

高等学校计算机基础教育规划教材

新编C语言程序设计

黄超 陈龙猛 刘梦琳 郑桂兴 杨中国 编著

清华大学出版社

高等学校计算机基础教育规划教材

新编 C 语言程序设计

黄 超 陈龙猛 刘梦琳 郑桂兴 杨中国 等编著

清华大学出版社

北 京

内 容 简 介

本书是在 C 语言程序设计精品课程建设的基础上,遵循学生的认知规律,系统讲解了 C 语言程序设计的知识、理论与方法。本书共分 12 章,有 9 个附录,附录包括常用的库函数、常见编译错误信息、二级 C 考试大纲与模拟题等丰富的参考资料。

本书的讲解简练、通俗,突出思路分析,强化算法设计,注重编程实践能力的培养,注重知识体系结构的完整性、合理性、实用性,每章都配有丰富多样的习题、上机实验,以及习题参考解答。课程设计单列为一章,进行综合程序的开发与训练。本书还兼顾多层次读者的不同需求,设计了有针对性的学习、应考(等级考试)指导材料。

本书内容齐全,体现教学规律,便于教与学,有一定的创新性。本书可作为高等院校计算机各专业的教材,也可作为各类 C 语言程序设计的培训教材、相关技术人员的技术参考书,还适合作为自学参考书。

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

新编 C 语言程序设计 / 黄超等编著. —北京:清华大学出版社,2012.7

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-28935-7

I. ①新… II. ①黄… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 110754 号

责任编辑:张龙卿

封面设计:徐日强

责任校对:李 梅

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>, 010-62795764

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:23.25

字 数:535 千字

版 次:2012 年 7 月第 1 版

印 次:2012 年 7 月第 1 次印刷

印 数:1~3000

定 价:38.00 元

产品编号:045077-01

前 言

教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具。教材的质量直接影响着教师的教与学生的学,它与教师、学生、教学方法、教育技术手段等一起构成了教学过程的一个统一的整体,是实现教学培养目标的关键因素之一。多年来,在 C 语言程序设计精品课程建设过程中,对课程体系、教学模式、教材结构和内容等进行了研究;本教材是在根据 C 程序设计教学的新发展,吸取很多教材优点的基础上编写而成的。

C 语言程序设计是计算机专业及理工等专业开设的一门专业基础课,也是全国计算机等级考试的二级考试科目之一。作为大学计算机基础教学中第一门系统地讲授程序设计的课程,由于程序设计的思维模式及课程特点与一般传统课程有所不同,加之学时紧、任务重的现实,使不少初学者感到困惑,并产生挫折感。为使初学者顺利地掌握 C 语言程序设计,作者在长期的 C 语言教学过程中,对 C 语言程序设计课程的教学内容、教学方法进行了认真的思考,根据我们多年成熟的一线教学经验积累及研究,按照“体系完整、内容齐全、符合教学规律、便于教与学、有创新、质量高”的宗旨进行了教材的设计与编写。

在课程体系结构安排上,根据教学目的和要求,以任务驱动的方式,尽可能地将概念、知识点与例题结合起来。每章的开始有本章学习目标和导引段,结尾有本章小结,章末配有类型丰富的习题与上机实验。全书共 12 章,从程序设计的基本概念入手,对 C 语言的基本语法元素、运算符与表达式、流程控制语句、函数、构造数据类型、指针、课程设计等内容进行了由浅入深的讲解。本书从软件工程的角度出发,注重程序设计方法和解题思路的讲解与任务驱动教学模式的体现,力求理论结合实际、学以致用。讲解中借助“说明”、“提示”、“注意”、“技巧”、“思考”、“即学即用”、“一题多解”、“试一试”、“优化改进”、“举一反三”、“知识扩展”等教学提示与活动,帮助读者理解、运用、掌握教学内容。针对学生学习 C 语言后普遍感觉提高和综合应用难的问题,在第 12 章,通过进行综合性实例的设计与训练,加深学生对全书内容的理解和综合应用。附录部分,配有各章

