

C语言程序设计学习 指导与实验教程(第二版)

- ◆ C语言基础知识
- ◆ 顺序结构程序的设计
- ◆ 选择结构程序的设计
- ◆ 循环结构程序的设计
- ◆ 数组的使用
- ◆ 函数的使用
- ◆ 预处理命令
- ◆ 指针的使用
- ◆ 结构体、链表与共用体

运算

- ◆ 文件的概念和操作

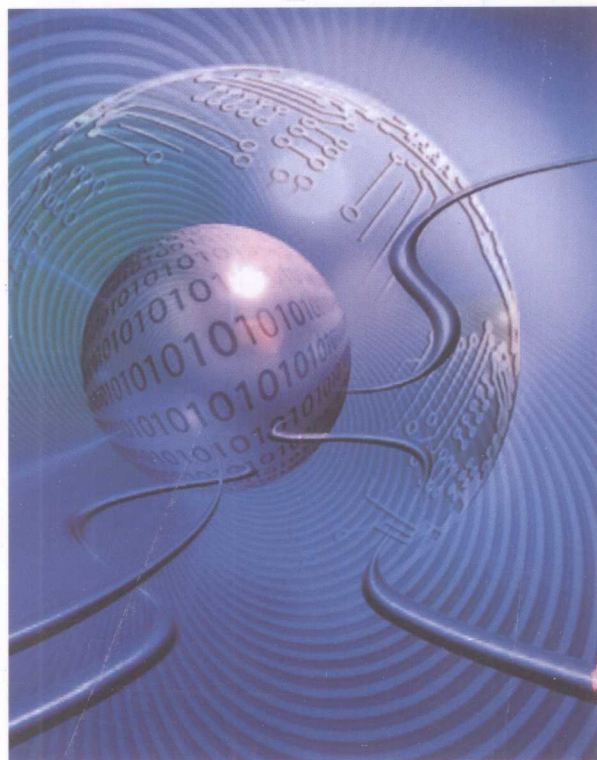
要点与难点

例题分析

多种类型的习题

习题参考答案

上机实验指导



潘洪军 陈荣品 谭小球

冯相忠 主 编
高 禹 主 审
张建科 副主编



清华大学出版社

013061003

TP312C
1895-2

高等学校计算机应用规划教材 简 容 内

C 语言程序设计学习指导与 实验教程(第二版)

冯相忠 主 编

潘洪军 陈荣品 谭小球 张建科 副主编



清华大学出版社



北航

C1666590

TP 312C

1895-2

013081002

内 容 简 介

本书是与教材《C 语言程序设计(第二版)》配套的学习指导与实验用书,内容包括 C 语言程序设计学习指导和 C 语言程序设计实验教程两部分。在学习指导部分,介绍了 C 语言各章知识的要点和难点,选择了一些典型例题进行分析,选编了许多练习题,这些习题题型丰富、覆盖面广,有选择题、填空题、阅读程序写结果题、编写程序题,每道题都给出了参考答案。通过这些练习题,可以训练读者理解和掌握 C 语言的基本概念与基本语句,编写程序、学习编程方法和技巧。在实验教程部分,介绍了基于 Windows 平台的 C 语言开发工具 Win-TC 的使用方法;介绍了 11 个实验内容,为读者在计算机上进行程序的编辑、调试和运行进行详细指导。对于每个实验,列出了实验目的、实验知识内容提要、实验的具体内容以及实验完成后的思考问题。通过这些实验,提高读者的实际动手能力。

本书条理清楚、语言流畅、通俗易懂,实用性强,既可以作为高等院校应用型本科专业学生的教材,也可以供自学者以及参加 C 语言计算机等级考试者阅读使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计学习指导与实验教程/冯相忠 主编. —2 版. —北京:清华大学出版社,2013.7
(高等学校计算机应用规划教材)

