高等职业技术教材

C语言程序设计基础

姜武中　主编

2C-43

中国商业出版社

TP312C-43
J47

高等职业技术教材

C语言程序设计基础

主　编　姜武中
副主编　赵钟元
　　刘景春

中国商业出版社

图书在版编目(CIP)数据

C 语言程序设计基础/姜武中主编．—北京：中国商业出版社，2000.5
ISBN 7-5044-3932-0

Ⅰ.C… Ⅱ.姜… Ⅲ.C 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2000)第 27069 号

责任编辑：刘树林

中国商业出版社出版发行
(100053 北京广安门内报国寺 1 号)
新华书店总店北京发行所经销
北京印刷集团有限责任公司印刷二厂印刷

*

787×1092 毫米 16 开 17 印张 384 千字
2000 年 5 月第 1 版 2000 年 5 月北京第 1 次印刷
定价：24.00 元

* * * *

(如有印装质量问题可更换)

前　言

有比较才能有鉴别。自20世纪70年代我有幸第一次接触AGOL60语言以来，曾先后学习了BASIC、COBOL、FORTRAN、PASCAL、C等许多计算机语言。虽然每种语言都各有所长，但在我看来最独具匠心出类拔萃的要数C语言。它不仅博采众长，很好地继承了其它计算机语言的诸多优点，而且它还标新立异，巧妙地推出了许多新思路、新概念、新方法，使人耳目一新悟性疾增。它拥有丰富的数据类型和表达式，因此它的运算功能十分强大。它允许各种不同类型的数据之间同时进行数值和逻辑运算，因此它的数据处理非常灵活。它把其它语言中普遍采用的子程序、过程、函数等多层程序结构统一归集到函数这一单一模式，因此程序结构清晰明了井然有序。程序的函数化处理使数据的输入输出格式统一、操作简便，同时使文件的操作手段也殊途同归趋向一致。函数之间相对独立互不制约，却彼此间自由调用，因此程序模块化的程度很高。它的编译预处理功能，使源程序的书写变得简单，且易于扩充程序规模和控制编译范围。它给数组名冠以地址常量的合法外衣，为其参与运算提供了根据。它又把两维数组按一维化处理，使数组的操作变得更加得心应手。它既可以按直接寻址方式访问内存数据，又可以按间接寻址方式访问内存数据，因此为指针参与表达式运算创造了条件，从而极大地增强了表达式的运算功能，以至于在其它语言中无法完成的功能，在C语言却能得以顺利实现。它具有位处理功能，因此把数据的操作范围由原来的一个存储单元可以缩小到其中的每一个二进位。它把不同类型的若干数据集中定义成一个特殊的数据类型，因此可以像数据库一样便于描述某一事物的各种不同属性。它允许将所有标准的数据类型随意更名为一个新的类型，因此程序的可移植性较好。它打破了其它语言按记录管理文件的常规模式，采用了以字节为基本单位的流式文件，因此文件操作非常规范、简单易行。它有强大的系统函数支持，且呼之即出，无须用很大的精力也能编写出较强功能的程序。

C语言的上述这些特点概括起来就是，它功能丰富、表达能力强，使用灵活方便、应用面广、目标效率高、可移植性好，既具有高级语言的优点，又具有汇编语言的许多特点，而特别适合于编写系统软件。以往操作系统等系统管理程序绝大多数是用汇编语言编写的，可是自从有了C语言之后，这些程序可以不用汇编语言编写了。汇编语言虽然目标程序的效率很高，但编写起来却比较困难，比起高级语言麻烦得多。因此，C语言的应用前景十分广阔，据说就是现在最流行的WINDOWS操作系统也是用C语言编写的。

强大的功能往往是以复杂的结构为代价的。正是由于C语言具有与众不同的特点，因此它的语法结构也相对复杂，掌握起来有一定的难度，所以有一本好的教材是至关重要的。现在

多数高等学校,已经把C语言作为学习计算机语言的首选课程。因此与之相配套的教材亦如同雨后春笋竞相推出。然而,这些教材绝大多数都是面向本科以上学历的读者的,很少发现针对大专和高职学生,甚至于自学读者为对象的。于是我们按学科组的安排,为这部分读者编写一本既通俗易懂,又包容几乎所有基本概念的C语言教材。

