

高等学校计算机基础教育系列教材

C语言程序设计与软件开发基础

施小英 冯矢勇 汤宝骥 编著

上海交通大学出版社



TP312
560

392485

高等学校计算机基础教育系列教材

C 语言程序设计与软件开发基础

施小英 冯矢勇 汤宝骥 编著



上海交通大学出版社

内容简介

本教材全面系统地介绍计算机软件最通用的开发工具：C语言、C语言程序设计和软件开发基础。全书共12章，分别叙述C语言概况、基本数据类型及操作符和表达式、基本的数据输入输出函数的应用、C语言语句、数组、函数、指针、结构体及共用体与枚举、编译预处理和文件；最后概述一些结构化程序设计方法和软件工程。书中配有大量的例子程序，用于阐明C语言语法和演示程序设计的风格。

本书的特点是全面性、实用性和现代性。本书的全面性是既阐述基础语法、程序设计方法，又介绍软件工程的观念；既讲述语言，又演示程序设计风格；既显示C的优点，又点评C的缺点。本书的实用性是在内容上不但满足教学大纲和计算机等级考试的要求，并对C的不易掌握之处加以重点说明，而且指出实际工作时可能出现的致命的错误。本书的现代性是内容符合最新的ANSI C标准，例子程序既适用于Microsoft C和Turbo C的开发环境，也适用于Visual C++和Borland C++的开发环境。

本书适用于从事计算机程序设计人员、软件开发人员和希望以计算机作为工具处理各种事务的人员。阅读本教材的基础是了解计算机组成的基本知识和DOS操作系统的基础知识，而不需要其他高级语言知识。本书可作为高等院校和培训班的教材、计算机等级考试的参考书和软件设计人员的参考书。

C语言程序设计与软件开发基础

上海交通大学出版社出版

(上海市华山路1954号 邮政编码 200030)

新华书店上海发行所发行

上海新华印刷厂印刷

开本：787×1092(毫米) 1/16 印张：14.5 字数：329000

版次：1996年8月 第1版 印次：1996年8月 第1次

印数：1—4000

ISBN 7-313-01713-8/TP·316 定价：18.00元

前 言

通过近 20 年的超高速发展,计算机已经成为现代社会中必不可少的工具。计算机的应用几乎涉及每一个单位、每一种社会活动。而微型计算机以其越来越优异的性能、低廉的价格、诱人的前景继续吸引着大批的队伍投入使用和开发的洪流中。许多优越的应用软件已经为人们所熟悉、所使用。但有一大批人员希望能掌握一种通用的开发工具,以满足各种不同的需要。经过 20 年的应用实践,在过程式的高级语言中, C 语言成为最为理想的通用开发工具。其理由很多,可能最主要的是两点:一是其他过程式的高级语言能够处理的事务 C 语言都可以处理;二是 C 可以通过对资源的管理而开发新的功能。后一点往往是其他过程式高级语言所不能的。C 语言不但可用于编写大型的、系统的软件,而且用于处理日常的各种事务也是很方便的。

但是在过去相当长的时间里,学习 C 语言的队伍是受局限的。原因中重要的一条是掌握 C 的困难性,过去采用的办法是先学习一门 BASIC, PASCAL 或其他高级语言作为过渡,把 C 作为第二语言。但这样的效果并不理想:学员需要学习两门语言的时间,而实际使用只要一门;更加不幸的是在学习 C 的过程中只注重了对两种语言形式上的对比,不能掌握 C 的真正特点;因而编写的程序往往埋下“定时炸弹”。从程序员的角度看,与其他过程式高级语言相比, C 的真正特点是“责任制”不同:其他过程式高级语言是编译系统负责制(程序调试运行中的错误和正确是很直观的);但 C 是程序员负责制(数组下标的使用、指针的移动、动态分配内存的释放等都是程序员的责任)!往往一个 C 程序在编译时没有任何错误,而且运行的结果也“似乎”符合设计要求,但确实是错误的程序,甚至是危险的程序。这种情况不只是发生在初学者身上,也会发生在有经验的程序员身上,甚至在一些书刊的例子中也有出现。这个特点是不可能通过学习其他语言掌握的。除了这一个特点外, C 语言的其他语法,与各种过程式高级语言没有本质上的区别。既然如此,为什么不直接学习 C 呢?当前,一些教育工作者已有这种共识。本教材就是以 C 为第一学习语言安排的,以培养程序员为目的。建议已经有其他高级语言经验的学员,重点要领会 C 的程序员负责