ISBN 978-7-302-32621-2

I. ①C… II. ①冯… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 122442 号

责任编辑:胡辰浩 袁建华

装帧设计:牛艳敏

责任校对:邱晓玉

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:16

字 数:369 千字

版 次:2011 年 1 月第 1 版

2013 年 7 月第 2 版

印 次:2013 年 7 月第 1 次印刷

印 数:1~4000

定 价:29.00 元

前 言

C 语言是广泛使用的一种计算机语言,由于它具有功能丰富、灵活性强、可移植性好、语言简洁、应用面广等特性,因此深受广大用户的喜爱。

初学 C 语言的人,可能感觉学起来比较吃力,具体表现为:不知该如何理解 C 语言的概念,如何正确使用 C 语言的语句和语法,如何设计编写程序,如何综合运用 C 语言知识处理实际问题。有鉴于此,为了帮助读者学好 C 语言,顺利通过各级各类相关考试,并能熟练将其运用于实际工作,我们组织长期从事 C 语言教学工作的教师,编写了这本《C 语言程序设计学习指导与实验教程(第二版)》。

本书分为两方面内容,一方面是为读者提供 C 语言的学习指导,另一方面是 C 语言的实验教程。在学习指导部分,对 C 语言各章知识的要点和难点进行了整理归纳和深入分析,以使读者明确各章应该着重掌握的知识和深入理解的问题。在学习指导部分,选择了一些典型例题进行分析,以使读者理解各章知识,学会灵活运用各章知识。在学习指导部分,选编了相应练习题,每道题都给出了参考答案。这些练习题既可以训练读者去理解和掌握 C 语言的基本概念与基本语句,又可以训练读者编写程序、学习编程方法和技巧。对于每一道编写程序题,本书只给出了一种参考答案,因为每个问题一般都有多种编程方法,所以读者不要受参考答案的限制。在实验教程部分,介绍了基于 Windows 平台的 C 语言开发工具 Win-TC 的使用方法;根据 C 语言的教学内容,本书设计了 11 个上机练习实验,为读者在计算机上进行程序的编辑、运行和调试进行详细的指导。在实验教程部分,对于每个实验,本书均列出了该实验要达到的目的、该实验所涉及的 C 语言知识要点、该实验要做的具体内容以及实验完成后的思考问题。认真完成这些实验,能够极大地提高读者的编程能力。

除主编和副主编外,参加本书编写的还有乐天、叶其宏、李慧、陈洪涛、吴远红、郑芸、张艳艳、姚笑秋、黄海峰、崔振东等。

由于编者水平有限,书中难免存在错误与不足之处,诚恳欢迎读者批评指正。我们的联系方式为邮箱: huchenhao@263.net, 电话: 010-62796045。

编 者

2013 年 5 月

目 录

第 1 篇 C 语言程序设计学习指导	1
第 1 章 C 语言概述	1
1.1 本章要点	1
1.1.1 C 语言的特点	1
1.1.2 C 语言程序的结构	1
1.1.3 C 语言源程序的编辑、编译、 连接和运行	1
1.2 习题	2
1.3 习题参考答案	2
第 2 章 数据类型、运算符和表达式	3
2.1 本章要点	3
2.1.1 C 语言的数据类型	3
2.1.2 标识符	3
2.1.3 常量	3
2.1.4 变量	4
2.1.5 数据类型的转换	4
2.1.6 C 语言运算符的优先级和 结合性	5
2.1.7 算术运算符和算术表达式	6
2.1.8 自增运算符和自减运算符	6
2.1.9 赋值运算符和赋值表达式	6
2.1.10 复合赋值运算符	6
2.1.11 逗号运算符	6
2.2 本章难点	6
2.2.1 运算符的优先级	6
2.2.2 运算符的结合性	7
2.2.3 复合赋值运算	7
2.2.4 赋值类型转换	7
2.3 例题分析	8

2.4 习题	9
2.4.1 单项选择题	9
2.4.2 填空题	12
2.4.3 阅读程序写结果题	13
2.4.4 编写程序题	15
2.5 习题参考答案	15
2.5.1 单项选择题答案	15
2.5.2 填空题答案	15
2.5.3 阅读程序写结果题答案	16
2.5.4 编写程序题参考答案	16
第 3 章 程序设计初步	17
3.1 本章要点	17
3.1.1 格式输入函数 scanf()	17
3.1.2 格式输出函数 printf()	18
3.1.3 单字符输入函数 getchar()	18
3.1.4 单字符输出函数 putchar()	18
3.1.5 赋值语句	19
3.1.6 复合语句和空语句	19
3.1.7 顺序结构	19
3.2 本章难点	20
3.2.1 printf()函数中的格式字符、 附加格式符和转义符	20
3.2.2 scanf()函数中的格式字符和 附加格式符	20
3.3 例题分析	21
3.4 习题	22
3.4.1 单项选择题	22
3.4.2 填空题	24
3.4.3 阅读程序写结果题	26
3.4.4 编写程序题	26

