



语言程序设计

学习指导、实验指导与

课程设计

盛夕清 赵阳 林科学 徐大华 编著



中国水利水电出版社
www.waterpub.com.cn



语言程序设计

学习指导、实验指导与

课程设计

盛夕清 赵阳 林科学 徐大华 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是为学习 C 语言程序设计, 结合计算机等级考试二级 C 语言大纲而编写的。全书共分三篇: 第一篇是 C 语言程序设计学习指导, 包括知识点回顾、典型例题解析和补充习题; 第二篇是 C 语言程序设计实验指导, 其中十个精编实验涵盖了 C 语言的大部分内容; 第三篇是 C 语言程序设计课程设计, 包括了若干个精心设计的课程设计题目。本书内容丰富、概念清晰、实用性强, 是学习 C 语言的一本好的参考书。

本书可作为 C 语言程序设计初学者的参考书, 也可以作为大学生计算机二级考试的辅助教材。

图书在版编目 (C I P) 数据

C 语言程序设计学习指导、实验指导与课程设计/盛夕清等编著. —北京: 中国水利水电出版社, 2006
ISBN 7-5084-3633-4

I. C... II. 盛... III. C 语言—程序设计—自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 018407 号

书 名	C 语言程序设计学习指导、实验指导与课程设计
作 者	盛夕清 赵阳 林科学 徐大华 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京中科洁卡科技有限公司
印 刷	北京市兴怀印刷厂
规 格	787mm×1092mm 16 开本 15 印张 356 千字
版 次	2006 年 3 月第 1 版 2006 年 3 月第 1 次印刷
印 数	0001—5000 册
定 价	24.50 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前言

本书是在C语言二级考试大纲及历年考试试卷进行充分分析研究的基础上，为学习C语言程序设计、参加C语言二级考试而编写的。

本书分为三篇。第一篇为C语言程序设计学习指导。该篇结合C语言教材的章节进行编写，每章分知识点回顾、典型例题解析和补充习题三部分。知识点回顾部分是对C语言教材中各章涉及的知识点进行总结和回顾；典型例题解析部分是围绕重点难点精选了一些灵活性强、容易失分的例题，并对其进行精心解析；补充习题部分是精选与二级C语言程序设计试题题型一致的补充练习题，以弥补教材中习题的不足。

第二篇是C语言程序设计实验指导。十个精编实验涵盖了C语言程序设计的大部分内容，通过对实验教学认真严格的组织，可以提高学生编程和调试程序的能力。

第三篇是C语言程序设计课程设计。包括了若干个精心编写的课程设计题目，目的是通过课程设计的综合训练培养学生实际分析问题、编程和动手能力。

本书对学习C语言程序设计中的重点、难点进行了详细的讲解，精心编制的实验内容及课程设计可以保证学生编程技能的培养。本书可以作为C语言程序设计初学者理想的参考读物，也可以作为大学生计算机二级考试的辅助教材。

本书由盛夕清、赵阳、林科学、徐大华编著。另外，参加本书编写及资料收集工作的还有杨建平、陆明洲、邹修国、蒋波、魏权华、宋瑞琨、牟岚、赵红梅等。在编写过程中，我们虽然力求精益求精，但书中难免存在考虑不周及疏漏之处，热忱欢迎广大读者批评和指正。

作者
2006年2月

目录

前言

第一篇 C 语言程序设计学习指导

第一章 程序设计基础	1
1.1 知识点回顾.....	1
1.2 典型例题解析.....	3
1.3 补充习题.....	5
第二章 C 语言数据类型及表达式	7
2.1 知识点回顾.....	7
2.2 典型例题解析.....	12
2.3 补充习题.....	15
第三章 顺序结构程序设计	19
3.1 知识点回顾.....	19
3.2 典型例题解析.....	21
3.3 补充习题.....	25
第四章 选择（分支）结构程序设计	31
4.1 知识点回顾.....	31
4.2 典型例题解析.....	33
4.3 补充习题.....	39
第五章 循环结构程序设计	48
5.1 知识点回顾.....	48
5.2 典型例题解析.....	51
5.3 补充习题.....	58
第六章 数组	67
6.1 知识点回顾.....	67
6.2 典型例题解析.....	71
6.3 补充习题.....	80
第七章 函数	88
7.1 知识点回顾.....	88

