 $\square=$ 、 $*=$ 、 $/=$ 、 $\%=$

- 优先级&结合性：同赋值运算符

- 若 $a=2$ ， $b=3$ ，计算 $a *= b+1$

- 计算过程如下：

- ( 1 ) 先将 “=” 右边整体用小括号括起来： $a *= (b+1)$

- ( 2 ) 再将复合赋值写成等价的赋值表达式： $a = a * (b+1)$

- ( 3 ) 最后进行运算： $a = 2 * (3+1) = 8$ ，即，变量 $a$ 的值为8



# 逗号运算符

- 逗号 ( Comma ) 运算符

- $\langle \text{表达式1} \rangle , \langle \text{表达式2} \rangle , \dots , \langle \text{表达式n} \rangle$

- 按顺序依次计算各表达式，整个表达式的值就是表达式n的计算结果

- 优先级&结合性：最低、左结合

- $a=b=1+2, 3*b$

- 先计算  $a=b=1+2$ ，得到  $a=b=3$ ；再计算  $3*b$ ，得到9，则该表达式的值为9，这时a和b的值都是3。

- $a=(b=1+2, 3*b)$

- 先计算逗号表达式  $(b=1+2, 3*b)$ ，有： $b=1+2=3$ ， $3*b=9$ ；再将逗号表达式的值9赋给a，则整个表达式的值就是9，这时a的值为9，b的值为3。

# 自增、自减运算符

- 自增 ( **Increment** )、自减 ( **Decrement** )
  - **++、--**
    - 使被操作的**变量**的值增加1或减少1
- 优先级：单目运算符，高于双目运算符
- 结合性：右结合
- 分为两类：
  - “前++” 和 “前--” : **++<变量>、--<变量>**
  - “后++” 和 “后--” : **<变量>++、<变量>--**

# 自增、自减运算符

## • 运算规则

- “前++”、“前--”：先对变量自增、自减1运算，再将变量更新后的值作为自增、自减表达式的值。
- “后++”、“后--”：先将变量原来的值作为自增、自减表达式的值，再对变量自增、自减1运算。

## • 若a=1，则：

- $b = ++a$ ; 相当于  $a = a + 1$ ;
- $b = a++$ ; 相当于  $b = a$ ;  $a = a + 1$ ;
- $c = (a++) + (++b)$ ; 相当于  $c = a + b + 1$ ;
- $d = -b++$ ; 相当于  $d = -b$ ;  $b = b + 1$ ;

```
int a = 4;  
printf("%d, %d, %d\n", a += 1, a++, ++a);
```

## 3.4 运算过程中的数据类型转换

- 自动类型转换 ( **Implicit Type Conversation** )
  - 表达式中的自动类型转换
  - 赋值中的自动类型转换
- 强制类型转换 ( **Explicit Type Conversation** )

**类型转换是C语言非常重要的概念！**

# 表达式中的自动类型转换

- 转换的原则：
  - 将参与运算的操作数转换成其中占用内存**字节数最大的操作数的类型**，即**数据类型的长度由低向高进行转换**，以防止计算精度的损失。
- 具体转换规则：**见图3\_2**
- 举例：
  - 'A'+32：将字符'A'转成整型65，再与整型32相加，结果为整型97。
  - 1.0/2\*3.0：将1.0/2中的整型2转换成double型，计算1.0/2.0得0.5，再乘以3.0，最后的结果为double型1.5。
  - 1/2\*3.0的计算结果为0.0，因为1/2为整除结果为0。

# 赋值中的自动类型转换

- **转换规则：**
  - 将右边表达式的值转成左边变量的类型
- **转换时的问题：**
  - 当由低长度类型向高长度类型赋值时，数据直接可以转换，不会出现数据信息丢失；但是反之，由高长度类型向低长度类型赋值时，就有可能出现数据精度下降、甚至数据溢出等问题。