习题的参考解答,方便学生的学习与使用,体现了“体系完整、内容齐全、案例实用”的原则。

本书的特点是对程序设计能力的训练进行了分层次的、有针对性的设计,对程序设计学习的三个环节——读程序、写程序、调试程序,进行了详细介绍。通过例题、每章习题和上机实验,对程序学习的三个环节进行有效地实施;通过完成每章的习题,能够加深理解并进一步巩固课堂所学知识与重点;通过各章设置的实验进行实践,帮助学生边学边练、融会贯通、举一反三,逐步深入提高程序设计能力。这些编写内容突出了实用性和易学性,体现了“符合教学规律,便于教、利于学”的编写宗旨。

本书所有的例题均在 Visual C++ 6.0 和 Turbo C 环境下调试通过,并以主流的 C 语言程序集成开发环境 Visual C++ 6.0 进行讲解。为方便教师教学和学生学习,本书提供了配套的教学课件,读者可从清华大学出版社网站 <http://www.tup.com.cn> 下载。

本书由黄超、陈龙猛负责全书的统筹与安排,参加编写的还有刘梦琳、郑桂兴、杨中国、苏万力、王蕊、王宁。具体分工为:第 1、8、12 章及附录 E、G、H、I 由黄超编写,第 2 章及附录 A、B、C 由王蕊编写,第 3 章及附录 F 由陈龙猛编写,第 4 章由苏万力编写,第 5 章由杨中国编写,第 6、7 章及附录 D 由郑桂兴编写,第 9~11 章由刘梦琳编写,王宁老师参与了部分章节的编写与材料整理工作;全书的审核与统稿工作由黄超完成。

本书的编写参考了大量的书籍与网络资源,吸取了很多专家同仁的宝贵经验,同时得到了很多领导的支持及帮助,在此一并表示衷心的感谢!

由于编者水平所限,书中肯定存在疏漏与不足之处,敬请读者不吝提出宝贵意见,以帮助我们今后进一步完善。联系方式:hch1618@126.com

编 者

2012 年 4 月

目 录

第 1 章 初识 C 语言程序设计	1
1.1 C 语言的特点和应用	1
1.2 C 语言的词法单位	3
1.2.1 C 语言的字符集	4
1.2.2 关键字(通常也称为保留字)	4
1.2.3 标识符(名词)	4
1.2.4 C 语言语句	5
1.3 C 语言程序的基本结构	6
1.3.1 简单的 C 语言程序介绍	6
1.3.2 C 程序的结构与书写规则	8
1.4 程序与算法	10
1.4.1 程序与程序设计	10
1.4.2 算法	11
1.4.3 算法的表示	12
1.4.4 结构化程序设计方法	14
1.5 C 语言程序的上机调试	15
1.5.1 C 语言的编译环境	15
1.5.2 C 语言程序上机运行的方法与步骤	16
1.6 本章小结	20
1.7 习题	21
1.8 上机实验	23
实验项目 VC++ 6.0 的使用与程序调试	23
第 2 章 数据类型、运算符和表达式	26
2.1 数据的表示与数据类型	26
2.1.1 常量	26
2.1.2 变量	30
2.2 基本数据类型的运用	31

2.2.1	整型数据	31
2.2.2	实型数据	34
2.2.3	字符型数据	36
2.3	运算符与表达式	37
2.3.1	算术运算符和算术表达式	38
2.3.2	赋值运算符和赋值表达式	41
2.3.3	关系运算符和关系表达式	43
2.3.4	逻辑运算符和逻辑表达式	44
2.3.5	位运算符和位运算	45
2.3.6	其他运算符和表达式	47
2.4	常用数学库函数的使用	48
2.5	本章小结	49
2.6	习题	50
2.7	上机实验	52
第 3 章	顺序结构程序设计	54
3.1	C 语句概述	54
3.2	C 语言的输入与输出	56
3.3	字符数据的输入与输出	57
3.3.1	字符输出函数 putchar()	57
3.3.2	字符输入函数 getchar()	58
3.4	格式输入与输出函数	59
3.4.1	格式输出函数 printf()	59
3.4.2	格式输入函数 scanf()	64
3.5	顺序结构程序设计举例	68
3.6	本章小结	70
3.7	习题	70
3.8	上机实验：输入/输出操作及顺序结构程序设计	73
第 4 章	选择结构程序设计	77
4.1	if 语句	77
4.1.1	单分支结构	77
4.1.2	双分支结构	78
4.1.3	多分支结构	79
4.1.4	if 语句的嵌套	80
4.2	switch 语句	81
4.3	条件运算符	83
4.4	选择结构程序举例	84