制，否则过去的经验反而会带来误导。

C 程序员的“负担”是很重的，那么为什么要这样设计 C 呢？因为这个沉重的代价可以换得掌握资源的能力，得到最大的编程灵活性。

当前，使用 C 语言的队伍又出现迅速扩大的现象。其原因是 90 年代软件界的几个新动向推动了 this 浪潮：

(1) Windows 的普及和开发 Windows 程序的需求，Windows 提供大量的接口函数是 C 语言的，使用 C 语言开发是基本的手段。

(2) 各种专用软件的接口绝大多数提供 C 语言接口，各种专用软件往往留出接口，让用户有自由发挥的余地，以补充其功能。例如：Novell 网络有 C 函数给用户访问 B-Treive 数据库，AUTOCAD 留有 C 的接口，不少数据库软件也有 C 的接口。

(3) 面向对象技术的兴起和面向对象程序设计语言 C++ 的成熟，无疑为迅速扩大 C 语言的队伍起了很大的推动作用。C++ 的最基础语法使用了 C 的，C++ 的开发系统可以编译和运行 C 程序。有这样的提法：“C++ 是 C 的扩充”。但实际上 C++ 与 C 是不同的两种语言体系，这有些像数学中代数建立在算术基础上，代数不同于算术一样。因此掌握 C 语言是进入 C++ 所必须。

全书共分 12 章。第 1 章介绍程序设计与 C 语言的概况，使学员对 C 编程和上机调试有一个最初步的了解。第 2 章介绍基本数据类型、操作符和表达式，由于操作符和表达式比较接近代数式子，是比较容易理解的，而数据类型是计算机中最常用的概念之一；对这一章要建立这些基本概念，在以后的章节中逐步加深印象而能达到熟练使用。本章还介绍了 C 的词法。第 3 章介绍基本的数据输入、输出函数的应用，有些概念是“提前”介绍了，其目的是为了以后的例子程序所必须。第 4 章阐述 C 的各种语句，我们不但要掌握其语法规则，而且要逐渐地能运用自如，“语句”的概念是过程式高级语言的核心；同时要领会语句的“缩排”编写格式，这是培养程序设计良好风格的基础。第 5 章介绍数组，使用数组便于处理大量的同类型数据；除了阐明数组的概念和基本使用外，还要介绍通过数组处理字符串的特点；最后说明数组越界的情况、危险性和程序员的责任。第 6 章阐述函数，函数是 C 语言的基本模块单元，C 是典型的函数式语言，我们要牢牢掌握使用函数的三步（声明、定义和调用）过程。第 7 章阐明指针的概念和使用，C 的最大特色可以说是指针的使用，程序员的责任也在于管理好指针。第 8 章介绍结构体、共用体与枚举，这是三种便于用户定义的数据类型。第 9 章讨论编译预处理，这不是 C 语言，而是编译系统的指令，但有几个指令在 C 程序设计中是经常使用的。第 10 章介绍文件的概念和如何调用库函数处

理输入和输出事务。第 11 章叙述结构化程序设计方法，这是各种语言设计程序时最基本的方法。第 12 章介绍软件工程概述，以便读者从宏观理解软件的开发过程和技术。书中有大量的例子程序其目的既用于阐明 C 语言语法，同时用于演示程序设计的风格；这些例子程序适用于 Microsoft C 和 Turbo C 开发环境，也适用于 Visual C++ 和 Borland C++ 的开发环境。这些例子程序的代码清单和运行结果是直接通过软件进行排版的，所以避免了笔误和排版错误。

本书的特点是全面性、实用性和现代性。本书的全面性是既阐述基础语法、程序设计方法，又介绍软件工程的概概念；既讲述语言，又演示程序设计风格；既显示 C 的优点，又点评 C 的缺点。本书的实用性是在内容上不但满足教学大纲和计算机等级考试的要求，并对 C 的不易掌握之处加以重点说明，而且指出实际工作时可能出现的致命的错误。本书的现代性是内容符合最新的 ANSI C 标准，例子程序适用于多种最流行的 C/C++ 的开发环境，对 C 的不足还指出当前可更新的方向。