7.2	典型例题解析	97
7.3	补充习题	103
第八章	编译预处理	110
8.1	知识点回顾	110
8.2	典型例题解析	112
8.3	补充习题	116
第九章	指针	122
9.1	知识点回顾	122
9.2	典型例题解析	133
9.3	补充习题	141
第十章	结构体与共用体	148
10.1	知识点回顾	148
10.2	典型例题解析	155
10.3	补充习题	162
第十一章	位运算	169
11.1	知识点回顾	169
11.2	典型例题解析	170
11.3	补充习题	171
第十二章	文件	173
12.1	知识点回顾	173
12.2	典型例题解析	179
12.3	补充习题	183

第二篇 C 语言程序设计实验指导

第一章	上机实验	188
实验一	C 语言程序初步	188
实验二	C 语言基础知识	189
实验三	顺序结构程序设计	189
实验四	选择结构程序设计	190
实验五	循环结构程序设计	193
实验六	数组	194
实验七	函数	196
实验八	指针	197
实验九	结构体与共用体	198
实验十	文件操作	199
第二章	附录	201
附录一	Turbo C 编译、连接和运行时的常见错误	201

附录二 常见错误的解决·····	202
附录三 Turbo C 2.0 集成开发环境的使用·····	206

第三篇 C 语言程序设计课程设计

第一章 基本算法程序设计 ·····	214
1.1 翻扑克牌·····	214
1.2 八皇后问题·····	214
第二章 图形程序设计 ·····	216
2.1 绘制圆饼图形·····	216
2.2 图形时钟·····	217
第三章 游戏程序设计 ·····	218
3.1 五子棋游戏·····	218
3.2 黑白棋游戏·····	227
第四章 数据库管理程序设计 ·····	228
4.1 人员信息管理·····	228
4.2 学生成绩管理系统·····	230
参考文献 ·····	231

第一篇 C 语言程序设计学习指导

第一章 程序设计基础

1.1 知识点回顾

1.1.1 几个基本概念

1.1.1.1 程序

一个程序是完成某一特定任务的一组指令序列，或者说，是为实现某一算法的指令序列。

数据结构+算法=程序

数据结构：数据在计算机中的组织和存储形式的抽象。数据结构的具体化就是指某种计算机语言的数据类型。

算法：为解决一个问题而采取的方法和步骤，称为“算法”（algorithm）。解决一个问题的过程就是实现一个算法的过程。

1.1.1.2 程序设计语言

因为计算机不能直接理解和执行人类使用的自然语言，而只能接受和执行二进制的指令，所以，人机交互必须解决语言的沟通。程序语言可以分为低级语言和高级语言两大类。

低级语言：又叫面向机器的语言，它是特定的计算机系统所固有的语言。它又可分为机器语言和符号语言（汇编语言）两类。

高级语言：用一种接近自然语言和数学语言的专用语言来表示算法，它与具体的计算机无关，即用它所写的程序可以在任一种计算机上运行。如：FORTRAN、COBOL、ALGOL 60、BASIC、PASCAL、C、C++等。

