



A COURSE
IN
C PROGRAMMING

A COURSE
IN
C PROGRAMMING

C 语言程序设计教程

王庆延 姚 华 邓 斌 主编

A COURSE
IN
C PROGRAMMING



ZHEJIANG UNIVERSITY PRESS
浙江大學出版社

C 语言程序设计教程

主 编 王庆延 姚 华 邓 斌
主 审 程从从

浙 江 大 学 出 版 社

图书在版编目(CIP)数据

C 语言程序设计教程 / 王庆延, 姚华, 邓斌主编.
杭州: 浙江大学出版社, 2006. 10
ISBN 7-308-04941-8

I. C... II. ①王... ②姚... ③邓... III. C 语言 -
程序设计 - 高等学校 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 111627 号

C 语言程序设计教程

王庆延 姚 华 邓 斌 主编

责任编辑 徐宝澍

出版发行 浙江大学出版社

(杭州天目山路 148 号 邮政编码 310028)

(E-mail: zupress@mail.hz.zj.cn)

(网址: <http://www.zjupress.com>)

排 版 浙江大学出版社电脑排版中心

印 刷 浙江省煤田地质局制图印刷厂

开 本 787mm × 1092mm 1/16

印 张 16

字 数 410 千字

版 次 2006 年 10 月第 1 版 2006 年 10 月第 1 次印刷

书 号 ISBN 7-308-04941-8/TP · 312

定 价 30.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社新世纪事业部电话 (0571) 88273985

C 语言程序设计教程编委会

主 编	王庆延	姚 华	邓 斌				
副主编	吴迎军	熊小梅	王 勇				
编 委	王庆延	姚 华	邓 斌	吴迎军	熊小梅	王 勇	
	李淑珍	肖守柏	余学兵	徐军莉	黄雷鸣	罗 琳	
主 审	程从从						

前 言

近几年来,由于实用型本科和高职高专类的迅速发展,要求参加全国计算机等级考试(新大纲)二级 C 语言程序设计的学生愈来愈多,广大学生迫切希望有一套满足全国计算机等级考试二级 C 语言程序设计的教材和实训作为指导教材。我们为适应这种迅速发展及广大学生的需求,组织了从事多年 C 语言程序设计教学的教师,在认真学习和深刻领会考试大纲的精神实质基础上,编写了这套教材。

这套教材是根据 C 语言程序设计考试大纲的要求,以 C 语言程序设计的基本知识为内容,为学习后期课程奠定高级语言程序设计的基础,重新组织教材内容进行编写。本套教材由两册组成:《C 语言程序设计教程》和《C 语言程序设计实训指南》。

本教材共分 8 章,第 1、2 章由邓斌、吴迎军编写,第 3 章由李淑珍、罗琳编写,第 4 章由王勇、黄雷鸣编写,第 5 章由余学兵、徐军莉编写,第 6 章由熊小梅编写,第 7、8 章由肖守柏、姚华编写。全书由王庆延教授进行统编,由程从从教授主审。

本教材内容丰富,文字精练,着重实际应用,紧扣全国计算机二级 C 语言程序设计的内容,适合作为大学本科生、高职高专各专业学生的 C 语言程序设计教材,也可作为其他专业学习 C 语言程序设计的参考书。

由于时间紧迫,加上编者水平有限,书中定有不少不足之处,欢迎提出宝贵意见。

编 者

2006 年 6 月

目 录

第 1 章 C 语言基础	(1)
1.1 C 语言的发展与特点	(1)
1.1.1 C 语言的发展概况	(1)
1.1.2 C 语言的特点	(2)
1.2 程序设计中的几个基本概念	(3)
1.2.1 程序	(3)
1.2.2 程序设计	(4)
1.2.3 算法	(4)
1.2.4 数据结构	(4)
1.2.5 软件工程的概念	(4)
1.3 C 语言程序的结构	(5)
1.3.1 字符集	(5)
1.3.2 保留字	(6)
1.3.3 标识符	(6)
1.3.4 C 语言的词类	(7)
1.4 C 语言的基本语句	(7)
1.5 C 语言程序的基本结构	(8)
1.6 C 语言程序的编译与执行	(11)
本章小结	(12)
习题 1	(13)
第 2 章 数据类型及运算	(15)
2.1 数据类型基本概念	(15)
2.1.1 C 语言的数据类型	(15)
2.2 常量	(17)
2.2.1 数值常量	(17)
2.2.2 字符常量	(18)