4.5	本章小结	87
4.6	习题	87
4.7	上机实验	90
第5章	循环结构程序设计	93
5.1	循环的概念与分类	93
5.2	while 语句	94
5.3	do...while 语句	96
5.4	for 语句	97
5.5	循环的嵌套	100
5.5.1	循环嵌套的流程	101
5.5.2	循环嵌套的应用实例	101
5.6	无条件转向语句	102
5.6.1	break 语句	102
5.6.2	continue 语句	103
5.6.3	goto 语句	104
5.7	应用实例	105
5.7.1	穷举法	105
5.7.2	递推法	106
5.8	本章小结	107
5.9	习题	108
5.10	上机实验	114
第6章	数组	116
6.1	一维数组	116
6.1.1	一维数组的定义	116
6.1.2	一维数组的引用	118
6.1.3	一维数组的初始化	119
6.1.4	应用实例	121
6.2	二维数组	125
6.2.1	二维数组的定义	125
6.2.2	二维数组的引用	126
6.2.3	二维数组的初始化	127
6.2.4	二维数组的应用举例	127
6.3	字符数组与字符串	129
6.3.1	字符数组	130
6.3.2	字符串	131
6.3.3	字符串处理函数	132

6.4	本章小结	135
6.5	习题	136
6.6	上机实验	138
第 7 章	函数	141
7.1	函数的定义与调用	141
7.1.1	函数的分类	141
7.1.2	函数定义的一般形式	143
7.1.3	函数的调用	146
7.1.4	函数的参数传递	148
7.2	函数的嵌套调用与递归调用	151
7.2.1	函数的嵌套调用	151
7.2.2	函数的递归调用	152
7.3	变量的作用域和存储类别	153
7.3.1	变量的作用域	153
7.3.2	变量的存储类别	155
7.4	内部函数与外部函数	157
7.4.1	内部函数	157
7.4.2	外部函数	157
7.5	程序的多文件结构	158
7.6	应用实例	164
7.7	本章小结	167
7.8	习题	167
7.9	上机实验	170
第 8 章	编译预处理	172
8.1	宏定义 #define 命令	172
8.1.1	宏定义	172
8.1.2	带参数的宏替换	176
8.2	文件包含	177
8.3	条件编译	179
8.4	本章小结	181
8.5	习题	182
8.6	上机实验	185
第 9 章	指针	187
9.1	地址与指针类型	187
9.1.1	地址及取地址运算	187

9.1.2 指针类型与指针运算	189
9.2 指针变量	190
9.2.1 指针变量的定义	190
9.2.2 指针变量的引用	191
9.3 指针与数组	195
9.3.1 指向数组元素的指针	195
9.3.2 用指针法引用数组元素	196
9.3.3 多维数组与指针	197
9.4 指针与字符串	201
9.5 指针与函数	204
9.5.1 指针变量作函数的参数	204
9.5.2 指向函数的指针变量	209
9.5.3 指针型函数	211
9.6 指针型数组	213
9.7 多级指针	216
9.8 本章小结	217
9.9 习题	218
9.10 上机实验: 指针的应用	222
第 10 章 结构体、共用体和枚举类型	224
10.1 结构体类型	224
10.1.1 结构体类型的定义	224
10.1.2 结构体变量的说明与引用	225
10.1.3 位段	229
10.2 结构体数组	231
10.2.1 结构体数组的定义与初始化	231
10.2.2 应用举例	232
10.3 结构体与指针	234
10.3.1 结构体类型的指针变量	235
10.3.2 指向结构体数组的指针	236
10.3.3 结构体类型变量作函数的参数	237
10.4 动态数据结构 —— 链表	238
10.4.1 链表的概念与分类	238
10.4.2 动态内存分配函数	239
10.4.3 链表的建立与操作	240
10.5 共用体	245
10.5.1 共用体类型的定义与变量说明	245
10.5.2 共用体变量的引用	246