3.5 习题参考答案·····	27	5.1.5 多重循环·····	50
3.5.1 单项选择题答案·····	27	5.1.6 break 语句和 continue 语句·····	50
3.5.2 填空题答案·····	27	5.2 本章难点·····	51
3.5.3 阅读程序写结果题答案·····	28	5.2.1 循环结构的理解·····	51
3.5.4 编写程序题参考答案·····	28	5.2.2 3种循环结构的比较·····	51
第4章 选择结构程序的设计·····	31	5.2.3 多重循环·····	52
4.1 本章要点·····	31	5.3 例题分析·····	52
4.1.1 关系运算符与关系表达式·····	31	5.4 习题·····	55
4.1.2 逻辑运算符与逻辑表达式·····	31	5.4.1 单项选择题·····	55
4.1.3 二分支选择结构——if 语句·····	32	5.4.2 填空题·····	57
4.1.4 条件运算符和条件表达式·····	33	5.4.3 阅读程序写结果题·····	59
4.1.5 多分支选择结构—— switch 语句·····	34	5.4.4 编写程序题·····	63
4.2 本章难点·····	34	5.5 习题参考答案·····	64
4.2.1 if 语句的嵌套·····	34	5.5.1 单项选择题答案·····	64
4.2.2 条件表达式的使用·····	35	5.5.2 填空题答案·····	64
4.2.3 switch 语句和 break 语句的 使用·····	35	5.5.3 阅读程序写结果题答案·····	65
4.3 例题分析·····	36	5.5.4 编写程序题参考答案·····	66
4.4 习题·····	38	第6章 数组·····	72
4.4.1 单项选择题·····	38	6.1 本章要点·····	72
4.4.2 填空题·····	41	6.1.1 一维数组·····	72
4.4.3 阅读程序写结果题·····	42	6.1.2 二维数组·····	72
4.4.4 编写程序题·····	44	6.1.3 字符数组·····	73
4.5 习题参考答案·····	44	6.2 本章难点·····	74
4.5.1 单项选择题答案·····	44	6.2.1 数组元素的下标·····	74
4.5.2 填空题答案·····	45	6.2.2 字符串和字符数组·····	74
4.5.3 阅读程序写结果题答案·····	45	6.2.3 字符串处理函数·····	75
4.5.4 编写程序题参考答案·····	45	6.3 例题分析·····	75
第5章 循环结构程序的设计·····	48	6.4 习题·····	78
5.1 本章要点·····	48	6.4.1 单项选择题·····	78
5.1.1 while 语句构成的循环·····	48	6.4.2 填空题·····	80
5.1.2 do-while 语句构成的循环·····	48	6.4.3 阅读程序写结果题·····	82
5.1.3 for 语句构成的循环·····	49	6.4.4 编写程序题·····	86
5.1.4 goto 语句以及 goto 语句 构成的循环·····	49	6.5 习题参考答案·····	87
		6.5.1 单项选择题答案·····	87
		6.5.2 填空题答案·····	87
		6.5.3 阅读程序写结果题答案·····	87