C 的最大缺点是太接近“机器”，“函数”的机制不能适应高速度、高质量开发大型软件的需要、不能满足人们对软件重用性的要求、不能适应变化的用户环境等等。因此，出现了非过程式语言。此外 C 也有不少可改进之处，在 C++ 中已经做到了。

软件的发展始终跟不上硬件的发展，软件危机始终存在。人们期望着更接近自然语言的高级语言的出现。在近期内，C/C++ 可能是我们最好的选择。

本书由施小英副教授、冯矢勇高工和汤宝骥副教授合作编著；由冯矢勇对全文进行统一编排。本书在出版过程中得到许多同行的支持和帮助，作者在此向他们表示衷心的感谢。

本书的作者曾为不同层次的学员讲授过多种计算机语言，编写过多种计算机教材，并开发过应用软件。作者期望利用这些经验和借鉴许多文献中精彩的观点，使本书有助于读者正确掌握 C 程序设计。作者热切希望读者的批评指正和建议，以便于在新版中吸纳。

作 者

1996 年 3 月

序

近年来,随着国民经济和科学技术的发展,计算机应用得到了日益广泛的普及。计算机作为一种工具与文化,在与科学技术各专业的渗透结合中,推动了各学科的发展,逐渐成为各学科的重要组成部分,甚至形成了新的学科方向。这一点正逐渐为人们所认识。因此,由一系列课程组成的计算机基础教育,如同数学、物理、外语那样,已成为高等学校非计算机专业的重要基础课程。培养大批的掌握计算机应用技术的跨世纪高级人才,已成为历史赋予我们高等学校的一项重任。

经过多年的教学实践与努力,高等学校的计算机基础课程大致可以归结为文化基础、技术基础与应用基础这三个层次。这不仅表现在高等学校课程体系的设置上,而且表现在教学内容上。这一点已成为高等学校的共识。高等学校应在注重基础、注重应用的原则下,适应新发展,不断更新计算机基础教育的教学内容。

为此,我们组织了上海、南京等地近 20 所大学、高等专科学校的几十位多年从事这方面教学、有丰富教学经验的教师,编写了这套高等学校计算机基础教育系列教材,以适应高等学校在新形势下的需要,为高等学校开展计算机基础教育提供教材的选择。

这套系列教材共分两辑,第一辑适用于高等学校非计算机专业少学时本科生与大专生,它包括了《计算机文化基础》、《程序设计与软件开发基础》、《FOX PRO 数据库及其应用》、《微型计算机原理》。第二辑适用于高等学校非计算机专业多学时本科生,它包括了《计算机文化基础教程》、《计算机通信网》、《数据库应用基础》、《计算机办公事务处理基础》、《计算机绘图与辅助设计》、《微型计算机原理》、《单片机原理与应用》、《实用 Internet 教程》等。