10.6	枚举类型	247
10.7	用 typedef 说明一种新类型名	250
10.8	本章小结	252
10.9	习题	254
10.10	上机实验：结构体的应用	257
第 11 章	文件操作	259
11.1	文件概述	259
11.2	文件类型与指针	261
11.3	文件的打开与关闭	262
11.3.1	文件的打开函数	262
11.3.2	文件的关闭函数	264
11.4	文件的输入与输出	264
11.4.1	字符读写函数	265
11.4.2	格式化读写函数	268
11.4.3	字符串读写函数	271
11.4.4	块读写函数	274
11.5	文件的定位和随机读写	276
11.5.1	rewind 函数	276
11.5.2	fseek() 函数	276
11.5.3	ftell() 函数	277
11.6	出错检测函数	278
11.6.1	ferror() 函数	278
11.6.2	clearerror() 函数	278
11.7	本章小结	278
11.8	习题	279
11.9	上机实验：文件的使用	283
第 12 章	课程设计	285
12.1	课程设计的目的与要求	285
12.2	课程设计的选题与实施过程	286
12.2.1	选题	286
12.2.2	实施过程	287
12.3	课程设计的规范	289
12.3.1	编程规范	289
12.3.2	课程设计报告的内容	290
12.4	课程设计参考题目	291
12.5	本章小结	295

附录 A C 语言的关键字	296
附录 B 常用字符与 ASCII 码表	298
附录 C C 语言运算符	300
附录 D C 常用库函数	302
附录 E C 程序的调试与测试	310
附录 F C 语言常见编译、链接错误提示信息	314
附录 G 全国计算机等级考试二级 C 语言考试大纲	321
附录 H 全国计算机等级考试二级 C 语言(笔试)模拟题	324
附录 I 习题参考答案	335
参考文献	360

第 1 章 初识 C 语言程序设计

本章学习目标

- 理解 C 语言的相关概念, C 语言的词法记号。
- 掌握 C 语言程序的构成、函数模块结构和书写风格。
- 掌握 C 语言程序的上机运行过程。
- 理解算法的概念、性质及表示方法。

目前,计算机已经广泛应用到我们工作、生活的方方面面,成为人们处理信息的一种必备的重要工具。计算机之所以能够解决不同领域的问题,是因为计算机上装配了各种类型的软件(程序);计算机系统的工作是由程序来控制的,程序则是用程序设计语言根据问题的需求设计、编写的。

C 语言是在 20 世纪 70 年代初问世的,它是一种广泛流行的计算机高级语言。C 语言提供了直接操作计算机硬件的功能,可以直接访问内存的地址,进行位(bit)一级的操作,可以方便地嵌入汇编语言代码,具备低级语言的特点;它既可以编写应用软件,也可以编写系统软件。在 TIOBE 世界编程语言排行榜中(体现了编程语言的流行趋势与热门程度),C 语言的流行度一直居排行榜前两位。

1.1 C 语言的特点和应用

1. 计算机语言的分类

利用计算机解决实际问题,必须将解决问题的方法步骤转化为计算机程序,这就需要用程序设计语言(也称计算机语言)来描述。自从 1946 年 2 月世界上第一台电子计算机 ENIAC 在美国的宾夕法尼亚大学诞生以来,计算机语言(Computer Language)的发展十分迅速,曾出现过机器语言(也称二进制代码语言)、汇编语言(也称助记符语言,出现于 20 世纪 50 年代中期)和高级语言(也称为算法语言,出现于 20 世纪 50 年代中后期)等。现在,程序设计语言及编程环境正向第 4 代(面向对象语言及可视化编程环境)方向发展。机器语言和汇编语言是面向机器的,可移植性差,并且难学,不易推广,习惯上将这两代称为低级语言。高级语言接近于自然语言与数学语言,语言的语法形式类似于英文,它独立于计算机硬件系统,不再面向机器,可移植性好。因为是远离硬件的直接操作,使得普通人易于理解与使用,这极大地促进了计算机应用的普及。