学习计算机语言的关键是如何掌握它的基本语法要领。语法通过了,剩下的只是一个算法问题。如果这两者同时兼顾,势必会造成眉毛胡子一把抓顾此失彼的尴尬局面。为此,我们编写这本教材的整个过程中始终贯穿"语法"这一主线,一切叙述和举例部分都是紧紧围绕着"语法"这一主线而设计的。书中叙述力图详尽,举例突出针对性,每个程序都有思路分析和算法说明,并配有相当数量的图表,每条关键语句也都加了注释。

此书的章节安排,是根据初学者的特点精心考虑的。按照由浅入深循序渐进的思路,将语言中的难点部分化整为零分散到各个章节,每一章都引入新概念,采用分散出击各个击破的战术思想,目的是使读者,尤其是初学者学起来不会感到很困难。

C语言的内容十分丰富,发展也相当迅速,它已经达到C++、Visual C等较高的层面。我们不可能在一本书里把C语言所有的内容和盘托出。我们编写此书的初衷是,将此书作为学习C语言的基础教程,系统介绍C语言最基本的知识,使广大读者今后更进一步深入学习奠定基石。

编写此书的过程中,我们得到陈学庸教授的鼎力支持和帮助,为此,谨向他表示由衷的谢意。

本书中的所有程序是在Turbo C上通过的。

全书共分十二章,前六章由赵钟元和刘景春同志编写,后六章由姜武中编写。姜武中任主编并负责全书统稿和总纂,赵钟元、刘景春任副主编。

由于编者的水平有限且经验不足,书中如有不足甚至错误,恳请广大读者批评指正。

编者

2000年2月

此为试读,需要完整PDF请访问: www.ertongbook.com

第 1 章

C 语言概述

§1.1 C语言的由来和历史背景

C 语言是 1972 年前后美国贝尔实验室开发出来的一种计算机程序设计语言。C 语言的由来与 UNIX 操作系统密切相关。UNIX 操作系统是贝尔实验室在 1969 年开发成功的，那时它是用汇编语言编写的。由于汇编语言可读性差又不易移植，因此他们决定开发一种能够更有效地描述 UNIX 操作系统的一种高级语言。因为只有高级语言才能克服汇编语言所固有的上述弊病。1970 年，贝尔实验室在英国剑桥大学从 1967 年开发的 BCPL 语言的基础上，首先开发了 B 语言，并用 B 语言重新编写了 UNIX 操作系统。但是，由于 B 语言设计上过份依赖于计算机硬件，且其功能又比较简单，所以他们从 1971 年开始又着手开发一种新语言。他们取了 BCPL 语言的第二个字符，将此语言命名为 C 语言，这便是 C 语言的由来。

1972 至 1973 年，贝尔实验室用 C 语言再次编写了 UNIX 操作系统，并获得成功。由此可见，最初的 C 语言是为了描述和实现 UNIX 操作系统并提供一种工作语言而设计的。从某种意义上说，没有 C 语言，就没有 UNIX 操作系统今天的巨大成功。同样，没有 UNIX 操作系统，C 语言也不会有现在这样的发展。两者可以说是一对孪生兄弟，在彼此的发展过程中一直是相辅相成的。最初的 C 语言保持了 B 语言所拥有的比较精炼且和硬件关系紧密等诸多优点，同时又克服了 B 语言功能过于简单、无数据类型等缺点。它把设计的着重点放在短小精悍上，力图避免大而全和面面俱到，用一些简单的程序单元共同构筑复杂的程序，以实现处理复杂问题的能力。

后来，C 语言一改初衷，在将它的服务方向不断拓宽的同时，经多次修改，自身的功能得到很大的加强。1975 年贝尔实验室推出了 UNIX 第六版之后，C 语言的突出优点引起人们的普遍注意。到 1978 年，C 语言已先后移植到大、中、小及微型机上，且已经完全独立于 UNIX 和 PDP 等操作系统了。