这套系列教材既可作高等学校计算机基础教育、成人高等教育与继续教育的教材,也可供专业技术人员、管理人员与有关读者参考。

本套教材将先后陆续出版,因时间仓促,其中若有疏漏及不妥之处,恳请读者提出批评与宝贵意见,以便修订时更正。

盛 焕 烨

1996 年 5 月

高等学校计算机基础教育系列教材编纂委员会

编纂委员会顾问：张仲俊

编纂委员会主任：盛焕烨

编纂委员会副主任：侯文永 杭必政 钱培德 东鲁红 史九标 黄国建

编纂委员会委员（以姓氏笔画为序）：

王 蕾 东鲁红 史九林 冯矢勇 孙一平 孙平川 孙德文 张小明

张 立 张汝杰 张仲俊 刘伯生 祁惠民 乔沛荣 朱秀珍 汤宝骥

汤文彬 杨 健 吴忿勤 杭必政 周治仁 范荷芙 侯文永 姚天肪

施小英 俞丽和 俞德礼 钱培德 钱焕延 盛焕烨 章 鲁 黄国建

程自强 瞿彭志

编纂委员会秘书：朱秀珍 周少明 张国华

目 录

第 1 章 程序设计与 C 语言概况	1
1.1 程序设计的基本概念	1
1.2 程序设计语言的发展	2
1.3 C 语言概述	3
1.4 C 程序的基本结构和格式特点	5
1.5 C 程序的编辑、编译和运行	9
习题 1	13
第 2 章 基本数据类型、操作符和表达式	14
2.1 变量和基本数据类型	14
2.2 操作符和表达式	17
2.3 类型的转换	26
习题 2	26
第 3 章 基本的数据输入、输出函数的应用	28
3.1 非格式化输出函数	28
3.2 非格式化输入函数	30
3.3 格式化输出函数 printf()	31
3.4 格式化输入函数 scanf()	33
习题 3	34
第 4 章 语句	36
4.1 语句和语句的分类	36
4.2 空语句	36
4.3 块语句	37
4.4 表达式语句	37
4.5 表达式值的真假	38
4.6 if 语句	38
4.7 switch 语句	41
4.8 while 语句和 do...while 语句	44
4.9 for 语句	47
4.10 转移语句	51

习题 4	55
第 5 章 数组	57
5.1 数组的声明、定义和引用	57
5.2 数组的初始化	60
5.3 字符数组和字符串	61
5.4 数组的越界	64
习题 5	66
第 6 章 函数和存储类别	68
6.1 函数使用三步曲——声明、定义和调用	68
6.2 函数的定义	70
6.3 函数的调用	73
6.4 数组作为函数调用参数	74
6.5 存储类别	75
6.6 多文件程序	79
习题 6	80
第 7 章 指针	82
7.1 指针与地址的概念	82
7.2 指针操作	84
7.3 指针和数组	86
7.4 动态分配内存	93
7.5 字符串与指针	94
7.6 指向函数的指针	102
7.7 带参数的 main() 函数	105
7.8 指向指针的指针	106
习题 7	106
第 8 章 结构体、共用体与枚举	109
8.1 结构体的定义	109
8.2 结构体变量的使用	111
8.3 结构体变量的初始化	112
8.4 结构体数组	114
8.5 指向结构体的指针	116
8.6 结构体作为函数的参数	119
8.7 动态结构体：链表	122
8.8 共用体	131
8.9 枚举类型	133

8.10 类型定义	134
习题 8	136
第 9 章 编译预处理	138
9.1 宏定义	138
9.2 文件蕴含	144
9.3 条件编译	148
习题 9	151
第 10 章 文件	153
10.1 文件的概念及其两种组织形式	153
10.2 FILE 结构与标准文件	154
10.3 文件的打开和关闭	155
10.4 读写文件	156
10.5 文件的定位和测试	169
习题 10	171
第 11 章 结构化程序设计方法	173
11.1 问题的描述与求解	173
11.2 三种基本结构	176
11.3 程序的动态结构与静态结构	178
11.4 模块间的层次与关系	179
第 12 章 软件工程概述	181
12.1 软件工程的诞生及其发展	181
12.2 软件工程的基本概念	182
12.3 软件开发技术	184
12.4 结构化设计方法	191
12.5 软件测试方法	195
习题 12	198
附录 A ASCII 字符代码表	199
附录 B 操作符及其主要功能	200
附录 C 常用库函数	201
参考文献	204

第 1 章 程序设计与 C 语言概况

本章在阐述程序设计的基本概念之后, 简单介绍一下程序设计语言的发展历史, 然后引入 C 语言, 用简明的例子程序加以说明, 接着阐明 C 语言的基本元素的构成, 最后讨论 C 程序的编辑、编译、链接和运行的过程, 并简介集成开发环境的使用。

1.1 程序设计的基本概念

计算机科学技术的飞速发展, 使计算机更加“神通广大”。它能帮助人从事许多工作, 且能做到既正确效率又高。人要想让计算机工作, 必须告诉它怎么做。人与人之间的交流靠的是人类语言, 那么人与计算机之间的信息交流靠什么呢? 这种交流的最通用的手段就是程序设计语言, 简称程序语言。

当人们想利用计算机解决某个问题时, 必须先用程序设计语言安排好处理步骤并存入计算机内, 然后计算机按既定的处理步骤不折不扣地完成人们所规定的工作。70 年代人们把用程序设计语言描述的处理步骤称为计算机程序(简称程序)。那时一个计算机程序主要包括两方面的内容: 一是关于程序实现算法的操作步骤描述, 即动作描述; 二是关于算法对象的描述, 即数据描述。早期, 著名计算机科学家 N.Wirth 所提出的程序定义是:

程序=数据结构+算法

现在我们把用程序设计语言编制一个能完成某项任务的计算机程序的过程叫做程序设计。这样定义比较容易理解, 事实上, 程序设计牵涉的面较广, 它涉及到算法、数据结构、方法和工具。可用下面公式来概括: 程序设计=算法+数据结构+方法+工具。这四个方面是互相依存、不可分割的。程序设计的概念一直在演变在发展, 但上述概念适用于我们对 C 语言的学习。

算法与数据结构是人们用计算机解题时所作的两种抽象。算法是从计算机操作的角度对解题过程的抽象, 数据结构是从如何组织被处理对象的角度进行抽象。这两种抽象互相依赖、互相补充, 以有效地降低问题的复杂性。程序设计主要考虑和设计合适的数据结构和有效的算法, 但不可忽视, 程序设计的方法同样是保证程序设计质量的一个非常重要的因素。现在普遍采用的是结构化程序设计技术, 它的主要方法是“自顶向下, 逐步求精”。自顶向下, 逐步细化的方法符合人们解决复杂问题的普遍规律, 可以显著提高程序设计的效率。同时, 用先全局后局部、先整体后细节、先抽象后具体地逐步细化过程可以设计出具有结构层次清晰的、可读性强的程序。此外选择使用各种优质的软件工具, 如算法描述工具、编辑程序、工具软件和调试工具等, 也是提高程序设计效率和质量的一条重要途径。

要掌握程序设计技术, 必须首先学会一门程序设计语言, 因为程序设计语言是人与计算机进行交流的工具。世界上人类的语言有几百种, 计算机程序设计语言同样种类繁多, 发

展迅速。下节介绍程序设计语言的发展与分类。

1.2 程序设计语言的发展

程序设计语言通常被分成三类，也可以说分成三级，从低级到高级依次为：机器语言，汇编语言和高级语言。

1.2.1 机器语言

大家知道由硬件组成的数字计算机只能接收由“0”和“1”组成的二进制信息。要计算机执行一定的操作，就要编写一系列的二进制代码。这种不需翻译即由计算机所直接执行的指令称为机器指令。这些指令的集合叫机器语言。用机器语言编辑的程序是由一条条机器指令组成的。这是一种面向机器的语言，具有算法刻画细致、执行速度快、目标程序效率高、内存占用少以及可直接使用计算机的全部资源等优点。但它也具有相当多的无法忍受的缺点，例如，由于不同的机器有不同的机器指令系统，一个程序不能同时在两台不同型号的计算机上运行，缺乏通用性；程序可读性极差，犹如在读“天书”。

由于这种程序直观性差，与人们习惯使用的自然语言和数学表达式差别太大，故而机器语言难学、难记，程序难以编写和修改。可见机器语言的程序设计严重阻碍了计算机的推广应用。

1.2.2 汇编语言

为了克服机器语言存在的缺点，人们努力创造一种新的语言，这就是符号语言，也称汇编语言。如用“ADD”与“SUB”分别取代机器语言中的加法和减法。用汇编语言编写的程序机器不能直接识别，需用汇编程序把它翻译成机器代码。汇编语言比机器语言好学、好记、程序好写，使用起来灵活、方便，但它的通用性仍然很差。没有完全解决机器语言所存在的问题。但在某些特定的场合，用汇编语言处理的效率很高。因此 C/C++ 允许在自己的代码中嵌入汇编语言。

1.2.3 高级语言

以前把能直接表示数学公式和解题方法的语言，称为高级程序设计语言，简称高级语言。这种语言直观通俗，非常接近人们的“自然描述”语言，便于编写、阅读、修改和维护。它又不同于具体机器的结构，故通用性强，便于软件的移植。用高级语言编写的源程序，机器是不能直接识别和接受的，需通过编译程序或解释程序进行翻译，最终生成机器语言程序。

世界上第一个高级程序设计语言是 FORTRAN 语言，它是在 50 年代中期由 John Backus 领导的 IBM 的一个小组研制而成的。1972 年确立了国际标准(ISO)FORTRAN 语言。自从 FORTRAN 语言问世之后，新的程序设计语言如雨后春笋般地不断涌现，在这短短的 40 年间，已经诞生了成百上千种程序设计语言。例如：ALGOL 60, COBOL, BASIC, Pascal, LISP,