1.1.1.3 程序设计步骤

- (1) 建立数学模型。
- (2) 选定算法，并采用适当的工具描述。
- (3) 编程。
- (4) 测试及调试。

1.1.1.4 结构化程序设计

要设计出结构化程序，应当采用以下的方法：

- (1) 自顶向下。

(2) 逐步细化。

(3) 模块化。

1.1.2 C语言结构及特点

1.1.2.1 C语言的特点

(1) 语言表达能力强。

(2) 语言简洁、紧凑，使用灵活，易于学习和使用。

(3) 数据类型丰富，具有很强的结构化控制性。

(4) 语言生成的代码质量高。

(5) 语法限制不严格，程序设计自由度大。

(6) 可移植性好。

1.1.2.2 C程序的结构

(1) C程序是由一个或多个函数所组成的，每个函数完成相对独立的功能，函数是C程序的基本单位。

(2) 一个完整的C程序有且仅有一个主函数（main()）。

(3) 程序总是从main()函数的第一条语句开始执行，执行完main()函数中的所有语句则意味着整个C程序执行完成。其他函数都是在执行main()函数时，通过函数调用或嵌套调用来执行的。

(4) main()函数在程序中的位置：C语言规定main()函数在程序中的位置是任意的，可以放在程序的开头、中间或结尾。

(5) C语言规定每条语句以分号“;”结束，分号是语句不可缺少的一部分。

1.1.3 C语句概述

C程序的执行部分是由语句组成的，程序的功能也是由执行语句实现的。C语句可分为以下五类。

1.1.3.1 表达式语句

表达式语句由表达式加上分号组成。其一般形式为：

表达式；

执行表达式语句就是计算表达式的值。例如：

x=y+z; 赋值表达式语句

y+z; 算术运算语句（由于计算结果不保留，无实际意义）

i++; 自增语句，i值增1

1.1.3.2 函数调用语句

函数调用语句由函数名、实际参数加上分号“;”组成。其一般形式为：

函数名（实际参数表）；

执行函数语句就是调用某函数，将实际参数传递给被调函数中定义的形式参数，然后执行被调函数体中的语句，并求函数值（在后面函数章节中再详细介绍）。例如：

printf("C Program"); 调用库函数printf()，完成输出字符串：C Program。

1.1.3.3 控制语句

控制语句用于控制程序的流程，以实现各种不同的程序结构，它们由特定的语句定义

符组成。C 语言有九种控制语句。可分成以下三类：

条件判断语句：if 语句、switch 语句。

循环执行语句：do-while 语句、while 语句、for 语句。

转移语句：break 语句、goto 语句、continue 语句、return 语句。

1.1.3.4 复合语句

把多条语句用花括号 { } 括起来组成的一条语句称为复合语句。在程序中应把复合语句看成是单条语句，而不是多条语句。例如：

```
{ x=y+z; a=b+c;
  printf ("%d%d\n", x, a);
}
```

这是一条复合语句。复合语句内的各条语句都必须以分号“；”结尾，在右花括号“}”外不能再加分号。

1.1.3.5 空语句

只有分号“；”组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句经常用来作空循环体。例如：

```
while (getchar() != '\n');
```

本语句的功能是：只要从键盘输入的字符不是回车符就重新输入。这里的循环体为空语句。

1.1.4 注释信息和 C 程序的书写风格

(1) “/*”与“*/”之间的信息称为注释信息，它可以帮助人们阅读程序。注释信息不会被 C 语言编译程序所编译，它对程序的运行结果不会产生任何影响。插入注释的要求是：允许在任何能够插入空格符的位置插入注释。注意，C 语言不能嵌套注释。

(2) C 程序的书写基本没有限制，比较自由，即每条语句可从任意列开始。为了使程序层次清楚，建议采用缩进格式书写程序。C 程序中每行可以写多条语句，但要注意整体的美观，大多数情况下人们习惯采用一行一句的书写风格。

1.2 典型例题解析

【例1】 以下说法不正确的是（ ）。

- A. C 程序中必须有一个 main() 函数，C 程序是从 main() 函数的第 1 条语句开始执行
- B. C 程序中除主函数外的其他函数都是在执行主函数时，通过函数调用或嵌套调用而得以执行的
- C. C 程序中的 main() 函数位置可以任意指定
- D. C 程序中的 main() 函数必须放在程序的开始位置