1978 年，美国两位科学家 K&R 氏以 UNIX 第七版中的 C 编译系统为基础，出版了著名的《C 程序语言》一本书。此书中介绍的 C 语言便成为后来广泛使用的 C 语言版本的基础，被称为标准 C。1983 年，美国标准化协会 ANSI，集 C 语言问世以来各种版本之精华，制定出一个新的标准，称为 ANSI C。1987 年，该协会又进一步修改了此标准，推出了称为 87 ANSI C 的更新标准。目前流行的 C 编译系统都是以它为基础的。

现在C语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

§1.2 C语言的特点

C语言归纳起来有以下几个特点：

1. 语言简洁、紧凑，使用方便、灵活。C语言一共只有32个关键字和9个控制语句，程序书写也不像其它语言那样受诸多限制，可以随心所欲，自由方便。一般说来，编制同样功能的程序，C语言的源程序较其它语言短小。C程序主要用小写字母来书写。

2. 运算符号多。C语言一共有34种运算符号。它将传统意义上的一些语句和程序单元的功能也归结到运算符当中来，因此它的运算类型非常丰富，表达式也极其多样。灵活使用这些运算符，可以完成许多其它高级语言中难以实现的运算或操作，比如地址和位运算等。

3. 数据结构丰富。C语言在拥有整型、实型、字符型等基本数据结构的同时，还能在这些基本结构的基础上构造出更复杂的结构类型。这些复杂的数据结构，可以有效地将不同类型的数据组合成一个整体。

4. 具有结构化的控制语句。C语言不仅拥有功能很强的分支和循环控制语句，而且将传统的子程序、过程等程序单元均统一到函数单元来处理，且以单一的函数模式组成程序模块。所以，它的结构化水平很高，可以满足现代编程风格的要求。

5. 编译后的目标代码质量很高。目标代码的质量好坏直接影响着系统开销和程序的运行效率。完成相同功能的程序，C语言较汇编语言效率仅低10%～20%。

6. 可移植性好。一台机器上编制的C程序，拿到其它机器上一般都能通过。这一点其它语言不容易做到。

7. 语法要求不严，编程自由度大。用C语言编程时出现语法错误的概率相对较低。程序的正确与否主要取决于编程人员对C语言的熟练程度，因此对初学者来说可能稍难一点。

8. 能做位运算，还可以进行硬件操作。C语言具有汇编语言的特点，它可以访问系统存储器和接口的物理地址，所以能对系统硬件进行直接控制。C语言不仅具有高级语言的特点，能做科学计算，而且还具有低级语言的特点，也可做实时控制。所以，有人称它为“中级语言”，最适于编制系统软件。

习 题

1. C语言有哪些主要特点？其主要用途是什么？
2. 将C语言和你以前学过的其它高级语言做一下比较。

第 2 章

基本数据类型

数据是计算机程序处理的主要对象，程序中的每项数据不是常量便是变量。常量和变量的最大区别是，在整个程序运行当中常量是始终不变的，而变量则可以改变。数据均具有类型，不同类型数据的处理方法也不同。在 C 语言中有整型、实型和字符型三种基本数据类型。

§2.1 整型

2.1.1 整型常量

C 语言中的整型常量有四种表达方式，即十进制、八进制、十六进制和长整型整数。

十进制数：它的表达方式与我们的习惯写法完全相同。

如：23，－107，0

八进制数：它的表达方式是八进制整数前面加零。

如：0176，032，0251

十六进制数：它的表达方式是十六进制整数前面加零 x。

如：0xa3，0x98，0x7f

长整型整数：它的表达方式是十进制整数后面加 L 或 l。

如：123L，52l，37l

长整型整数只能给长整型变量赋值。

2.1.2 整型变量

变量是用来存放数据的。C 语言中的整型变量共分五种，它们是：基本整型、短整型、长整型、无符号整型、无符号长整型。它们的英文关键字分别是：

基本整型：int

短整型：short int 或 short

长整型：long int 或 long

无符号整型：unsigned int 或 unsigned

无符号长整型：unsigned long

其中 int 、short 、long 型变量所存数据为有符号数，一般按补码形式存储；而 unsigned 型变

量存的是无符号数,即它存的是数据的绝对值(正数)。

无论是什么变量,变量必须是先定义后才能使用。因为源程序中的变量最终要落实到存储器当中来,而编译系统给变量分配存储单元,完全是依据程序中的变量定义。变量如果不事先定义,存储器中就没有它的位置,所以更谈不上什么使用了。