2.2.3 字符串常量	(19)
2.3 变量及其常见类型	(21)
2.3.1 基本数据类型	(22)
2.3.2 变量的定义与初始化	(23)
2.3.3 数据类型间的转换	(24)
2.4 运算符和表达式	(26)
2.4.1 运算符和表达式概述	(26)
2.4.2 赋值运算符和赋值表达式	(27)
2.4.3 算术运算符和算术表达式	(28)
2.4.4 关系运算符和关系表达式	(32)
2.4.5 逻辑运算符和逻辑表达式	(32)
2.4.6 其他运算符	(34)
2.5 位运算	(35)
2.5.1 逻辑位运算	(35)
2.5.2 移位运算	(36)
2.5.3 位运算与赋值运算组成复合赋值运算符	(37)
本章小结	(37)
习题 2	(38)
第 3 章 基本结构及其实现	(41)
3.1 结构化程序设计	(41)
3.1.1 程序的基本结构	(41)
3.2 顺序结构	(43)
3.2.1 赋值语句	(43)
3.2.2 输入输出库函数	(44)
3.2.3 顺序程序结构程序设计举例	(54)
3.3 选择结构	(56)
3.3.1 选择结构 if — else 语句	(56)
3.3.2 switch 分支结构	(66)
3.4 循环结构	(71)
3.4.1 while 循环语句	(71)
3.4.2 do while 循环语句	(75)
3.4.3 for 循环语句	(76)
3.4.4 三种循环语句的比较	(80)
3.4.5 辅助语句	(81)
3.4.6 多重循环	(84)
3.5 goto 语句	(88)
3.5.1 goto 语句	(88)
3.5.2 用 goto 语句组成循环	(88)
本章小结	(89)

习题 3	(89)
第 4 章 数 组	(98)
4.1 一维数组	(98)
4.1.1 一维数组的定义	(98)
4.1.2 一维数组的存储形式	(99)
4.1.3 一维数组元素的引用	(99)
4.1.4 一维数组的初始化	(101)
4.1.5 一维数组的应用	(101)
4.2 二维数组	(104)
4.2.1 二维数组的定义	(104)
4.2.2 二维数组的存储形式	(105)
4.2.3 二维数组元素的引用	(105)
4.2.4 二维数组的初始化	(108)
4.2.5 二维数组的应用	(108)
4.3 字符数组	(110)
4.3.1 字符数组处理字符串	(110)
4.3.2 多个字符串的存储	(111)
4.3.3 字符串的输入与输出	(112)
4.3.4 字符串处理函数	(115)
4.4 综合应用举例	(117)
本章小结	(125)
习题 4	(125)
第 5 章 函 数	(128)
5.1 概述	(128)
5.1.1 结构化程序设计简介	(128)
5.1.2 函数的分类	(129)
5.2 函数的定义与调用	(129)
5.2.1 函数的定义	(129)
5.2.2 函数的调用与返回	(131)
5.2.3 形式参数和实际参数	(133)
5.2.4 对被调函数的声明	(134)
5.3 函数的嵌套调用与递归调用	(136)
5.3.1 函数的嵌套调用	(136)
5.3.2 函数的递归调用	(138)
5.4 数组作函数的参数	(139)
5.4.1 数组元素作为函数参数	(139)
5.4.2 数组名作为函数的参数	(140)
5.4.3 多维数组元素作为函数参数	(143)
5.5 变量的作用域与存储属性	(143)