【分析】 考核的知识点是 main() 函数的作用与位置。一个完整的 C 程序有且仅有一个主函数 (main())；C 语言规定 main() 函数在程序中的位置是任意的；程序总是从 main() 函数的第 1 条语句开始执行，到 main() 函数最后一条语句结束，其他函数都是在执行 main()

函数时，通过函数调用或嵌套调用而得以执行的。

【答案】 D

【例2】 以下说法不正确的是（ ）。

- A. C 程序中的一行可以写多条语句
- B. C 程序中的每行只能写一条语句
- C. C 程序中的语句可以采用缩进格式书写
- D. C 程序中可以用注释提高程序的可读性

【分析】 考核的知识点是 C 程序的书写风格。C 程序的书写风格比较自由：每条语句可从任意列开始；每行可以写多条语句，一条语句也可以写在多行上；可以适当地添加注释信息以提高程序的可读性。

【答案】 B

【例3】 C 语言的程序一行写不下时，可以（ ）。

- A. 在任意空格处换行
- B. 用逗号换行
- C. 用回车符换行
- D. 用分号换行

【分析】 C 语言对格式的要求不严格，基本上可以按随意的方式书写。一行上可以写一条语句，也可以写多条语句。如果一条语句很长，在一行写不完，也可以分写在多行上。除了编译预处理语句，在多行的分割处不需要做特殊的处理，只要将分割处选择在空格处即可。对于编译预处理语句，如果分成多行书写，则必须在前一行的末尾处添加符号“、”。

【答案】 A

【例4】 以下说法正确的是（ ）。

- A. C 语言程序是从第一个定义的函数开始执行
- B. 在 C 语言中，要调用的函数必须在 main 函数中定义
- C. C 语言程序是从 main 函数的第 1 条语句开始执行
- D. C 语言程序中的 main 函数必须放在程序的开始部分

【分析】 C 语言程序总是从程序的 main 函数处开始执行。main 函数可以放在 C 程序的任何位置，包括最前面和最后面。C 程序中的函数可以任意地相互调用，它们之间的关系是平等的。

【答案】 C

【例5】 下列说法中，正确的是（ ）。

- A. 主函数名是由程序设计人员按照“标识符”的规则选取的
- B. 分号和回车符都可以作为一个语句的结束符号
- C. 程序的执行，总是从源程序清单的第 1 行开始
- D. 在程序清单的任何地方都可以插入一个或多个空格符

【分析】 主函数名规定为“main”，不是程序设计人员自由选取的；分号是表示语句结束的惟一符号；程序的执行总是由主函数的第 1 条语句开始执行。

【答案】 D

1.3 补充习题

一、选择题

- 以下不正确的概念是 ()。
 - 一个 C 程序由一个或多个函数组成
 - 一个 C 程序必须包含一个 main 函数
 - 在 C 程序中, 可以只包括一条语句
 - C 程序的每行上可写多条语句
- 一个 C 程序的执行是从 ()。
 - 本程序的 main 函数开始, 到 main 函数结束
 - 本程序文件的第一个函数开始, 到本程序文件的最后一个函数结束
 - 本程序的 main 函数开始, 到本程序文件的最后一个函数结束
 - 本程序文件的第一个函数开始, 到本程序 main 函数结束
- 下述源程序的书写格式不正确的是 ()。
 - 一条语句可以写在几行上
 - 一行上可以写几条语句
 - 分号是语句的一部分
 - 函数的首部, 其后必须加分号
- 以下叙述正确的是 ()。
 - 在 C 程序中, main 函数必须位于程序的最前面
 - C 程序的每行只能写一条语句
 - C 语言本身没有输入/输出语句
 - 在对一个 C 程序进行编译的过程中, 可发现注释中的拼写错误
- 以下叙述不正确的是 ()。
 - 一个 C 源程序可由一个或多个函数组成
 - 一个 C 源程序必须包含一个 main 函数
 - C 程序的基本组成单位是函数
 - 在 C 程序中, 注释说明只能位于一条语句的后面
- C 语言规定: 在一个源程序中, main 函数的位置 ()。
 - 必须在最开始
 - 必须在系统调用的库函数的后面
 - 可以任意
 - 必须在最后
- 一个 C 语言程序是由 ()。
 - 一个主程序和若干子程序组成
 - 函数组成
 - 若干过程组成
 - 若干子程序组成
- C 语言源程序的基本单位是 ()。
 - 过程
 - 函数
 - 子程序
 - 标识符