高级语言种类繁多,每一种语言都有自己的适用领域及自身的长处和缺点;如面向过程的 ALGOL (Algorithmic Language)、Basic (Beginner's All-purpose Symbolic Instruction Code)、FORTRAN (Formula Translator)、COBOL (Common Business Oriented Language)、C 语言及结构化程序设计语言 Pascal;面向对象的 Java、Visual Basic、Visual Foxpro、C++、SQL 语言等均为常用的高级语言。用高级语言编程,不必过多考虑机器的内部构造和不同机器的特点,只要按照解题过程(即算法)写出相应的程序,在装配了该种高级语言编译程序的计算机上就能执行所编的程序。

2. C 语言的发展

C 语言是由贝尔实验室的丹尼斯·利奇(Dennis. M. Ritchie, 1983 年图灵奖获得者),于 1972 年至 1973 年间在 B 语言的基础上设计出来的。C 的命名体现了 C 语言的发展过程,C 取自 B 语言的前身 BCPL 的第二个字母。C 语言保持了 BCPL (Basic Combined Programming Language)语言和 B 语言的精练且接近硬件的优点,又克服了它们过于简单、数据无类型的缺点。1973 年 Ken Thompson 和 D. M. Ritchie 成功地将 UNIX 操作系统的 90%以上用 C 语言改写成功(即 UNIX 第 5 版)。C 语言因 UNIX 操作系统而诞生,UNIX 也因 C 语言而得以快速推广,两者相辅相成。1978 年开始,C 语言独立于 UNIX 和 PDP,被先后移植到大、中、小及微型机上。C 语言很快便风靡全世界,成为应用广泛、影响深远的程序设计语言之一。C 语言一方面在计算机程序语言研究方面具有一定价值,由它引出了许多后继语言,C++、Java 都是以 C 语言为基础发展起来的后起之秀;另一方面,C 语言对整个计算机工业和应用的发展都起到了很重要的推动作用。

3. C 语言的特点

C 语言为何能成为一种广为流行的、极受欢迎的程序设计语言?从语言体系和结构上看,它与 Pascal、ALGOL 60 等高级语言相类似,是结构化程序设计语言;但从用户应用、实现难易程度、程序设计风格等方面来看,C 语言的主要特点如下。

(1) 语言简洁、紧凑、使用灵活。在源程序表示方法上,与其他语言相比,一般功能上等价的语句,C 语言书写的源程序更直观、精练。

(2) 运算符丰富,表达能力强。C 语言的运算符共有 34 种之多,括号、赋值、强制类型转换等均作为运算符处理。C 语言可直接处理字符、访问内存物理地址、进行位操作,可以直接对计算机硬件进行操作,因此 C 语言既具有高级语言功能,又具有低级语言的许多功能,可用来编写系统软件。鉴于 C 语言的这种双重性,故有人把 C 语言称为“高级语言中的低级语言”或“中级语言”,意为兼有高级语言和低级语言的特点。

(3) 数据类型丰富。C 语言具有现代化语言的各种数据结构。C 语言的数据类型包括整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型等。且具有数据类型的构造能力,因此,能用来实现各种复杂的数据结构的运算。

(4) 结构化。C 语言具有结构化控制语句(如 if...else 语句、do...while 语句、for 语句等),并以函数作为程序的模块单位,能更好地实现程序的模块化。