6.5.4 编写程序题参考答案	88	8.2 本章难点	115
第7章 函数	95	8.2.1 宏展开	115
7.1 本章要点	95	8.2.2 条件编译	115
7.1.1 函数的定义	95	8.3 例题分析	115
7.1.2 函数的参数和返回值	96	8.4 习题	116
7.1.3 函数调用	96	8.4.1 单项选择题	116
7.1.4 函数声明	97	8.4.2 填空题	119
7.1.5 数组名作为函数参数	97	8.4.3 阅读程序写结果题	120
7.1.6 全局变量和局部变量	97	8.4.4 编写程序题	121
7.1.7 变量的存储类别	97	8.5 习题参考答案	121
7.1.8 内部函数与外部函数	98	8.5.1 单项选择题答案	121
7.2 本章难点	98	8.5.2 填空题答案	121
7.2.1 参数的传递	98	8.5.3 阅读程序写结果题答案	122
7.2.2 函数的声明	98	8.5.4 编写程序题参考答案	122
7.2.3 函数的递归调用	98	第9章 指针	124
7.2.4 数组名作为函数参数	99	9.1 本章要点	124
7.2.5 变量的作用范围	99	9.1.1 指针变量的定义	124
7.2.6 静态存储类别	99	9.1.2 指针变量的赋值	124
7.3 例题分析	100	9.1.3 对指针变量的操作	125
7.4 习题	101	9.1.4 指向指针变量的指针变量	126
7.4.1 单项选择题	101	9.1.5 数组与指针	126
7.4.2 填空题	103	9.1.6 指针数组	127
7.4.3 阅读程序写结果题	106	9.1.7 指向函数的指针变量	127
7.4.4 编写程序题	108	9.1.8 带参数的 main()函数	128
7.5 习题参考答案	109	9.1.9 void 类型的指针	128
7.5.1 单项选择题答案	109	9.2 本章难点	128
7.5.2 填空题答案	109	9.2.1 指针变量的概念	128
7.5.3 阅读程序写结果题答案	110	9.2.2 对指针变量的操作	129
7.5.4 编写程序题参考答案	110	9.2.3 字符指针变量与字符数组的 区别	130
第8章 预处理命令	113	9.3 例题分析	130
8.1 本章要点	113	9.4 习题	136
8.1.1 不带参数的宏	113	9.4.1 单项选择题	136
8.1.2 带参数的宏	113	9.4.2 填空题	140
8.1.3 文件包含	114	9.4.3 阅读程序写结果题	141
8.1.4 条件编译	114		

9.4.4 编写程序题 (要求使用指针).....	144	10.5.4 编写程序题参考答案.....	173
9.5 习题参考答案.....	145	第 11 章 位运算.....	179
9.5.1 单项选择题答案.....	145	11.1 本章要点.....	179
9.5.2 填空题答案.....	145	11.1.1 位运算符和位运算.....	179
9.5.3 阅读程序写结果题答案.....	145	11.1.2 位运算的优先级.....	180
9.5.4 编写程序题参考答案.....	145	11.1.3 位段.....	181
第 10 章 结构体与其他数据类型.....	150	11.2 例题分析.....	181
10.1 本章要点.....	150	11.3 习题.....	183
10.1.1 结构体概述.....	150	11.3.1 单项选择题.....	183
10.1.2 定义结构体类型变量的 方法.....	150	11.3.2 填空题.....	183
10.1.3 结构体变量的引用和 初始化.....	150	11.3.3 阅读程序写结果题.....	184
10.1.4 结构体数组.....	151	11.3.4 编写程序题.....	185
10.1.5 指向结构体数据的 指针.....	151	11.4 习题参考答案.....	185
10.1.6 用指针处理链表.....	152	11.4.1 单项选择题答案.....	185
10.1.7 共用体.....	153	11.4.2 填空题答案.....	186
10.1.8 枚举类型.....	153	11.4.3 阅读程序写结果题答案.....	186
10.1.9 用 typedef 定义类型.....	154	11.4.4 编写程序题参考答案.....	186
10.2 本章难点.....	154	第 12 章 文件.....	188
10.2.1 在函数之间传递结构体 数据.....	154	12.1 本章要点.....	188
10.2.2 结构体与共用体的区别.....	154	12.1.1 文件概述.....	188
10.2.3 链表操作.....	155	12.1.2 文件类型指针.....	188
10.3 例题分析.....	155	12.1.3 文件的打开和关闭.....	188
10.4 习题.....	159	12.1.4 文件的读/写.....	190
10.4.1 单项选择题.....	159	12.1.5 文件定位.....	191
10.4.2 填空题.....	162	12.1.6 文件检测.....	192
10.4.3 阅读程序写结果题.....	166	12.2 本章难点.....	192
10.4.4 编写程序题.....	170	12.2.1 文件位置指针的合理 定位.....	192
10.5 习题参考答案.....	171	12.2.2 各文件读/写函数的区别.....	193
10.5.1 单项选择题答案.....	171	12.3 例题分析.....	193
10.5.2 填空题答案.....	171	12.4 习题.....	197
10.5.3 阅读程序写结果题答案.....	172	12.4.1 单项选择题.....	197
		12.4.2 填空题.....	199
		12.4.3 阅读程序写结果题.....	201
		12.4.4 编写程序题.....	202

