

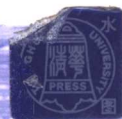
普通高校本科计算机专业 **特色** 教材精选

C 语言程序设计

马靖善 秦玉平 主 编
冯佳昕 刘福德 副主编
王秀坤 主 审

<http://www.tup.com.cn>

清华大学出版社



普通高校本科计算机专业 **特色** 教材精选

C 语言程序设计

马靖善 秦玉平 主 编
冯佳昕 刘福德 副主编
王秀坤 主 审

清华大学出版社
北 京

内 容 简 介

本书共 8 章。前 6 章介绍了 C 语言的基本语法、各种数据类型、程序设计方法及文件操作,后两章介绍了 C 语言的图形处理以及 C 语言与 DOS 和 BIOS 的接口与调用。

书中对常规的 C 语言教材的体系结构作了适当调整,将指针部分分散到各个相应的章节中分别进行介绍,使读者很容易理解指针的概念,很自然地掌握各种指针的用法。本书内容循序渐进,结构清晰,层次分明,通俗易懂,通过大量例题验证语法和说明程序设计方法。为了使读者更好地掌握各章节内容,章末配有大量精选的习题。

本书可作为高等学校相关专业教材,也可作为自学者或各种计算机培训班的教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C 语言程序设计/马靖善,秦玉平主编. —北京:清华大学出版社,2005. 11

(普通高校本科计算机专业特色教材精选)

ISBN 7-302-11597-4

I. C… II. ①马… ②秦… 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(2005)第 092544 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:焦 虹

文稿编辑:王冰飞

印 装 者:清华大学印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:20 字数:458 千字

版 次:2005 年 11 月第 1 版 2005 年 11 月第 1 次印刷

书 号:ISBN 7-302-11597-4/TP·7575

印 数:1~5000

定 价:25.00 元

编审委员会

主任：蒋宗礼

副主任：李仲麟 何炎祥

委员：(排名不分先后)

王向东 宁 洪 朱庆生 吴功宜 吴 跃

张 虹 张 钢 张为群 余雪丽 陈志国

武 波 孟祥旭 孟小峰 胡金初 姚放吾

原福永 黄刘生 廖明宏 薛永生

秘书长：王听讲

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地说，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。



2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系，本套教材力求突出学科的理论与实践紧密结合的特征，结合实例讲解理论，使理论来源于实践，又进一步指导实践。学生通过实践深化对理论的理解，更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性，并做到通俗易懂，易教易学，使学生不仅知其然，知其所以然，还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容，学生通过学习和练习，能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力，也注重培养学生解决问题的能力，以适应新经济时代对人才的需要，满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示，学生上机实验指导等辅助教学资源，有些教材还提供能用于网上下载的文件，以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同，所以对特色教学的理解也不尽一致，我们恳切希望大家在使用本套教材的过程中，及时地给我们提出批评和改进意见，以便我们做好教材的修订改版工作，使其日趋完善。

我们相信经过大家的共同努力，这套教材一定能成为特色鲜明、质量上乘的优秀教材，同时，我们也希望通过本套教材的编写出版，为“高等学校教学质量和教学改革工程”作出贡献。

清华大学出版社

前言

PREFACE

C 语言是一种结构化程序设计语言，它兼有高级语言的特点和低级语言的功能。其程序设计功能强大，既可用于编写应用软件，又可用于设计制作系统软件。C 语言是一门较为通用的编程语言，目前越来越得到广泛的应用。

本书分为 8 章。前 6 章详细地介绍了 C 语言的基本语法规则和功能实现方法。其中，第 1 章为 C 语言基础知识，介绍了 C 语言的发展、C 语言的特点、C 语言的 32 个关键字和语句的分类以及 Turbo C 上机操作指导；第 2 章为 C 语言的基本语法规则，介绍了数据类型、运算符和表达式、基本输入输出函数、分支和循环程序设计语句；第 3 章为数组，介绍了一维数组与二维数组的定义和使用，以及常用字符串处理函数；第 4 章为函数，介绍了函数的定义、调用、命令行参数的使用，以及全局变量与局部变量等；第 5 章为结构体、共用体和枚举，介绍了结构体、共用体和枚举类型的定义与使用、链表的基本操作等；第 6 章为文件系统，介绍了缓冲区系统文件的常用处理方法。后两章主要讲述 C 语言图形处理和 DOS、BIOS 功能调用。其中，第 7 章为 C 语言绘图，介绍了 C 语言的图形处理功能；第 8 章为 C 程序与 DOS、BIOS 接口，介绍了常用的 DOS、BIOS 功能和 C 语言中常用的调用 DOS、BIOS 功能的函数。