二、填空题

- 为解决一个问题而采取的方法和步骤, 称为_____。
- 结构化程序设计中的三种基本结构是_____。

3. 循环结构分_____和_____两种。
4. 计算机能够直接识别和执行的指令集合称为_____。
5. 程序语言可以分为_____和_____两大类。
6. C程序的执行是从_____开始执行的。
7. C语言源程序的语句分隔符是_____。
8. 一个函数由两部分组成：_____和_____。
9. 一个函数体的范围是以_____开始，以_____结束。

第二章 C 语言数据类型及表达式

2.1 知识点回顾

2.1.1 数据类型

数据类型是按被定义变量的性质、表示形式、占据存储空间的大小、构造特点来划分的。在 C 语言中，数据类型可分为：基本数据类型，构造数据类型，指针类型，空类型四大类。

(1) 基本数据类型：包括整型、浮点型（含单精度浮点型和双精度浮点型）、字符型和枚举型。基本数据类型最主要的特点是，其值不可以再分解为其他类型。

(2) 构造数据类型：包括数组，结构体和共用体。构造数据类型是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说，一个构造类型的变量可以分解成若干个“成员”或“元素”。每个“成员”或“元素”都是一个基本数据类型或又一个构造类型。

(3) 指针类型：指针是一种特殊的，同时又具有重要作用的数据类型。其值用来表示某个变量在内存中的地址。注意，虽然指针变量的取值类似于整型量，但这是两个类型完全不同的量，因此不能混为一谈。

(4) 空类型（void）：在调用函数值时，通常应向调用者返回一个函数值。这个返回的函数值是具有一定数据类型的，应在函数定义及函数说明中给以说明。如：

```
int max(int a, int b) {.....}
```

其中“int”类型即表示该函数的返回值为整型量。但是，也有一类函数，调用后并不需要向调用者返回函数值，函数可以定义为“空类型”。如：

```
void max(int a, int b) {.....}
```

2.1.2 标识符

(1) C 语言中，合法的标识符由字母、下划线和数字组成，并且开头第 1 个字符必须为字母或下划线。如：

xa, ba_c, _xyz0123, _135 都是合法的标识符。

而 5abw, xq.pa, _m, sp/= 都是不合法的标识符。

(2) 在 C 语言中大写字母和小写字母被认为是两个不同的字符。如：

XXG 和 xxg 是两个不同的标识符。

(3) 在 C 语言中用户标识符既要符合标识符的命名规则，同时不能和关键字同名。C 语言中的关键字：

auto	continue	enum	if	short
switch	volatile	break	default	extern

int	signed	typedef	while	case
do	float	long	sizeof	union
char	double	for	register	static
unsigned	const	else	goto	return
struct	void			