12.5 习题参考答案.....	203	实验五 循环结构程序设计.....	226
12.5.1 单项选择题答案.....	203	实验六 数组的应用(一)——	
12.5.2 填空题答案.....	203	数值数据的数组.....	229
12.5.3 阅读程序写结果题答案.....	203	实验七 数组的应用(二)——	
12.5.4 编写程序题参考答案.....	203	字符数据的数组.....	232
第 2 篇 C 语言程序设计实验教程.....	208	实验八 函数.....	235
实验一 熟悉 Win-TC 的运行环境和		实验九 指针.....	238
开发 C 程序的步骤.....	208	实验十 结构体与共用体.....	241
实验二 数据类型、运算符、表达式.....	215	实验十一 文件.....	244
实验三 简单 C 程序设计.....	219		
实验四 选择结构程序设计.....	222		

第1篇 C语言程序设计学习指导

第1章 C语言概述

1.1 本章要点

1.1.1 C语言的特点

- (1) 语言简洁、紧凑，使用方便、灵活，具有丰富的运算符和数据结构。
- (2) 允许直接访问物理地址，能够进行位操作，可实现汇编语言的大部分功能，可以直接对硬件进行操作。
- (3) 具有结构化的控制语句，是结构化的理想语言。
- (4) 语法限制不太严格，程序设计自由度高。
- (5) 编写的程序可移植性好。
- (6) 生成目标代码质量好，程序执行效率高。

1.1.2 C语言程序的结构

- (1) 一个源程序由若干个函数组成，其中有且仅有一个主函数(main 函数)。
- (2) 一个函数由函数首部(函数第一行)和函数体(函数首部后面大括号内的部分)组成。函数首部包括函数类型、函数名和括号中的若干个参数；函数体由声明部分和执行部分组成。
- (3) 程序书写格式自由，一行内可写多条语句，一个语句也可以分写在多行上，且语句中的空格和回车符均可忽略不计。
- (4) 程序的注释内容放在/*和*/之间，/和*之间不允许有空格；注释部分允许出现在程序中的任何位置。

1.1.3 C语言源程序的编辑、编译、连接和运行

使用 Turbo C 或 Win-TC 编写好 C 语言源程序，经过编译、连接生成可执行文件，即可

运行。

1.2 习 题

1. 简述 C 语言源程序的结构特点。
2. 写出一个 C 语言源程序的构成。
3. 编写完成如下任务的程序，然后上机编译、连接并运行。

输出两行字符，第 1 行是 “The computer is our good friends!”，第 2 行是 “We learn C language.”。

4. 编写完成如下任务的程序，然后上机编译、连接并运行。

已知 $a=10$ 、 $b=5$ ，计算 $a+b$ 、 $a-b$ 的值，输出计算结果。

1.3 习题参考答案

1. (略)

2. (略)

3.

```
main()
{printf("The computer is our good friends!\n");
printf("We learn C language.\n"); return 0; }
```

4.

```
main()
{int a=10,b=5,c,d; c=a+b; d=a-b;
printf("a+b=%d,a-b=%d \n",c,d); return 0; }s
```

第2章 数据类型、运算符和表达式

2.1 本章要点

2.1.1 C 语言的数据类型

C 语言提供了丰富的数据类型,通过这些数据类型可以实现如链表、栈等复杂数据结构的编写。程序中的每一个量(包括常量和变量)都属于某一个特定的数据类型。

2.1.2 标识符

所谓标识符就是指在程序中用于标识函数、变量以及常量且符合一定命名规则的字符串。C 语言标识符可以包括一个或多个字符,字符可以是英文字母、数字或下划线,但第一个字符必须是字母或下划线。

特别提醒: C 语言标识符区分字母大小写;标识符命名不宜过长,最好有明确的含义。

2.1.3 常量

常量是在程序运行过程中其值不可被改变的量。注意是不可改变,而不是不改变,不改变的量不一定就是常量。常量的类型包括整型常量、实型常量、字符常量、字符串常量和符号常量 5 类,下面分别予以介绍。