变量的定义包括两个内容:一是给变量命名;一是给变量指定类型。

变量的命名规则是:必须以英文字母或下划线打头,且只由英文字母(汉字也可)、下划线和数字组成,长度一般不超过八个字符位(一个汉字占两个字符位)。

变量的定义是通过一条语句来实现的,它的一般格式是:

〈类型〉变量名序列;

例如:
```
int i,j,m,l;
short n,c;
long a,b;
unsigned d;
```

例中 int、short、long、unsigned 为整型变量的类型,i,j,n,a,d 为变量名。一次定义多个变量时,变量名之间必须用逗号隔开。每行最后的分号是一条 C 语句的结束标志。编译系统在给整型变量分配存储单元时,是根据不同的类型分配不同的长度。具体的分配规模是:

int 型	2 个连续字节
short 型	同上(short 和 int 型在 IBM 系列机上是相同的)
long 型	4 个连续字节
unsigned 型	2 个连续字节
unsigned long 型	4 个连续字节

至于分配哪个地址单元,完全由编译系统根据存储器的实际情况自行决定。下面以 int 和 unsigned 型为例,介绍有符号数和无符号数在存储器中的存储形式。

```
int  i;
i=4;
```

上面两条语句中第一条语句定义了 int 型变量 i,第二条语句将整数 4 赋值给变量 i。因为 i 是 int 型变量,所以编译系统给它分配 2 个字节。又因为 i 是有符号数,所以将 4 的补码存入 i 所对应的存储单元。4 的原码为 0000000000000100。其中最高位为符号位,0 表示该数为正数。因为 4 是正数,所以它的补码等于原码,故变量 i 中整数 4 的存储形式为:

i

00000000	00000100

如果赋给变量 i 的整数不是 4 而是 -4,那么因为 -4 的原码为 1000000000000100,根据负数变补的规则,将 -4 的原码进行如下变换:符号位 1(代表负数)保持不变,其余的位取反后最低位又加 1,如此所得的值便是 -4 的补码。故变量 i 中整数 -4 的存储形式为:

i

11111111	11111100

long 型变量中有符号数的存储形式也是如此，它们跟 int 型变量的区别仅仅在于数据长度的不同。至于无符号数的存储形式，我们再看下面这个例子。

```
unsigned j;
j= 65535;
```

例中 j 是无符号整型变量，故编译系统给它分配 2 个字节的存储单元，又因为它是无符号整型变量，故存入存储单元的是 65535 的绝对值，无符号位。其具体形式如下：

j

11111111	11111111

整型变量的数据存储范围是：

int、short 型 —32768～32767 (1000000000000000～0111111111111111)

long 型 —2147483648～2147483647 (同上，但长度为 32 位)

unsigned 型 0～65535 (0000000000000000～1111111111111111)

unsigned long 型 0～4294967295 (同上，但长度为 32 位)

§2.2 实型

2.2.1 实型常量

实型数据又称浮点数。实型常量有以下两种表示形式：

1. 十进制小数形式。如：0.35，4.5，78.0，0.0，－18.7，.69 等。注：小数点不可缺。

2. 指数型式。如：－32e2，1.4e－4，0.5e－3 等。它们分别等于 -32×10^{2}，1.4×10^{-4}，0.5×10^{-3}(注：e 可以大小写，但不可缺；e 前面必须有数据，整数小数均可；e 后面一定要为整数)。

实型常量只有表示方式的不同，而没有类型之分。

2.2.2 实型变量

实型变量有两种类型：单精度 float 型和双精度 double 型。实型变量所存的数据均为有符号数。它的数据存储方式也与整型变量大不相同，通常用浮点格式。因此，它的数据存储范围比整型变量大得多。

float 型为：$10^{-38}\sim10^{38}$

double 型为：$10^{-308}\sim10^{308}$

double 型不仅数据范围比 float 型大而且它的运算精度也比 float 型高。它们的有效数字长度分别是：double 型 15～16 位；float 型 7 位。比如：