PL/I, Ada, Prolog, C, C++, Smalltalk 等。这些语言都有各自的特点, 在应用领域方面也有所侧重。其中 FORTRAN, ALGOL 60, BASIC 语言具有很强的数值计算功能, 主要用于科学计算。COBOL 语言主要应用于商业事务的数据处理。Pascal 语言是典型的结构化程序设计语言, 它继承和发展了 ALGOL 60 的特点, 不仅形成了程序设计理论研究的基础与对象, 而且对后来的语言设计发生很大影响。如: Ada 就是在 Pascal 的基础上研制而成的, 它强调可靠性、效率与可维护性。LISP 语言适用于非数值计算, 在早期较多地应用于人工智能领域。PL/I 语言汇集了 ALGOL 60, FORTRAN 与 COBOL 的特点, 任用户选用, 适用于科学计算和数据处理, 称为“无所不包”的大型通用语言。Prolog 语言是一种基于一阶谓词演算的逻辑型语言, 广泛应用于人工智能、数据库领域中。C++和 SmallTalk 是面向对象的程序设计语言。在这里值得一提的是: C 语言是一种通用的高级程序设计语言, 它与其他高级语言相比, 具有不可替代的优点。它向用户提供了在低级水平上使用计算机的能力, 即像汇编语言一样, 在 C 程序中可直接与机器硬件接口打交道, 因此有人称 C 语言为“结构化语言”、“高级汇编语言”, 它不仅适合开发各类应用软件而且适合开发系统软件。鉴于上述特点, 在学习高级程序设计语言课程的安排中, 我们选择了 C 语言。

1.3 C 语言概述

1.3.1 C 语言的起源与发展

C 语言是一种被广泛应用的计算机高级程序设计语言。它经历了四大发展阶段。特别要了解后三个发展阶段, 因为它们与我们用 C 语言设计程序和开发密切相关。

1. 第一阶段: C 语言的形成阶段

C 语言是在 B 语言的基础上发展而成的。1963 年, 英国剑桥大学在 ALGOL60 程序设计语言基础上, 推出了 CPL(Combined Programming Language)。因其规模大、实现难等原因, 1967 年 Martin Richards 对 CPL 作了简化, 推出了 BCPL(B 代表 basic)。1970 年, 美国贝尔实验室的 Ken Thompson 又在 BCPL 的基础上, 再次作进一步的简化, 设计出简单而又接近硬件的 B 语言, 并用 B 语言编制了 UNIX 操作系统和大量的实用程序。因 B 语言只有单一的字类型、过于简单等原因而未能流行。1971 年 D.M. Ritchie 在 B 语言基础上设计了 C 语言, 于 1972 年投入使用。1972 年由美国 Bell 实验室为描述 UNIX 操作系统和 C 编译程序而开发了 C 系统描述语言。1973 年 K.Thompson 和 D.M. Ritchie 把 UNIX 系统用 C 重写一遍, 增加了多道程序设计功能, C 语言的编译程序也都建立在 C 语言的基础上。随着 UNIX 的巨大成功和被广泛移植到各种机器上, C 语言也被人们所接受, 并移植到大型机、小型机和微机上。

2. 第二阶段: C 语言的成熟阶段

1978 年, B.M.Kernighan 和 D.M.Ritchie 合写了书, 一本“名著”(见参考文献[1]), 奠定了 C 的基础。称为 K&R C。1983 年, 美国国家标准化协会(ANSI)下的一个 X3J11 小组制定了 C 的标准, 称为 ANSI C。1987 年又公布了更新的标准, 称为 87 ANSI 标准 C。现在的标准为 ANSI 标准 X3.159-1989。国际标准化组织(ISO)也采取了相应的行动: 组织了

一个技术委员会 JTC1/SC22/WG14 讨论了 ANSI X3J11 小组的工作，制定了 ISO 的标准。最近的 C 语言标准是 ISO 9899:1990，其实质是与 X3.159-1989 相同的，仅仅是章节编号有所不同。