1. 整型常量

整型常量有 3 种形式,分别为十进制整型常量、八进制整型常量和十六进制整型常量。

特别提醒:

八进制数以数字“0”开头,十六进制数以“0x”开头,第一位都不是字母“o”,以字母“o”开头不是常量的写法,这需要初学者特别注意。

2. 实型常量

实型常量有两种表示形式,分别为小数形式和指数形式。

3. 字符常量

一个字符常量代表 ASCII 字符集中的一个字符,在程序中用单引号(')引起来。

特别提醒:

'a'和'A'是两个不同的字符常量;除了这类可打印字符使用的格式以外,对于一些特殊字符常量,采用转义字符的方式表示,特别注意转义字符的3种构成方式。

4. 字符串常量

字符串常量指的是用双引号括引起来的一个或多个字符。

5. 符号常量

符号常量是使用宏定义“#define”来定义的常量,即用一个标识符代表一个常量,程序中出现的该标识符,等同于使用定义时所用到的常量。符号常量可以是上述各常量的任何一种类型。

2.1.4 变量

变量就是其值在程序运行过程中可以改变的量。注意:变量的值可以改变,但并不是非变不可。变量要有变量名,变量的实质是代表一定的存储单元,存储单元中存储的是该变量的值,通过使用变量名可以引用其所代表的存储单元中的内容。不同类型的变量存储单元的大小不同,变量的使用必须符合“先定义,后使用”的原则。变量的类型包括整型变量、实型变量、字符型变量3类,下面分别予以介绍。

1. 整型变量

整型变量有6种,分别为:有符号基本整型(signed int)、无符号基本整型(unsigned int)、有符号短整型(signed short int)、无符号短整型(unsigned short int)、有符号长整型(signed long int)、无符号长整型(unsigned long int)。

2. 实型变量

实型变量包括单精度(float)和双精度类型(double)两种。

3. 字符型变量

字符型变量用于存放一个字符,用关键字 char 来声明。

在C语言中,字符型变量也有带符号与无符号之分,一般情况下,直接使用 char 声明的字符变量常常是带符号的,其数值范围为-128~127;可以使用 unsigned char 声明无符号字符型变量,其数值范围为0~255。

2.1.5 数据类型的转换

当同一表达式中各个数据的类型不同时,需要把它们转换成同一类型后再进行计算,这种转换可以由编译程序自动实现,即自动转换;也可以由程序人员在编译程序时使用类

型转换运算符实现，即强制转换。

1. 自动转换

自动转换又可以分为两类，一类是必然实现的转换，即不论参与运算的数据类型是否一致，某些类型的数据也必然转换为另一种类型，主要包括如下两种情况。

(1) 凡属于 char、short 类型的变量在运算时一律转换为 int 类型。

(2) 凡属于 float 类型的变量在运算时一律转换为 double 类型。

另一类是当运算对象的数据类型不同时，按照从低到高的顺序进行转换。例如，若 a 是 int 型，b 是 double 型，计算 a/b 时，将 a 转换成 double 型后，再与 b 相除。

2. 强制转换

格式为：(数据类型标识)表达式

其作用是把表达式的结果转换为由“数据类型标识”指定的数据类型。例如，“(double)(a+b)”，是将(a+b)的值强制转换为 double 类型，注意与“(double)a+b”不同。

2.1.6 C 语言运算符的优先级和结合性

所谓运算符的优先级，是指当一个运算量的两边连接两个运算符时，根据运算符的优先级决定先运算其左边的还是右边的运算符。运算符的结合性，是指优先级相同的运算符(也包括相同的运算符)出现在同一表达式中时，根据运算符的结合性来决定先对哪一个运算符进行操作，可分为自左向右结合和自右向左结合两种。

本章涉及的具体运算符的优先级和结合性如表 1-2-1 所示。

表 1-2-1 C 语言运算符的优先级和结合性

优 先 级	运 算 符	名 称	结合方向
1	++	增 1 运算符	自右向左
	--	减 1 运算符	
	-	负号运算符	
	(类型)	类型转换运算符	
	sizeof	长度运算符	
2	*	乘法运算符	自左向右
	/	除法运算符	
	%	取模运算符	
3	+	加法运算符	自左向右
	-	减法运算符	
4	= += -= *=	赋值运算符	自右向左
	/= %=		
5	,	逗号运算符	自左向右