本书是根据编者多年的教学经验编写的，书中对常规的 C 语言教材的体系结构作了适当调整，主要是将指针部分分散到各个相应的章节中分别进行介绍，这样读者很容易理解指针的概念，很自然地掌握各种指针的用法。本书内容循序渐进，结构清晰，层次分明，通俗易懂，通过大量例题验证语法和说明程序设计方法，并且，为了使读者更好地掌握各章节内容，章末配有大量精选的习题。通过本书的学习和解题练习，读者既能迅速掌握 C 语言的基础知识，又能很快地学会 C 语言的编程技术，得到解决实际问题的能力。

本书的第 1 章、第 6 章由冯佳昕编写，第 2 章、第 3 章由秦玉平编写，第 4 章、第 5 章由马靖善编写，第 8 章和附录由刘福德编写。各章

末习题由秦玉平编写。全书由大连理工大学王秀坤教授主审。

在本书编写过程中，编者参考了大量有关 C 语言程序设计的书籍和资料，在此对这些参考文献的作者表示感谢。

由于编者水平有限，书中难免存在错误和不当之处，恳请广大读者批评指正，以便再版时改进。

编 者
2005 年 6 月

目 录

CONTENTS

第 1 章 C 语言概述	1
1.1 C 语言功能简介	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	2
1.1.3 C 语言中 32 个关键字和语句形式	3
1.1.4 程序的三种基本结构与流程图简介	4
1.2 Turbo C 上机操作指导	7
1.2.1 Turbo C 用户窗口界面	7
1.2.2 Turbo C 的主要菜单功能	8
1.2.3 获取帮助与编辑操作功能	11
1.3 简单的 C 程序介绍	13
习题	16
第 2 章 基本语法规则	17
2.1 常量、变量和指针	17
2.1.1 标识符	17
2.1.2 常量和变量	18
2.1.3 整型数据	20
2.1.4 实型数据	21
2.1.5 字符型数据	22
2.1.6 字符串常量	23
2.1.7 变量及指针	24
2.2 运算符与表达式	27
2.2.1 算术运算符和算术表达式	29
2.2.2 关系运算符和关系表达式	33
2.2.3 逻辑运算符和逻辑表达式	34
2.2.4 位运算符	36

2.2.5	赋值运算符和赋值表达式	38
2.2.6	逗号运算符和逗号表达式	40
2.2.7	条件运算符和条件表达式	41
2.2.8	求字节数运算符	42
2.2.9	指针运算符	43
2.3	基本输入输出函数	43
2.3.1	字符输出函数——putchar()	44
2.3.2	字符输入函数——getchar()	44
2.3.3	格式输出函数——printf()	46
2.3.4	格式输入函数——scanf()	48
2.4	分支语句	50
2.4.1	if 语句	50
2.4.2	switch...case 语句	53
2.5	循环语句	55
2.5.1	while 语句	55
2.5.2	do...while 语句	56
2.5.3	for 语句	57
2.5.4	循环语句的嵌套	60
2.5.5	break 语句和 continue 语句	60
2.5.6	goto 语句	62
2.6	编译预处理	62
2.6.1	宏定义	62
2.6.2	文件包含	65
2.6.3	条件编译	66
2.7	程序设计举例	68
习题		71
第 3 章 数组		79
3.1	一维数组及指针	79
3.1.1	一维数组的定义和初始化	79
3.1.2	一维数组元素的下标法引用	80
3.1.3	一维数组的指针	81
3.1.4	一维数组元素的指针法引用	83
3.2	一维字符数组及指针	85
3.2.1	一维字符数组的定义和初始化	85
3.2.2	字符数组的输入和输出	87
3.2.3	用字符数组实现字符串	89
3.2.4	用字符指针变量实现字符串	91

3.2.5 常用字符串处理函数	92
3.3 多维数组及指针	95
3.3.1 二维数组的定义和初始化	95
3.3.2 二维数组元素的下标法引用	97
3.3.3 二维数组的指针	97
3.3.4 二维数组元素的指针法引用	98
3.4 指针数组	101
3.5 二级指针	102
3.6 程序设计举例	103
习题	107
第4章 函数	117
4.1 函数概述	117
4.2 主函数与命令行参数	119
4.3 辅助函数与参数传递	120
4.3.1 函数的定义	120
4.3.2 函数调用与参数传递	121
4.3.3 函数的引用说明	122
4.3.4 函数调用中指针与参数的关系	122
4.3.5 函数与带参数的宏的区别	125
4.4 函数的嵌套调用与递归调用	128
4.4.1 函数的嵌套调用	128
4.4.2 函数的递归调用	129
4.5 函数指针与返回指针的函数	130
4.5.1 函数指针	130
4.5.2 函数指针作参数	131
4.5.3 返回指针的函数	131
4.6 变量的作用域与存储类别	132
4.7 程序设计举例	136
习题	144
第5章 结构体、共用体和枚举	153
5.1 结构体类型与变量	153
5.1.1 结构体类型定义	153
5.1.2 结构体变量的定义和引用	154
5.2 结构体数组	156
5.3 结构体类型数据的指针	158
5.3.1 结构体变量的指针与指针变量	158