5.5.1 变量的作用域	(144)
5.5.2 变量的存储属性	(147)
5.5.3 存储类别汇总	(152)
5.6 内部函数和外部函数	(153)
5.7 编译预处理	(155)
5.7.1 宏定义	(155)
5.7.2 文件包含	(158)
本章小结	(159)
习题 5	(159)
第 6 章 指 针	(162)
6.1 指针的概念	(162)
6.1.1 指针变量的定义与使用	(163)
6.1.2 指针运算	(165)
6.2 指针与数组	(166)
6.2.1 指向一维数组的指针变量	(166)
6.2.2 指向二维数组的指针变量	(170)
6.2.3 指针数组	(175)
6.2.4 多级指针	(176)
6.3 指针与函数	(178)
6.3.1 指针作为函数的参数与返回指针值的指针函数	(178)
6.3.2 指向函数的指针	(185)
6.3.3 宏命令行参数的主函数	(189)
6.4 程序设计举例	(190)
本章小结	(192)
习题 6	(192)
第 7 章 结构体与共用体	(197)
7.1 结构体类型的定义	(197)
7.1.1 结构体类型的引入	(197)
7.1.2 结构体类型的定义	(198)
7.1.3 定义结构体类型的变量	(199)
7.1.4 结构体类型变量成员的引用	(201)
7.1.5 结构体类型变量的初始化	(203)
7.2 结构体数组	(204)
7.2.1 结构体数组的定义	(204)
7.2.2 结构体数组初始化	(204)
7.2.3 结构体数组应用举例	(205)
7.3 结构体与指针	(206)
7.3.1 指向结构体变量的指针	(207)
7.3.2 指向结构体数组的指针	(208)

7.4	共用体	(209)
7.4.1	共用体概念	(209)
7.4.2	共用体成员的引用方式	(211)
7.4.3	共用体变量特点	(212)
7.5	枚举类型	(214)
7.6	用 typedef 定义类型	(215)
	本章小结	(217)
	习题 7	(218)
第 8 章	文 件	(220)
8.1	文件概述	(220)
8.2	文件类型指针	(221)
8.3	文件的打开与关闭	(222)
8.3.1	文件的打开(fopen 函数)	(222)
8.3.2	文件的关闭(fclose 函数)	(223)
8.4	文件的读写	(224)
8.4.1	读写一个字符	(224)
8.4.2	读写一个字符串	(226)
8.4.3	读取数据字段	(227)
8.4.4	文件的格式化读写	(229)
8.5	文件的定位	(231)
8.5.1	重定位函数 rewind	(231)
8.5.2	fseek 函数和随机读写	(232)
8.5.3	位置函数	(235)
8.6	出错的检测	(235)
	本章小结	(236)
	习题 8	(236)
附录 I	常用字符与 ASCII 代码对应表	(238)
附录 II	运算符与运算结合性	(239)
附录 III	Turbo C 常用库函数	(240)

第1章

C语言基础

本章要点: C语言是一种通用的程序设计语言,它具有高效、灵活、功能丰富、表达力强、移植性和应用广泛等特色,许多操作系统都是用C语言编写的。本章介绍了C语言的发展以及C语言的特点,介绍了C语言的字符集、基本词类、基本句类和程序的基本结构以及C语言的保留字、关键字,并通过一些实例讲述了C语言程序的结构。

1.1 C语言的发展与特点



1.1.1 C语言的发展概况

C语言是目前国际上最流行的一种结构化的程序设计语言,在系统软件和应用软件的开发中发挥着非常重要的作用。C语言是程序员必须掌握的计算机语言之一,也是掌握C++和VC++的重要基础。至今,它深受广大程序设计者的欢迎。C语言是在B语言的基础上发展起来的,1970年美国贝尔实验室的Ken Thompson以BCPL(Basic Combined Programming Language)语言为基础,作了进一步简化,设计出了一个简单而且很接近硬件的B语言(取BCPL的第一个字母),但B语言过于简单,功能有限。1972年到1973年间,贝尔实验室的D. M. Ritchie在B语言的基础上设计出了C语言(取BCPL的第二个字母)。C语言既保持了BCPL和B语言的优点,同时又克服了它们的缺点。最初的C语言只是为描述和实现UNIX操作系统提供一种工作语言而设计的。1973年,K. Thompson和D. M. Ritchie两人合作把UNIX的90%以上用C改写,即UNIX第5版。(原来的UNIX操作系统是1969年由美国贝尔实验室的K. Thompson和D. M. Ritchie开发成功的,是用汇编语言写的。)

后来,C语言多次作了改进,但主要还是在贝尔实验室内部使用。直到1975年UNIX第6版公布后,C语言的突出优点才引起人们普遍注意。1977年出现了不依赖于具体机器的C语言编译文本《可移植C语言编译程序》,使C语言移植到其他机器时所需做的工作大

大简化了,推动了 UNIX 操作系统迅速地在各种机器上实现。1978 年以后,C 语言已先后移植到大、中、小、微型机上,已独立于 UNIX 了。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。