2.1.7 算术运算符和算术表达式

算术运算符包括加(+)、减(-)、乘(*)、除(/)、模(%),其运算符符合数学上的运算规则。需要说明的是:两个整型量相除时,所得的结果为整型,如 3/2 的结果为 1,而非 1.5;模运算要求连接的两个量必须是整型数据。

算术表达式即指用算术运算符和括号将运算量连接起来、符合 C 语言语法规则的表达式。括号可以改变运算符的自然运算顺序,注意只能使用小括号。

2.1.8 自增运算符和自减运算符

自增运算是使变量的值增 1,而自减运算是使变量的值减 1。

特别提醒:

- (1) 只有变量才能用自增(减)运算符连接,不可以将该类运算符用于表达式或常量。
- (2) 自增(减)运算有前缀方式和后缀方式两种,使用时注意两者的区别。
- (3) 自增(减)运算符为单目运算符,其结合性都是自右向左结合。

2.1.9 赋值运算符和赋值表达式

赋值运算符的一般使用形式为“变量=表达式”,用于连接一个变量(准确一点说是内存单元)与一个表达式,其功能是把右边表达式的值赋予左边的变量。

左边的变量(内存单元)称为左值,右边的表达式称为右值。

2.1.10 复合赋值运算符

在赋值运算符之前加上其他运算符可以构成复合赋值运算符,这主要是为了提高 C 语言编译器的编译效率。例如,与算术运算符结合可以形成+=, -=, *=, /=, %=。

需要说明的是:复合赋值运算符的两个符号之间一定不能有空格,复合赋值运算符与赋值运算符(=)具有相同的优先级和结合性。

2.1.11 逗号运算符

用逗号运算符将两个表达式连接起来所形成的表达式称为逗号表达式,一般形式如下:

表达式 1, 表达式 2

需要说明的是:逗号运算符连接的表达式 1 或表达式 2 也可以是逗号表达式。

逗号表达式通常用于连接一些处于并列关系的表达式,如可以用在 for 语句中。

2.2 本章难点

2.2.1 运算符的优先级

不同优先级的运算符出现在同一表达式中时,必须根据各运算符的优先级来考虑对它

们的使用, 否则运算的结果就可能不正确。

各种类型运算符的优先级有一定的规律(可参照 2.1 节中关于运算符的优先级的内容)可循, 如果难以确定各运算符的优先级, 可以在表达式中适当地加上一些小括号来避免意外结果的产生。另外, 在运算符的两边有意加上若干个空格, 也可增强表达式的清晰度。

例如, 对于表达式 $(M) ?(a++): (a--)$, 其中的运算符 $++$ 、 $--$ 的优先级相同, 且都高于条件运算符 $(?:)$, 所以这个表达式也可写成 $M ?a++:a--$, 其作用与原表达式完全相同, 但清晰度就远不如原表达式了。

2.2.2 运算符的结合性

运算符的结合性其实是一个经常处理的问题, 如常说的“负负得正”, 对于表达式“ $--5$ ”(为了与 C 语言的自减运算符区分, 两个负号之间特地用了一个空格, 这个表达式与 $-(-5)$ 等价, 即等于 5), 负号作为一个运算符, 显然后面不能直接跟一个运算符, 而应该跟一个运算量, 这样一来, 就应该先处理后面(右边)的负号, 也就是说后面的负号与运算量 5 结合, 得到一个“ -5 ”, 在此基础上, 再运算前一个(左边)的负号, 得结果为 5。这里就存在一个先计算哪一个负运算的问题, 显然两个运算符的优先级相同(同一个运算符), 此时, 便根据运算符的结合性来决定先运算哪一个, 从上面的分析可以看出, 负运算是自右向左结合的。