5.3.2	结构体数组的指针与指针变量	159
5.3.3	结构体指针与变量作函数参数	160
5.3.4	自定义类型	161
5.4	链表的基本知识	162
5.4.1	动态分配和释放空间的函数	163
5.4.2	建立和输出链表	164
5.4.3	链表的基本操作	165
5.5	结构体位段	172
5.6	共用体类型与变量	174
5.6.1	共用体类型定义	175
5.6.2	共用体变量的定义和引用	175
5.7	枚举类型与变量	177
5.7.1	枚举类型定义	177
5.7.2	枚举变量与枚举元素	178
5.8	程序设计举例	181
习题		186
第 6 章	文件系统	193
6.1	文件系统概述与文件类型	193
6.2	文件的打开与关闭	194
6.2.1	文件的打开	194
6.2.2	文件的关闭	196
6.3	文件的读写操作	196
6.3.1	读写一个字符的函数	196
6.3.2	读写一个字符串函数	198
6.3.3	读写一个数据字块函数	200
6.3.4	文件的格式化读写函数	202
6.4	文件定位与随机读写	203
6.5	程序设计举例	206
习题		215
第 7 章	C 语言绘图	221
7.1	PC 机的显示系统	221
7.2	Turbo C 基本绘图函数	224
7.2.1	图形控制函数	224
7.2.2	颜色控制函数	225
7.2.3	基本绘图函数	226
7.2.4	填充	230

7.2.5 获取当前信息函数	233
7.3 图形方式下的文本输出	235
7.3.1 文本输出函数	235
7.3.2 定义文本字型	236
7.3.3 文本输出字符串	238
7.4 动画设计技术	238
7.4.1 存图取图	239
7.4.2 页操作	240
7.4.3 图视口操作	241
7.4.4 图形运行控制和声音函数	242
7.5 文本窗口	243
7.5.1 文本方式的控制	243
7.5.2 屏幕信息与存取	247
7.5.3 菜单设计	249
7.6 创建独立的图形运行程序	253
7.7 程序设计举例	254
第8章 C程序与DOS、BIOS接口	269
8.1 常用DOS功能调用说明	269
8.2 常用BIOS功能调用说明	275
8.3 C程序调用DOS、BIOS功能	279
8.3.1 C程序中寄存器的使用格式	279
8.3.2 调用DOS、BIOS功能的系统函数	280
8.4 程序设计举例	282
附录A 常用字符与ASCII代码对照表	287
附录B C库函数	289
参考文献	303

第 1 章

CHAPTER

C 语言概述

1.1 C 语言功能简介

C 语言是一种已经得到普遍应用、功能强大的计算机高级语言。虽然它是面向 DOS 编程的,但它是计算机的底层编程工具之一。由 C 语言编写的程序,其代码质量和运行速度仅次于用汇编语言编写的程序。汇编语言是编写底层代码的最好编程环境,但它的语句不易记忆,并且涉及很多的计算机硬件知识,非专业人员很难掌握。又由于汇编语言是面向机器的,在不同类的机器上,汇编代码有很大的不同,可移植性较差。高级语言是面向读者的,其语句都是用英文或其缩写来标识,学起来较为简单,并且所有的高级语言可移植性都非常好。然而一般的高级语言生成的代码质量都不高,运行速度较汇编语言要慢得多。而 C 语言作为高级语言的一种,它具有高级语言简单易学和可移植性好的特点,同时它又具有汇编语言生成代码质量高的优点。所以,C 语言具有较强的生命力和广泛的应用前景,常用它来编写一些系统软件和应用软件。

1.1.1 C 语言的发展

C 语言是在早期的 B 语言基础上发展起来的。早期的计算机都是用机器语言和汇编语言来编写程序代码,到了第二代计算机才有了高级语言。1960 年出现的 ALGOL-60 对其后的高级语言的发展起到了很好的推进作用,但是,它是一种面向问题的语言,过于抽象,难以描述系统,因此没有得到真正的推广。1963 年由英国剑桥大学推出了 CPL (Combind Programming Language) 语言,它比 ALGOL-60 更接近于硬件,但其规模较大,难以实现和学习。1967 年英国剑桥大学的 Martin Richards 对 CPL 作了简化,推出了 BCPL (Basic Combind Programming Language) 语言。1970 年由贝尔实验室的 Ken Thompson 所写的 B 语言又是对 BCPL 的进一步简化,且更接近于硬件,取 BCPL 当中第一个字母来命名,并且用 B 语言编写了 UNIX 操作系统。由于 B 语言过于简单、功能有限,所以没有流