(5) 生成的目标代码质量高,程序执行效率高。C 语言程序生成的目标代码一般只比汇编语言程序生成的目标代码效率低 10%~20%。

另外,C 语言还有程序通用性强、可移植性好(与汇编语言相比)等特点,可以方便地在不同操作系统平台之间转换使用。

C 语言也存在一些不足,如运算符和运算优先级过多,不便于记忆;语法定义不严格,如语法限制不太严格、对数组下标越界不作检查等。编程自由度大,编译程序查错纠错能力有限,这就要求编程者自己注意可能的的问题,程序的正确性主要靠人来保证;程序员不能过分依赖 C 编译程序去检查程序。

4. C 语言的应用

C 语言的功能强大、应用范围广,从底层的嵌入式系统、工业控制、智能仪表、编译器、硬件驱动,到高层的行业软件后台服务、中间件等。

C 语言一般用于底层开发,嵌入式开发,或者要求效率高,可移植性的地方。通常,嵌入式系统是一个控制程序存储在 ROM 中的嵌入式处理器控制板。事实上,带有数字接口的设备,如手表、微波炉、录像机、汽车等,都使用嵌入式系统,大多数嵌入式系统都是由单个程序实现整个控制逻辑。

对 MS-DOS 操作系统来说,系统软件和实用程序、Windows 操作系统大部分是用 C 语言编写的;许多著名的系统软件,如 DBASE III PLUS、DBASE IV 也是用 C 语言编写的。C 语言也用于应用软件的开发,如数值计算、文字处理 WORDSTAR、游戏等应用软件是用 C 语言加上一些汇编语言子程序的方法编写的。

C 语言用于就业、升学、计算机技术水平认证。C 语言是为专业程序员设计的。C 语言程序设计是全国计算机等级考试(除 PC 技术外)三级上机考试的限定科目,C 语言也是程序设计比赛、全国中学生奥林匹克信息技术——程序设计竞赛,指定使用的编程语言之一,C 语言程序设计也是计算机专业专升本及一些硕士研究生入学考试的科目。

注意:对一般的编程任务而言,使用其他计算机语言也能胜任,但难易程度、效率存在差异;开发效率还和程序员的语言熟练程度密切相关。编程语言在领域中的运用没有绝对的界限,一个软件系统中,可能是多种语言、技术的结合。一个软件工程师,往往需要掌握多门计算机语言。

1.2 C 语言的词法单位

语言是人类活动与交流的工具,每一种语言都有各自的语法规则及使用规定;计算机语言是一种形式化的人工语言,相对于自然语言,其规模小、语义简单、规范,如 C 语言的单词词汇、句子相对较少,并且用于特定的计算机环境。C 语言与自然语言相比也有字、词、句、段、篇章等概念及自己的语法规则。要用 C 语言编写程序,首先必须学习、掌握 C 语言的词法单位及语法规则。

1.2.1 C 语言的字符集

任何语言都是由一些基本符号组成的,C 语言字符集由字母、数字、空格、标点和特殊字符组成。

- (1) 英文字母(52 个),包括大写字母 A、B、…、Z 和小写字母 a、b、…、z。
- (2) 阿拉伯数字(10 个): 0、1、2、3、4、5、6、7、8、9。
- (3) 运算符,+、-、.、*、/、%、=、<、>、?、:、&、()、[]、!、~、^、|等。
- (4) 分隔符#、{}、<>、,、;、"、'、\。分隔符用于分隔单词或程序正文的单词,不表示任何实际的操作,仅用于构造程序。
- (5) 空白符。包括空格、制表符、换行符等,这些字符属于不可打印的字符,在源程序中一般作为字符常量或字符串常量,使用时需要使用转义符号(详见第 2 章)。
- (6) 其他字符。

注意: 以上基本符号在程序设计时,要用西文半角形式,不能用中文全角符号代替。C 语言除了能使用其规定的符号集中的符号外,不得使用任何其他符号。如:数学中的 +、-、 $\times$ 、 $\div$ 、 $\neq$ 、 $\geq$ 、 π 、 $\int$ 、 $\sum$ 、 α 、 β 、 γ 、 δ 、 Δ 等符号都不得在 C 语言程序中直接应用。

1.2.2 关键字(通常也称为保留字)