任何标准的修改是一件不容易的事，同样 C 语言标准的更新必然有深刻的原因，这种更新或者为程序设计带来更多的方便性、或者提高安全性、或者使编译系统多做工作，以便早日发现错误。但为了保持版本的兼容性，新的版本往往在更新方面不作强制规定，而作“兼容”处理，“允许”老版本的一些规定继续有效。本书全部内容和例子程序采用最新的版本。这决不是形式上“追求时髦”，而是为了提高程序设计的效率和树立良好的设计风格。请读者在阅读或参考各种书刊时，注意版本上的差异，应吸收最新版本的精华。

3. 第三阶段：C 语言的全面推广应用阶段

80 年代后半期好几个软件公司相继推出(并不时更新版本)微机上的 C 语言开发系统。例如 Microsoft 公司的早期的 Quick C, Microsoft C 4.0, 5.0, 6.0, 到 C/C++ 7.0；Borland 公司的 Turbo C 1.0, 1.5 和 2.0；Zorthec 公司的 Zorthec C；Lattice 公司也推出了几个版本的 Lattice C 等。它们对 C 的推广应用起了很大的作用。比较起来 Turbo C 2.0 在国内应用最广；其原因在于该集成开发环境不但对学习者，而且对应用者都是比较理想的。所以本书以 Turbo C 2.0 作为使用工具。

4. 第四阶段：C 语言和 C++ 语言的交溶发展阶段

C++ 的基础是 C，C++ 比 C 要复杂得多，但 C++ 的开发系统可以编译和运行 C 程序。ANSI 和 ISO 除了为 C 制定标准外，还为 C++ 制定了标准。在 C++ 的标准中，对于“C 的部分”进行了许多扩展和改动；这些扩展和改动都对程序的开发很有利。一些大的软件公司，到了 90 年代就不再更新 C 的版本了，而是不断地更新 C++ 的版本，其中最令人赞赏的是 Borland 和 Microsoft 公司：Borland 接连推出 Turbo C++ 1.0, Borland C++ 2.0, 3.0, 3.1, 4.0 和 4.5。Microsoft 在推出 C/C++ 7.0 后，接连推出 Visual C++ 1.0, 1.5 和 2.0。整个软件已庞大到 100 兆字节。实际上开发 C 程序在 C++ 系统下比在 C 的系统下更方便、更优越，希读者在以后有条件时实现这种转换。

1.3.2 C 语言的特点

C 语言具有高级语言的特点，又具有汇编语言的特点，既是成功的系统语言，又是一个实用的程序设计语言，能用它方便地编写不依赖于计算机硬件设施的应用程序，又能用它编写各种系统程序。C 语言具有多方面的特点，其主要特点如下：

1. 语言简洁、结构清晰、使用方便

C 语言编写的程序通常是由若干个函数组成的。C 语言强大的函数功能为程序的模块化、结构化提供了手段，因而使得源程序显得简洁清晰、代码行少、可读性强。另外，C 语言程序在运行时所需要的支持少，占用的存储空间也小。

2. 语言表达能力强

C 语言包含丰富的操作符以及可进行低级处理的位操作、地址操作，足以取代汇编语言编写各种系统程序和应用程序中相应的操作。此外，还提供先进的程序控制结构和强有力的数据类型构造能力。

3. 高效率

C 是一种高效率的编译型语言，由于 C 可以进行接近汇编的精细描述，因此，其生成的目标程序的质量较高，即可以开发出执行速度很快的程序。在任何系统上 C 语言的编译速度都很快。对于较大的程序，C 源程序可分开存放在若干个文件中，待分别编译后再连接在一起，形成一个可执行文件。

4. 通用性、可移植性强

几乎在各类通用机的各种操作系统上都配有 C 语言编译系统，且大部分是由 C 语言编译移植得到的。C 源程序稍加修改即可从一个环境搬到另一个环境上运行。

5. 丰富的函数库

函数库不是 C 语言本身的要求。但软件公司都配置了几乎相同的函数库。它为方便用户提供了丰富的函数，函数库中包括图形功能、文件处理、数据库支持、屏幕窗口控制、数据访问、通信控制和通用函数。这些库函数大大地节省了开发时间。因为本书的程序以 Turbo C 2.0 为基础，所以，如果 Microsoft C 的个别库函数与此有所不同时，将予以说明。