变量 a 定义成 float 型且给它赋值 222222.222，如：

```
float a;
```

a=222222.222;

结果变量a实际获得的值却为222222.2(7位),后两位小数丢失。如果将a定义成double型,则a变量准确无误地将222222.222全部接收。如:

```
double a ;
a= 222222.222;
```

再比如,定义三个实型变量a、b、c,且将a+b的和送c。若a、b、c为float型时,如:

```
float a,b,c;
a=111111.111;
b=222222.222;
c= a+b;
```

运算后变量c的结果不是333333.333,而是333333.328125,只有前7个有效位才是精确的。而如果将变量定义成double型时,如:

```
double a,b,c;
a=1111111111111.111111111;
b=2222222222222.222222222;
c=a+b;
```

运算后的c的结果为3333333333333.333010 ,前16位是精确的。很显然,double型比float型运算精度要高得多。

凡属变量不管它是什么类型,都得要先定义后才能使用,这一点实型变量也不例外。实型变量的定义格式为:

```
float f1,f2,f3;
double d1,d2,d3;
```

式中,上一行为单精度实型变量的定义,其中float为类型符,f1、f2、f3为变量名。变量的命名和被定义的变量数目原则上是随意的。下一行是双精度实型变量定义,其含义与上一行是基本相同的,只是d1、d2、d3为双精度变量而已。

§2.3 字符型

2.3.1 字符常量

C字符常量也有两种表示形式:一种为普通型;一种为转移型。普通型一般用来表示一个可显ASCII码字符。一个可显ASCII码字符用单引号把它括起来便成为一个字符常量。如:'a','B','5',' '等。在这里空格实际上也是可显字符。

转移字符多半为不可显示的控制字符,它主要是用来控制输出格式。转移字符表示方式一般是反斜杠后面加一个规定字符。见表2-1。

此为试读,需要完整PDF请访问: www.ertongbook.com

表 2—1

功　能	字　符　形　式
换行	\n
横向跳格	\t
竖向跳格	\v
退格	\b
回车	\r
走纸换页	\f
一个反斜杠字符	\\
一个单引号字符	\'
一个 ASCII 码字符的八进制表示(d 最大三位)	\ddd
一个 ASCII 码字符的十六进制表示(h 最大二位)	\xhh

无论是普通字符还是转移字符,它们都是用来表示一个 ASCII 码字符的,而每一个 ASCII 码字符,在机器中是用 8 位二进制整数(0～255)表示的,故它在存储器中只占一个字节。例如:

字符常量'a'的二进制 ASCII 码为 01100001,即十进制整数 97(见附录Ⅰ)

字符常量'\n'(转移字符)的二进制 ASCII 码为 00001010,即十进制整数 10(见附录Ⅰ)。

2.3.2　字符变量

字符变量是用来存放字符常量的,所以它一次只能接收一个字符(二进制 ASCII 码)。

字符变量的定义格式为:

```
char a,b,c;
```

其中 char 为类型符,a、b、c 为字符变量名。

下面举一个字符变量中存放字符常量的例子:

```
char c1;
c1 = 'a';
```

根据变量定义,编译系统给字符变量 c1 分配一个字节的存储单元。通过赋值语句将字符常量'a'的 ASCII 码 97 存入 c1 存储单元。

c1

01100001

2.3.3　字符串常量

字符串常量,顾名思义,就是一个或若干字符的链集。字符串常量的表示方法是:将一串字符用双引号括起来即可。如:

"CHINA", "Beijing","c"等。

字符常量一般是一个字符组成的,而字符串常量通常为多个字符。就是一个字符它们之间也是有区别的:比如 A 字符,当它是字符常量时用'A'来表示,当它为字符串常量时则以"A"来表示。另外,它们的存储方式也不同:字符常量一般存入字符变量中,而字符串常量则