在运算符的结合性不是很清楚的情况下, 可以使用适当地添加小括号或适当地添加空格的方式来加强表达式的清晰度。

2.2.3 复合赋值运算

复合赋值运算符的存在可以在一定程度上提高程序的编译效率, 但复合赋值运算符的存在会影响程序的清晰度, 尤其对于初学者, 应该限制性地使用这类运算符。

在使用复合赋值运算符时, 一定要弄清楚它们的含义, 其功能是复合赋值运算符左边的变量与右边的表达式进行运算, 其结果再赋值给左边的变量, 因此也称为自反运算。

当复合赋值运算符右边是一个表达式, 而非单个变量时, 需特别注意。如表达式 $a*=b/c$, 该表达式相当于 $a=a*(b/c)$, 而不能简单地看做 $a=a*b/c$, 这两个表达式的结果在某些情况下可能一样, 但有些情况下就不太一样, 例如 a, b, c 都是整型, 不妨设 $a=3, b=1, c=2$, $a*=b/c$ 的结果 a 被赋值为 0, 而 $a=a*b/c$ 的运算结果 a 被赋值为 1。

2.2.4 赋值类型转换

如果赋值运算符连接的左值与右值的数据类型不一致, 在赋值前系统将自动把右值按左值的数据类型进行转换(也可以用强制类型转换的方式, 把右值类型转换为左值类型再进行赋值), 但这种方式仅限于某些数据类型之间, 通常称为“赋值兼容”。

常用的赋值转换规则如下。

- (1) 当右值为实型, 左值为整型时, 把实型数据的小数部分截断。
- (2) 当右值为整型, 左值为实型时, 数值不变, 但以浮点数形式存储于左值中。
- (3) 当右值为 `double` 类型, 左值为 `float` 类型时, 取右值的前 7 位有效数字, 赋值于左值。

(4) 当右值为字符型, 左值为整型时, 将右值进行扩展(带符号数进行符号位扩展, 无符号数进行 0 扩展), 再赋值于左值。此规则同样适用于右值为较短的整型, 而左值为较长的整型的情况。

(5) 当右值为整型, 左值为字符型时, 截取右值的低 8 位, 原封不动地赋值于右值。

2.3 例题分析

例 2.1 以下各项中不合法的用户标识符是()。

A. st.n

B. file

C. Main

D. GO

解: C 语言规定标识符只能由字母、数字与下划线构成, 由此得出答案为 A, 因为它含有非法字母“.”。B 是合法的标识符。答案 C、D 中包含了大写字母, C 是区分字母大小写的, 所以 C、D 不是 C 语言的保留字符, 可以作为合法的标识符。

例 2.2 以下各项中不正确的实型常量是()。

A. 7.375E - 1

B. 0.3048e2

C. - 44.44

D. 123e - 2.5

解: 实型常量有小数形式与指数形式两种表示法。答案中的 C 是小数形式表示的实型常量, 是正确的。指数形式的实型常量由两部分组成, 即尾数与指数, 其中尾数必须是一个合法的小数形式的实型常量, 而指数必须是一个合法的整型常量, 根据以上分析, 答案 D 的指数部分不是一个合法的整型常量, 即是不正确的实型常量, 故选 D。

例 2.3 下列不合法的十六进制整型常量是()。

A. 0xff

B. 0Xabd

C. 0x16

D. 0x345

解: C 语言规定十六进制整型常量以“0x”或“0X”开头, 数字符号范围为 0 至 9、'A'至'F'或'a'至'f'。答案 A 以字母'o'开头, 不符合 C 语言的规定, 是不合法的十六进制数, 故选 A。顺便指出, 以字母('o')开头恰好符合标识符命名规则, 故 A 可以作为合法标识符, 但不是合法的十六进制常量。

例 2.4 以下程序的运行结果是()。

```
main()
{ int a=2;
  printf("%d,%d,%d\n", a--,a+=2, ++a);
}
```

A. 4, 5, 3

B. 1, 3, 4

C. 2, 4, 5

D. 5, 5, 3

解: 自增(减)运算符有前缀与扩展名两种使用形式, 其区别是, 使用前缀方式时, 先实现变量的自增(减)1, 而使用扩展名方式时, 先引用变量的当前值, 然后再自增(减)1。另外, 在此题中还要注意函数参数的计算顺序, 在 Turbo C 2.0 中, 实参的求值自右向左进行。本题是先计算++a, 然后计算 a+=2, 最后计算 a--。根据以上分析, 不难得出正确答案为 D。

例 2.5 若变量已正确定义并赋值, 符合 C 语言语法的表达式是()。

A. a+=b+c=5+d

B. a=7+b+c,a++

C. a=a-8;

D. 4.5%2