C 语言有诸多优点，但存在缺陷。在前言中提到 C 是属于过程式语言，所有过程式语言的“函数”或“过程”的机制都不能适应高速度、高质量开发大型软件的需要，不能满足人们对软件重用性的要求，不能适应变化的用户环境等等。此外 C 也有不少“小的”缺陷：例如，它不是强类型语言，如对变量的类型约束不够严格，这会影响程序的安全性；缺少函数重载机制，用户不得不创建大量的函数名，使整个软件管理复杂化；缺少对类型的 const 修饰机制，不得不使用#define，又导致不能约束类型。又因为 C 是“程序员责任制”，对指针在运行期不进行检查，这可能导致系统出现很多奇怪的和偶发的错误。另外，C 语言的操作符优先级太多，难于记忆等。上述大多数问题在 C++中得到了相当理想的解决。

1.4 C 程序的基本结构和格式特点

本节从简单的 C 程序例子出发介绍 C 程序的基本组成结构，让读者对 C 语言有一个初步的印象。例子中提及的概念、名称在以后各章中都将有详细介绍。

1.4.1 简单的 C 例子程序

例子程序 CH01-01.C，其功能是从键盘上输入任一整数 n，计算 n 的立方，并在 CRT 上显示输出结果。下面是程序清单：

```
1  /* CH01-01.C  The example is print the cube of n */
2  #include <stdio.h>
3  main( )
4  {
5      int n, s;
6      printf("n=");
7      scanf("%d",&n);
8      s=n*n*n;
9      printf("s=%d\n", s);
10     return 0;
```

```
11 }
```

上面程序运行的结果如下：

```
n=3<CR>
s=27
```

说明：C 语言源程序中并无行号，上例中的 C 程序中给出行号只为说明程序方便。

第 1 行是注释行(或称为注解)。C 语言的注释是由 “/*” 和 “/” 括起来的一系列字符，注释可出现在语句之后，也可以出现语句之前。注释可连续占几行。注释不影响程序的运行，因为它只用来对程序进行说明。

第 2 行是预处理命令。C 程序中若调用了 C 的某个库函数，就必须用(#include)预处理命令，将包含该函数说明的“头文件”嵌入到源程序文件中。本例中调用了包含在 Stdio.h 头文件中说明的输入输出函数 scanf, printf, 所以本行命令进行了嵌入说明。

第 3 ~ 11 行是名为 main() 的无参数函数。main() 是一个特殊的函数，称为主函数。它是任何一个完整的 C 程序中必不可少的，它是能直接运行的函数。在 C 语言中，程序编译、运行的基本单位是函数。

第 4 行的花括号 “{” 和第 11 行的花括号 “}” 将函数的语句部分即函数体括起来，这是 C 函数要求的形式(组成一个“块”)。

第 5 行为声明语句，说明 n,s 为整型变量，在 C 语言中，对所使用的变量必须进行类型声明。“;” 为语句终止符，C 语言中，语句一般以 “;” 号为结束标志。

第 6 行 printf() 为格式输出函数，本句的作用是显示字符串 “n=”。

第 7 行 scanf() 为格式输入函数，本句的作用是从键盘上为整型变量 n 输入一个值。

第 8 行是计算变量 n 的立方值，然后赋给变量 s。

第 9 行也是一个格式输出函数。其作用是在显示器上先显示 “s=”，再显示整型变量 s 的值。程序中出现的 %d 是一种格式命令，它告诉系统在该位置上变量 s 要按十进制整数格式输出。“\n” 表示输出要换一个新行。

第 10 行是一个返回语句。

C 程序在磁盘上以文本方式存储，主文件名由用户在存储时指定，只要符合 DOS 的要求；文件的扩展名必须为 “.C”。

1.4.2 C 程序的基本结构

综上所述，C 程序的基本结构为：

```
预处理命令      /* 嵌入命令、宏定义等 */
main()           /* 主函数名 */
{                /* 函数体起始符*/
    若干语句      /* 函数体*/
}                /* 函数体终止符*
```

总之，C 语言编写的程序通常是由包括 main() 在内的一组函数组成。每个函数都有一个函数名，而函数本身是由花括号引起来的一系列语句构成的，每个语句必须以分号结尾。一个程序总是从主函数 main() 开始执行，通过主函数，可以直接或间接调用其他函数。在