只能存入一个字符数组中。它们之间的最大区别是:字符串常量存入字符数组时,串的末尾系统自动追加一个控制符'\0'(不可显示),作为此字符串的结束标志,而字符常量则没有。如"ABC",在字符数组中的存储形式如图 2-1 所示:

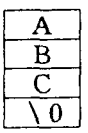

图 2—1

C语言对英文大小写字符是加以区别的,因此,字串"abc"和"ABC"是两个不同的字符串常量。

§2.4 变量的初始化

变量的初始化,指的是定义变量的同时给变量赋以初始值的功能。请看下面一个例子:

```
int i,j,k=3;
float f,g=9.8;
char c1=c2='d',e2,c3;
```

这是由三条变量定义语句组成的一段程序。这里的每条语句,都是在定义变量的同时又给其中的一个或几个变量赋初始值,即 k、g、c1、c2 四个变量被初始化。它在功能上等价于下面一段程序:

```
int i,j,k;
k=3;
float f,g;
g=9.8;
char c1,c2,c3;
c1='d';
c2='d';
```

但此程序并没有对任何变量进行初始化,其中 k=3;g=9.8;c1='d';c2='d';为四条赋值语句,并不是变量初始化。

变量的定义语句通常放在程序的开头。

习 题

1. C语言为什么规定对所有的变量要"先定义,后使用"?
2. 请将下面各数用八进制和十六进制数表示。

   (1)9 (2)34 (3)97 (4)-637

   (5)-212 (6)2348 (7)-31654 (8)27003
3. 写出不同类型的数据在存储器中的存储形式。

	37	-6	9
int(16 位)			
long(32 位)			
unsigned(16 位)			
unsigned long(32 位)			

4. 确定下列字符常量和字符串常量的长度(以字节为单位)。

(1)'a'　(2)"b"　(3)"\023\19,n"　(4)"\\012m"

(5)"\'\x2ak"　(6)"abc\n"　(7)'\b'

5. 下列变量定义中哪些是不正确的?

(1)character a1,a2,a3;

(2)char a1,int i;

(3)a1,a2,a3;char;

(4)long int b1,b2,b3;

(5)double int f1,f2;

(6)float if;

(7)unsigned long k1,k2;

6. 变量的命名规则是什么?

7. 字符常量与字符串常量有何区别? 如何区分 long 型常量与其它整型常量? 实型常量有无 float 和 double 之分?

第 3 章

C 表达式

常量、变量、函数等运算对象之间用运算符号相连并符合 C 语法规则的式子叫 C 表达式。独立的常量、变量、函数也是表达式。

运算符号是表达式的重要成分。运算符号具有两个属性:一是优先级别;一是结合方向。进行表达式运算时,我们首先应考虑运算符号的优先级别。级别高的先做,级别低的后做。级别相同时还要考虑它的结合方向。因为值得注意的是,C 运算符中有不少是与我们习惯的自左向右的结合方向相反,是为自右至左地结合。

§3.1 算术表达式

运算对象之间用算术运算符相连,而且其值为算术值的表达式称作算术表达式。算术运算符共有八种。它们是:

+	加法运算符
-	减法或负数运算符
*	乘法运算符
/	除法运算符
%	求余运算符
++	增 1 运算符
--	减 1 运算符

它们的优先次序是:++,--

*,/,%

+,-

说明:(1)两个整数相除后的商一定是整数。如:5/3=1(不舍入)

(2)进行求余运算的两个数一定为整数。如:7%4=3;而 3.4%2 则无意义。

算术表达式举例:a/b+c-2.7*3-7%2

3.1.1 增 1 减 1 运算

增 1 减 1 运算符为一目运算符,它只能跟一个变量相结合。如:++i、i++、--j、j--。对常量不能做++、--运算,即 5++是错误的。

++i 和 --i 的运算功能是,先对变量 i 的值增 1 或减 1,即 i=i+1 或 i=i-1,然后用增 1 或减 1 后的值来参与表达式的运算。例如:

设变量 i 的值为 2:那么表达式 (++i)+3 的值是 6,而变量 i 的值是 3;表达式(--i)+3 的值是 4,而变量 i 的值是 1。

++i 和 --i 的运算功能可以概括为:"先自增(自减)后使用"。自增(自减)的是变量 i 的值,而参与表达式运算的是它的使用值。在这里使用值恰好等于变量自增(自减)后的值。

再看 i++ 和 i-- 的功能。其功能是先用 i 变量的原值作为使用值参与表达式运算,然后对变量 i 增 1 或减 1。此功能可概括为:"先使用后自增(自减)"。例如:

设变量 i 的值为 2:那么表达式(i++)+3 的值是 5,变量 i 的值是 3;表达式(i--)+3 的值是 5,变量 i 的值是 1。

++和 -- 的优先级是相同的,它们的结合方向为自右到左。如:

-i++ = -(i++)

两个运算符相连时,它们的结合方向为自左到右。如:

i+++j=(i++)+j

【例 1】设 j=3,计算下列表达式和变量的值。

(++j)+(++j)+(++j)

第一个括弧中使用值为 4,变量值为 4;第二个括弧中使用值为 5,变量值为 5;第三个括弧中使用值为 6,变量值为 6。所以,整个表达式的值是 4+5+6=15,而变量 j 的值是 6。

【例 2】设 j=3,计算下列表达式和变量的值。

(j++)+(j++)+(j++)

第一个括弧中使用值是 3,变量值是 4;第二个括弧中使用值是 4,变量值是 5;第三个括弧中使用值是 5,变量值是 6。因此,整个表达式的值是 3+4+5=12,而变量 j 的值为 6。

3.1.2 强制类型转换

可以利用强制类型转换运算符,将表达式中运算对象的类型转换成所需的类型。

强制类型转换运算符是:

(类型符)表达式。

例如: (double)a　　a 强制转换成 double 型,a 原来不是 double 型;

(int)(x+y)　　x+y 强制转换成 int 型;

(float)(5%3)　　5%3 强制转换成 float 型。

强制类型转换时,被转换类型的表达式应用括号括起来。若对表达式 x+y 进行强制类型转换,应写成:

(int)(x+y)

不应写成:(int)x+y

后者只是对 x 进行转换。

强制类型转换并不改变变量原来的类型。只是改变表达式中的变量类型。即表达式中的

变量改变类型后，其值存入临时开辟的中间变量中，而此中间变量的值才参与表达式运算。

【例 1】float x;

```
int i;
x= 4.7;
i=(int)x;
```

程序运行后 x 的值仍然是 4.7，而 i 的值却是 4。

一般来说，不同类型的数进行混合运算时系统自动完成类型统一。如：

```
3+6.5
```

系统自动将 3 和 6.5 转换成 double 型。那么，强制类型转换有什么用处呢？有时系统自动转换达不到预期的目的，此时就得进行强制性类型转换。

【例 2】%运算要求运算对象为两个整数。如果 x 为 float 型，那么写成

```
x%3
```

则出错。而把它写成

```
(int)x%3
```

就合法了。

在这里，强制类型转换符的运算级别要比%高，而且比其它运算符号都要高。

3.1.3 不同类型数据间的混合运算

C 语言允许不同类型的数据之间进行混合运算。它的运算规则是：

1. 表达式中所有 float 型数据将转换成 double 型；
2. 表达式中所有 char 和 short 型数据将转换成 int 型；
3. 转换后相同类型的数据可以直接进行运算；
4. 完成上述转换后，剩下的类型则按以下优先次序进行转换；

(低)int → unsigned → long → double (高)

5. 最终将类型统一到表达式中的最高级别。

【例】

```
int i;
float f,g;
long e;
double d,g;
g=5 + 'b' + i/f - d*e;
```

根据规则第一条，将 f 转换成 double 型；根据第二条将字符常量'b'转换成 int 型(98)；根据第三条 5 和 98 直接相加得 103；根据第四条 103、i、e 均转换成 double 型。至此，表达式中各运算量的类型全部统一成 double 型，据此可以进行直接运算，最终将 double 型表达式的值赋给左面的变量 g。

此为试读，需要完整PDF请访问：www.ertongbook.com