**具体参见表3\_7**

# 强制类型转换

- 由程序指定的类型转换为强制类型转换

- **(<类型>) <表达式>**

- 例如：

- **int a=(int)3.14;**

**计算过程是：将3.14强制转换取整，再赋值，即a=3。**

- **(double)1/2**

**计算过程是：将1强制转换为double型，再将2自动转换为double型，最后做double型实数除法，结果为0.5。**

## \*3.5 位运算符

- **位运算 ( Bit Operation )**
  - 对字节或字内的二进制数位进行测试、抽取、设置或移位等操作
  - 6种位运算符
- **优先级&结合性：参见表3\_8**
- **位运算规则：参见表3\_9、3\_10、3\_11**





优先级	运算符	含义	运算符类型	结合方向
1	( ) [] -> .	改变优先级、函数参数表 数组元素下标 通过结构指针引用结构体成员 通过结构变量引用结构体成员		从左至右
2	! ~ ++ -- - * & (类型标识符) sizeof( )	逻辑非 按位求反 自增 1 自减 1 求负数 间接寻址运算符 取地址运算符 强制类型转换运算符 计算字节数运算符	单目运算符	从右至左
3	* / %	乘法 除法 整除求余	双目算术运算符	从左至右
4	+ -	加法 减法	双目算术运算符	从左至右
5	<< >>	左移位 右移位	双目位运算符	从左至右

优先级	运算符	含义	运算符类型	结合方向
6	< <= > >=	小于 小于等于 大于 大于等于	双目关系运算符	从左至右
7	== !=	等于 不等于	双目关系运算符	从左至右
8	&	按位与	双目位运算符	从左至右
9	^	按位异或	双目位运算符	从左至右
10		按位或	双目位运算符	从左至右
11	&&	逻辑与	双目逻辑运算符	从左至右
12		逻辑或	双目逻辑运算符	从左至右
13	? :	条件运算符	三目运算符（唯一）	从右至左
14	= +=、-=、*=、/=、%= &=、^=、 =、<<=、>>=	赋值运算符 算术复合赋值运算符 位复合赋值运算符	双目运算符	从右至左
15	,	逗号运算符	顺序求值运算符	从左至右

# 习题



- 1. 设a,b,c,d均为0，执行(m=a==b)&&(n=c!=d)后，m, n的值为多少？
- 2. 若有int x=1, y=1, 表达式(!x || y-- )的值是多少？
- 3. a = (int)3.86，a的值为多少？

- 运算符、表达式、优先级、结合性
- 七种常用运算符
- 数据类型转换
- 整数除、逻辑短路、前/后缀自增/减等

## 数据操作

基本数据类型

字符串

复合数据类型

数组

枚举、结  
构...

指针、文件

运算符

表达式

## 流程控制

程序流程控制

判断

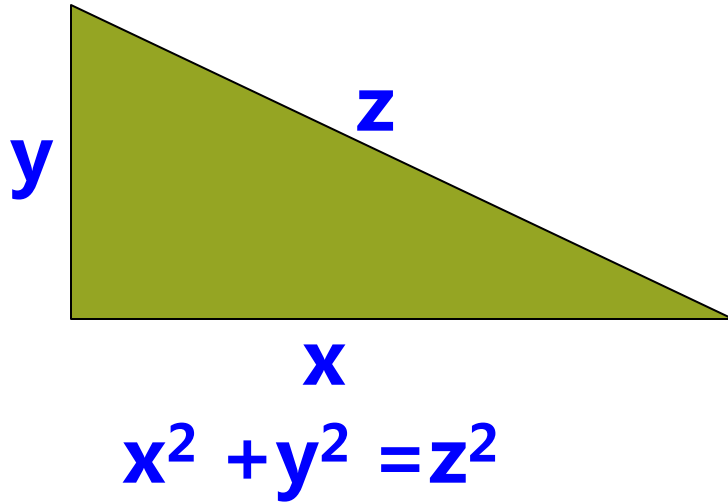
循环

函数

多文件工程



## 例3\_2 验证丢番图的规则



- 找两个正整数 $a$ 和 $b$ ，若 $2ab$ 为完全平方数，则勾股定理方程的解为：

$$x = a + \sqrt{2ab},$$

$$y = b + \sqrt{2ab},$$

$$z = a + b + \sqrt{2ab}$$

- 在VS2010环境下运行该程序并分析运行结果

思考题：若将代码第11行的强制类型转换“(int)”删除，重新编译、运行该程序结果会怎样？