2.1.3 基本数据类型

2.1.3.1 整型

在C语言中可以用十进制、八进制和十六进制形式的数字表示整型数据及常数。

十进制数：通常的整数表示法。如：0、-10、55、+37。

八进制数：书写时在八进制整数的前面加一个数字0。如：00、072、-010、+041等，它们分别表示十进制数：0、57、-8、33。

十六制整数：书写方法是在通常的十六进制整数的前面加0x。如：0x0、-0x13、+021、0x95等，它们分别表示十进制数0、-19、33、149。

说明：

(1) 正整数前面可以带“+”，也可以不带“+”（省略）。

(2) C语言中整型数据在计算机中一般占两个字节。其取值范围是：-32768~+32767。

(3) 为了扩大整型数据的范围，C语言提供了一种长整型数据，它在计算机中占4个字节（取值范围是：-2147483648~2147483647）。长整型数据的表示也有十进制、八进制和十六进制整数3种表示法。注意，长整型数的末尾须加上后缀字母“l”或“L”。例如15L，-036L，+0x431。

(4) 短整型，基本上与整型相同。

2.1.3.2 实型

C语言中有两种方法表示实型数据，即小数形式和指数形式。

(1) 小数形式的实数：它由整数部分、小数点和小数部分组成。如：3.14159、7.0、-3.9等都是合法的实型数据。

(2) 指数形式的实数：它由尾数部分、e (E)和指数部分组成。如：3.106可以表示成0.3106e1、3.106e0、31.06e-1等形式。

注意：字母e或E的前后都必须是数字，且其后表示指数的数必须为整数。如：e2、0.1e2.4、7.0e等都是不合法的实型数据表示法。

2.1.3.3 字符型

用一对单引号对括起来的一个符号称为字符。单引号对是字符的定界符。

转义字符：C语言中有一类以反斜杠开头的字符序列，称为转义字符。有以下几种情况的转义字符：

(1) 用反斜杠加一个字母代表一个控制字符。如：‘\n’表示换行控制符。

(2) 用反斜杠加1~3位八进制数字，它表示ASCII码为该八进制数的字符。如：‘\102’是字符‘B’的另外一种表示。

(3) 用反斜杠加x和1~2位十六进制数字，它表示ASCII码为该十六进制数的字符。如：‘\x41’是字符‘A’的另外一种表示。

(4) 一些特殊字符的转义字符。如'\w'表示字符'w'; '\''表示字符''等。

2.1.3.4 字符串型

用一对双引号对话起来的零个或多个字符序列称为字符串。如:"hello"。注意,字符串"hello"在内存中占有六个字节的存储单元,最后一个字节存放字符'\0',它将作为字符串的结束标记。

一个典型问题:字符'x'与字符串"x",在内存中占用相同的字节吗?

2.1.4 常量和变量

2.1.4.1 常量

常量(常数)是指在程序的运行过程中其值保持不变的量。C语言规定常量的类型有:整型常量、实型常量、字符常量、字符串常量。

常量无须事先定义,在程序中可以直接使用;常量的类型也不需要说明:由常量的书写形式可以直接判断常量的类型。

注意:长整型常量需在常量后加l或L。

2.1.4.2 变量

变量是在程序运行期间其值可以改变的量。每个变量都必须有一个标识名(变量名);变量其实代表了计算机内部的一个存储单元,不同类型的变量对应于内存中固定长度的存储单元;每个变量都有地址:变量对应的存储单元的首地址即为该变量的地址;每个变量都有一个值:变量对应的存储单元中存放的数据称为该变量的值。

说明:

(1) 要区分变量名和变量值这两个不同的概念。如有定义: `int x=5;`在下面的两个表达式: `"x*2"`、`"&x"`中,变量名体现出不同的属性。

(2) 变量定义必须放在变量使用之前,一般放在函数体的开头部分。格式为:

类型标识符 变量名1, 变量名2,

(3) 定义变量时,编译程序给变量分配存储单元,但并没有对存储单元赋初值,因此在这些存储单元中原有的数据并没有被清除,此时变量中的值对于程序是无意义。也就是说,一个变量在使用前要考虑应该先对它赋初值。