C 语言符号集中的符号按一定的组合规则形成 C 词汇,关键字是 C 语言中具有特定含义的专用英文词汇。C 语言关键字是由小写字母构成的字符序列,按其作用可以分为以下几类。

- (1) 数据类型说明与修饰符: char double enum float int long short signed struct unsigned union void。
- (2) 语句定义符: break case continue do default else for goto if return switch while。
- (3) 存储类型说明符: auto register static extern。
- (4) 其他: const sizeof typedef volatile。

关键字不允许另作他用,如关键字不能作为自定义的标识符使用。关键字的含义与用法将在后续章节中逐步介绍。

1.2.3 标识符(名词)

标识符是程序员为程序中涉及的变量、常量、函数、数组、文件等对象定义的名字。

C 语言标识符是以字母或下画线开头的字母、数字、下画线的组合。标识符中同一个大写字母与小写字母是不同的字符,如 a 和 A、AB 和 Ab 是互不相同的标识符。例如: X、x1、y、C、count1、_z23 均是合法的标识符。而 3x、x-y、s * T、-3x、 α 、 β 、ex10.2 则不是标识符,或称它们是非法的标识符。

标准 C 对标识符的长度(字符个数)没有限制,不同的 C 编译系统对标识符长度的规

定可能是不同的,使用时要注意。一般至少能识别8个字符,当两个标识符前8位相同时,则被认为是同一个标识符。有的C编译系统能识别32个字符,VC++标识符长度为1~247个字符。

注意:在C语言中,有些标识符具有特殊的含义,称它们为预定义标识符。如主函数名main,库函数名printf、scanf,预处理命令include、define等,它们的使用方法类似于保留字。

定义标识符时还应注意以下约定:

(1) 标识符的命名应尽量有相应的意义,做到“顾名思义”,以便阅读理解,如sum、name、max。

(2) 标识符不能使用C关键字,避免使用C语言中预定义的标准标识符。如float、int、main。

(3) 避免使用易混淆的字符,如l与I,o与0,no和n0,I1和l1,以减少出错。

【即学即用】判断下面哪些是合法的标识符,哪些不是合法的标识符,为什么?

MyFile 94Salary @Mn3 x2+y3 3xy void M.J.ttY \$xz_1238 for f3.5

【知识扩展】匈牙利命名法

编程时采用好的命名方法,可使标识符便于记忆、容易理解,提高程序的可读性。Microsoft采用匈牙利命名法来命名Windows API函数和变量。

匈牙利命名法是由匈牙利程序员Charles Simonyi发明的,标识符按属性、类型+对象描述的顺序组合而成。标识符以一个或者多个小写字母开头作为前缀,用这些字母来指定数据类型;前缀以后是一个或者多个第一个字母大写的单词,这些单词能清楚地指明对象的用途。

常见的数据类型的前缀有n或i表示整型int,l表示long,p或者lp表示指针,sz或者str表示字符串,u表示unsigned,f表示float,d表示double,m_表示成员。如m_szPersonName表示人名的成员变量,数据类型是字符串型。

1.2.4 C语言语句

语句是组成程序的基本单位,C语言语句用来向计算机发出操作指令。语句的有机组合能实现特定的计算处理功能。一条完整的C语句必须以“;”结束,“;”是一个语句不可缺少的部分,如“sum=a+b;”。

该赋值语句的作用是将a与b相加的和赋给变量sum。

当程序在计算机上执行时,程序中的语句用来完成具体的操作并控制计算机的执行流程,但是程序并不一定按照语句序列的书写顺序来执行。如果程序中的语句是按照书写顺序执行的,称其为“顺序结构”;如果语句是按照当时的某个条件来决定是否执行,则称其为“选择结构”;如果某些语句要反复执行多次,则称其为“循环结构”。

C语言语句可以从不同的角度进行分类,根据语句的复杂程度可以分为简单句(如“sum=a+b;”)和复杂语句,复杂语句的结构比较复杂,由几个简单句复合构成,如:

```
if (x>y) z=x; else z=y;
```