以 1978 年发表的 UNIX 第 7 版的 C 编译程序为基础,Brian W. Kernighan 和 Dennis. M. Ritchie(合称 K & R)合著了影响深远的名著《The C Programming Language》,这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,它被称为标准 C。1983 年,美国国家标准协会(American National Standard Institute,ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。K & R 在 1988 年修改了他们的经典著作《The C Programming Language》,按照 ANSI C 标准重新写了该书。1987 年,ANSI 又公布了新标准——87ANSI C。目前流行的 C 编译系统都是以它为基础的。目前最流行的 C 语言有以下几种:

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT & T C

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,使之更加方便、完美。本教材以 ANSI C 为基础,采用 Turbo C 2.0 版本来讲解 C 语言。

1.1.2 C 语言的特点

C 语言之所以能存在和发展,并具有很强的生命力,而且成为最受欢迎的语言之一,总是有其不同于(或优于)其他语言的特点。许多系统软件都是用 C 语言编写的。用 C 语言加上一些汇编语言子程序,就更能显示 C 语言的优势了,如 UNIX、MS-DOS、Microsoft Windows 及 LINUX 等就是用这种方法编写的。C 语言的主要特点如下:

1. 中级语言

程序设计语言是一组用来定义计算机程序的语法规则,程序设计语言分为两类:一类是低级语言;另一类是高级语言。低级语言的主要特点是直接依赖计算机的硬件系统的,一种机型能识别的低级语言,另一种机型可能完全不识别。低级语言包括机器语言和汇编语言。高级语言是比较接近自然语言和数学语言的程序设计语言,它非常符合人类的逻辑思维,不依赖计算机硬件系统,用高级语言编写的程序几乎可以不加修改地不同机型上运行。C 语言被称为是中级语言,这并不是说 C 语言功能差或使用困难。恰恰相反,C 语言把高级语言中的语句和基本结构等先进的思想与汇编语言的控制和灵活性有机地结合起来。而且 C 语言还具有汇编语言中直接对位、字节和地址的操作。因此 C 语言是具有低级语言功能的高级语言,所以人们把 C 语言称为是中级语言。

2. 结构化语言

结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。结构化的语言以函数和模块来组织代码,虽然从严格的学术观点上看,C 语言是块结构语言,但是它还是常被称为结构化语言。这是因为它在结构上类似于 ALGOL、Pascal 和 Modula-2(从技术上讲,块结构语言允许在过程和函数中定义过程或函数。用这种方法,全局和局部的概念

可以通过“作用域”规则加以扩展,“作用域”管理变量和过程的“可见性”。因为 C 语言不允许在函数中定义函数,所以不能称之为通常意义上的块结构语言)。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。结构化语言比非结构化语言更易于程序设计,用结构化语言编写的程序的清晰性使得它们更易于维护。在 C 语言中,是以函数形式提供给用户的,函数是一种构件(程序块),是完成程序功能的基本构件,这些函数可方便的调用。函数允许一个程序的诸任务被分别定义和编码,使程序模块化。这种方式可使程序层次清晰,便于使用、维护以及调试,并具有多种循环、条件语句控制程序流向(这些控制语句将在第 3 章进行详细介绍),从而使程序完全结构化。

3. 可移植性

同其他高级语言一样,C 语言也具有较高的程序可移植性,可移植性表示为某种计算机写的软件可以用到另一种机器上去。C 语言程序本身基本上可以不作任何修改,就能运行在各种不同型号的计算机和各种操作系统环境上。C 语言还有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX 等,本身不依赖于机器的硬件系统。

4. C 语言是一种模块化程序设计语言,适合大型软件的开发、研制和调试。

5. C 语言提供了大量的库函数供调用,大大地简化了程序设计工作。

6. 运算符丰富

C 语言除了具有其他高级语言所具有的运算符外,还具有 C 语言特有的运算符,比如增量运算符、赋值运算符、逗号运算符、条件运算符和强制类型转换运算符等,共有 34 种运算符。使 C 语言的运算类型极为丰富,表达式类型多样化,灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

7. 语言简洁、紧凑、灵活

C 语言一共只有 32 个关键字和 9 条控制语句,且源程序书写格式自由,一般都用小写字母表示。

1.2 程序设计中的几个基本概念



1.2.1 程序

所谓程序,就是一系列遵循一定规则和思想并能正确完成指定工作的代码(也称为指令序列)。一般一个计算机程序主要包括两部分的内容:一是描述问题中的每个对象及它们之间的关系,它涉及到数据结构的内容;二是描述这些对象进行处理的规则,就是求解某个问题的算法。著名的计算机科学家沃思(Nikiklaus Wirth)提出了一个有名的公式:

程序 = 数据结构 + 算法 + 程序设计方法 + 语言工具和环境

以上四个方面是设计程序时应综合考虑的内容。在这四个方面中,算法是灵魂,数据结构是加工对象,语言是工具,编程需要采用合适的方法。算法是解决“做什么”和“怎样做”的问题,程序中的操作语句实际上就是算法的体现。

1.2.2 程序设计

所谓程序设计,就是根据计算机要完成的任务,提出相应的需求,在此基础上设计数据结构和算法,然后编写相应的程序代码并测试该代码运行的正确性,直到能够得到正确的运行结果为止。程序设计的关键之一,是解题的方法与步骤,是算法。学习高级语言的重点,就是掌握分析问题、解决问题的方法,就是锻炼分析、分解,最终归纳、整理出算法的能力。与之相对应,某种具体语言,如 C 语言的语法工具,它是算法的一个具体实现。所以在高级语言的学习中,一方面应熟练掌握该语言的语法,因为它是算法实现的基础,另一方面必须认识到算法的重要性,加强思维训练,以写出高质量的程序。

1.2.3 算法

所谓算法,就是解决某一问题的方法。一般来说,一个算法就是一个步骤的有穷序列。是计算机科学的一个基本概念,也是程序设计的一个核心概念。

通常算法的建立过程是逐步求精的,一般是先给出粗略的计算步骤框架,然后对框架中的具体内容进行逐步细化,添加必要的细节,使之成为较为详细的描述,当然,细化可能做不到一步到位,因此还要进行更进一步的细化,直到能够把需求通过编程语言完全描述为止。描述算法的常用工具是流程图,也称程序框图,流程图是算法的图形描述,它往往比程序更直观,容易阅读和理解。

1.2.4 数据结构

数据结构是指数据对象及其相互关系和构造方法,程序中数据结构描述了程序中数据间的组织形式和结构关系。

数据结构与算法密不可分,一个好的数据结构,将使算法简单化;只有明确了问题的算法,才能较好地设计数据结构,因此两者是相辅相成的。

对于计算机程序而言,其构成与数据结构关系密切,程序在实现算法的同时,还必须完整地体现作为算法操作对象的数据结构,对于复杂问题的求解,常常会发现由于对数据的表示方式和结构的差异,对该问题的抽象求解算法也会不完全相同,一个问题的求解,允许存在不同的数据结构,而且按不同算法编写的操作代码,其执行效率是不一样的。

1.2.5 软件工程的观念

软件工程是开发、运行、维护和修复软件的系统方法,是指导计算机软件开发和维护的一门工程学科。采用工程的概念、原理、技术和方法来开发与维护软件,把经过时间考验并证明正确的管理技术和当前能够得到的最好的技术方法结合起来,经济地开发出高质量的软件并有效地维护它,这就是软件工程。软件工程包括三个要素:方法、工具和过程。

方法为软件开发提供了做的技术,如项目计划与估算、数据结构、算法设计、编码以及维

护等。工具为软件工程方法提供了软件支撑环境。过程则是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。软件工程就是包含上述方法、工具及过程在内的步骤。开发大型应用程序,就应从软件工程的思想出发,对需求进行系统性的分析与设计。读者在学习 C 语言的同时,必须树立软件工程的思维方法,系统、全面分析每一个细小问题,掌握编程技巧。

1.3 C 语言程序的结构

汉语是一种自然语言,学习汉语要首先学习单字,然后学习单词,再学习语句的构成规则,最后学习用语句构成文章。C 语言是一种计算机语言,也要先学习基本字符、基本词类,然后再学习语句的构成规则,最后学习用语句构成程序。

1.3.1 字符集

在 C 语言程序中允许出现的所有基本字符的组合称为 C 语言的字符集。C 语言的字符集就是 ASCII 字符集,主要分为下列几类:

- (1) 大小写英文字母(a~z 和 A~Z 共 52 个)
- (2) 0~9 的数字字符
- (3) 键盘符号(33 个),参见表 1-1。

表 1-1 键盘符号表

符号	中文含义	符号	中文含义	符号	中文含义
~	波浪号	)	右圆括号	:	冒号
`	重音号	_	下划线号	;	分号
!	感叹号	-	减号	"	双引号
@	a 圈号	+	加号	'	单引号
#	井号	=	等号	<	小于号
\$	美元号		或符号	>	大于号
%	百分号	\	反斜杠	,	逗号
^	异或号	{	左花括号	.	小数点
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	斜杠
(	左圆括号	]	右方括号		空格符号

(4) 转义字符

转义字符是以“反斜杠字符(\)”开始,后跟单个字符或若干八进制、十六进制字符组成的一个字符,转义字符具有特定的含义,不同于字符原有的意义,故称“转义”字符,通常用来表示键盘上的控制代码或特殊符号。例如回车换行(\n)、响铃符号(\a)等,详细内容在第 2 章中说明。

1.3.2 保留字

保留字是指由 C 语言规定的具有特定意义的字符串,通常也称关键字。主要用于构成语句,进行存储类型和数据类型定义。用户定义的标识符不能与关键字相同。C 语言的关键字分为以下几类:

C 语言的保留字都用小写英文字母表示,共有 31 个保留字,如表 1-2 所示。

表 1-2 C 语言的保留字

auto	自动	break	中止	case	情况
char	字符	continue	继续	const	常量
default	缺省	do	做	double	双精度
else	否则	enum	枚举	extern	外部
float	浮点	for	对于	goto	转向
int	整型	if	如果	long	长
register	寄存器	return	返回	short	短
signed	有符号	sizeof	求字节数	static	静态
struct	结构	switch	开关	typedef	类型定义
union	共用	unsigned	无符号	void	空
volatile	可变的	while	当		

1.3.3 标识符

标识符是用户自定义的一种字符序列,通常用来表示程序中需要辨认的对象名称,如符号常量、变量、数组、函数等对象的名字。标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(_)组成的字符串,并且其第一个字符必须是字母或下划线。除库函数的函数名由系统定义外,其余都由用户自定义。ANSI C 规定标识符长度不得大于 32 个字符,而微型机中通常是前 8 个字符有效。

以下标识符是合法的:

a, x, _3x, BOOK1, Int, sum5

以下标识符是非法的:

3s 以数字开头

s * T 出现非法字符 *

-3x 以减号开头

bowy - 1 出现非法字符 - (减号)

在使用标识符时还必须注意以下几点:

(1) 在标识符中,大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

(2) 标识符主要用来命名,所以应选取有意义的标识符,以便阅读理解,做到“顾名思义”。

(3) C 语言规定,用户定义的标识符不能是 C 语言规定的保留字。

1.3.4 C 语言的词类

C 语言的词类主要分为以下几种:

(1) 常量

在程序运行中其值不发生变化的数据,例如:10、-12、12.5 等。

(2) 变量

用来存放程序运行中变化的数据,例如:输入的原始数据、中间结果、最终结果等。

(3) 运算符

用来表示简单加工计算的符号,例如:+(加)、-(减)、*(乘)、/(除)等。

(4) 函数调用

形如“函数名(实际参数表)”的式子,它代表调用指定函数后获得的结果。

(5) 表达式

用常量、变量、函数调用、运算符组成的式子,用来表示简单的加工计算。

(6) 保留字(又称关键字)

在程序或语句中,用来表示特定语法含义的英语单词。

上述词类的构成规则将在后续章节中详细介绍。

1.4 C 语言的基本语句

C 语言的语句主要分为以下几种,并以“;”结束:

(1) 数据定义语句

用来定义程序中使用的各种能存放数据的对象的名称和特性。

(2) 赋值语句

形如“**变量 = 表达式**”的语句,功能是计算表达式的值并赋值给变量。

(3) 函数调用语句

形如“函数名(实际参数表)”的语句,功能是调用指定函数。

(4) 表达式语句

由任何表达式组成的语句。在 C 语言中,赋值和函数调用都是表达式,所以赋值语句和函数调用语句也是一种特殊的表达式语句。

(5) 流程控制语句

用来控制程序执行过程的语句,如选择控制语句、循环控制语句、中止语句、继续循环语句、返回语句、无条件转移语句等。

(6) 复合语句

用花括号括住的若干个任意语句。

(7) 空语句

无任何操作的语句。