2.1.4.3 内部变量和外部变量

(1) 内部变量:在一个函数内部定义的变量是内部变量。由于它只在本函数范围内有效,所以其作用域是局部的(局部变量)。

(2) 外部变量:在函数外定义的变量称外部变量。它的作用域是从定义处到程序的结束(全局变量)。

如果想在定义点之前的函数中引用该外部变量,则应该在该函数中或该函数前用关键字 `extern` 作“外部变量说明”,用于延伸该外部变量的作用域。

如果在程序中,外部变量与局部变量同名,在局部变量作用范围内,外部变量将被屏蔽(暂不起作用)。

2.1.5 C语言运算符

运算符是表示某种操作的符号,操作的对象叫运算量。用运算符把运算量连接成一个有意义的式子叫表达式。每种高级语言都有自己的运算符。C语言提供了大量的运算符和

丰富的表达式类型。

2.1.5.1 C语言的运算符分类

(1) 算术运算符：用于各类数值运算。包括加 (+)、减 (-)、乘 (*)、除 (/)、求余 (或称模) 运算 (%)、自增 (++)、自减 (--) 共7种。

(2) 关系运算：用于比较运算,包括大于 (>)、小于 (<)、大于等于 (>=)、小于等于 (<=) 和不等于是 (!=) 6种。

(3) 逻辑运算符：用于逻辑运算,包括逻辑与 (&&)、逻辑或 (||)、逻辑非 (!) 3种。

(4) 位操作运算符：参与运算的量,按二进制位进行运算,包括位与 (&)、位或 (|)、位非 (~)、位异或 (^)、左移 (<<)、右移 (>>) 6种。

(5) 赋值运算符：用于赋值运算,分为简单赋值 (=)、复合算术赋值 (+=、-=、*=、/=、%=) 和复合位运算赋值 (&=, |=, ^=, >>=, <<=) 三类共11种。

(6) 条件运算符：这是一个三目运算符,用于条件求值 (表达式1? 表达式2: 表达式3)。

(7) 逗号运算符：用于把若干表达式组合成一个表达式 (,)。

(8) 指针运算符：用于取内容 (*) 和取地址 (&) 两种运算。

(9) 求字节数运算符：用于计算数据类型所占的字节数 (sizeof (类型名))。

(10) 特殊运算符：有括号 (), 下标 [] , 成员 (→, .) 等几种。

2.1.5.2 运算符优先级小结

(1) C语言中,运算符的运算优先级共分为15级,1级最高,15级最低。在有多个不同级别的运算符出现的表达式中时,优先级较高的先于优先级较低的进行运算,如果在一个“目”(即运算对象)两侧的运算符优先级相等时,则按运算符的结合性所规定的结合方向处理。

(2) C语言中各运算符的结合性分为两种,即左结合性(自左至右)和右结合性(自右至左)。其中算术运算符的结合性是自左至右,即先左后右。如有表达式“x-y+z”则“y”应先与“-”结合,执行“x-y”运算,然后再执行“+z”的运算。这种自左至右的结合方向就称为“左结合性”。而自右至左的结合方向称为“右结合性”,最典型的右结合性运算符是赋值运算符。如“x=y=z”,由于“=”的右结合性,应先进行“y=z”。右结合性的运算符往往会引起一些错误,所以应特别注意。

(3) 同一优先级的运算符,运算次序由结合方向决定。例如“*”与“/”具有相同的优先级,其结合方向是自左向右,因此“3*5/4”的运算次序是先乘后除。“- (负号)”和“++”为同一优先级,结合方向为自右向左,因此“-i++”相当于“- (i++)”。

(4) 不同的运算符要求有不同的运算对象个数。如+、-是双目运算符,要求在运算符两侧各有一个运算对象(如“1+2”,“1-3”等),而“++”和“--”运算是单目运算符,只能在运算符的一侧出现一个运算对象(如“i++”,“j--”等)。条件运算符是C语言中惟一的一个三目运算符,如:“x? a: b”。

(5) 从以上分析大致归纳出各类运算符的优先级:

初等运算符: ()、[]、->、.

↓

单目运算符