行起来。1972 年至 1973 年,贝尔实验室的 D. M. Ritchie 在 B 语言基础上设计出了 C 语言,取 BCPL 中的第二个字母命名。

C 语言既保持了 BCPL 和 B 语言语法精练、接近于硬件的优点,又克服了它们过于简单、无数据类型等缺点。最初的 C 语言只是为描述和实现 UNIX 操作系统提供了一种工作语言而设计的。1973 年 K. Thompson 和 D. M. Ritchie 两人合作,把原来由两人用汇编语言编写的 UNIX 操作系统中 90% 以上的代码用 C 语言来重写,即 UNIX 5。后来 C 语言作了多次改进,不过它主要还是用在贝尔实验室内部。直到 1975 年用 C 语言写的 UNIX 6 公布后才引起业内人士的广泛关注。1978 年以后,C 语言已先后移植到大、中、小型计算机和微型计算机上,已独立于 UNIX 操作系统。

以 1978 年发表的 UNIX 7 中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的著作 *The C Programming Language*。该书中介绍的 C 语言成为后来被广泛使用的 C 语言的基础,称它为标准 C。以后相继出现了很多 C 语言版本,如 Microsoft C、Turbo C、Quick C、Borland C 等等。它们在语法上基本上是相同的,但在函数数量和功能上有较大的区别,用时要注意区分。1983 年美国国际标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。1988 年,K&R 按照 ANSI C 重写了 *The C programming Language*。1990 年,国际标准化组织 ISO(International Standard Organization)接受以 87 ANSI C 为 ISO C 的标准(ISO 9899—1990)。目前流行的 C 编译系统都是以它为基础的。

高级语言发展到现在,面向对象的程序设计方法越来越受到人们的青睐。比如:Visual FoxPro(VFP)、Visual Basic(VB)、Visual C++(VC++)、C++、Turbo C++、Borland C++、Java、J++、Power Builder(PB)等等。其中,功能比较强大的还是 C++ 语言,而这个面向对象的语言是以 C 语言为基础的。只有学好 C 语言,才能最终学习和掌握 C++。VC++、Borland C++、Turbo C++ 等都是 C++ 语言的编程环境。其中,VC++ 和 Borland C++ 的功能与编程环境深受广大程序设计人员的好评。

1.1.2 C 语言的特点

C 语言之所以能够在众多的高级语言竞争中脱颖而出,成为高级语言中的佼佼者,这主要是因为与普通高级语言相比,C 语言具有以下特点。

① 它的语言简洁、紧凑,使用方便、灵活。C 语言一共有 32 个关键字,9 种控制语句,程序书写形式自由,主要用小写字母表示。

② 数据类型丰富。C 语言数据类型包括整型、实型、字符型、枚举类型;结构体、共用体、数组和文件类型;指针类型;空类型。其中,整型、实型、字符型还有多种小的类型。用这些数据类型可以表示各种各样的数据结构(如链表、树、栈等)。指针类型是 C 语言中最具特点的一种数据类型,它使用起来非常灵活自如,把 C 语言的功能特点发挥得淋漓尽致,但是,由于指针在运用上非常灵活,所以,它也正是初学者最难以驾驭之处。

③ 运算符多样。C 语言中的运算符包含的范围非常广泛,一共有 44 个运算符。除包括算术运算符、关系运算符、逻辑运算符等常规运算符之外,还含有指针运算符、地址运算符、位运算符、自增自减运算符、条件运算符、复合的赋值运算符,甚至连圆括号、方括

号、逗号、小数点等都是运算符。由于C语言的运算类型极为丰富,表达式类型多样化,所以用C语言能够实现各种各样的高级和低级运算。

④ 函数是程序的主体。C语言中每一项功能的完成都是由函数来实现的,可以由C系统中已提供的功能函数来实现,也可以是用户自编函数来实现。函数是程序的基本单位,是C语言的灵魂。

⑤ 语法检查不太严格,程序书写自由度大。比如,数组下标不作超界检查;整型、字符型、逻辑型量可以通用;一个物理行可以写多个语句,一个语句也可以分在连续的多个的物理行上。对这些规则,初学者要认真掌握。

⑥ C语言允许直接访问物理地址。C语言中含有位运算和指针运算,能够直接对内存地址进行访问操作,可以实现汇编语言的大部分功能,即直接对硬件进行操作。所以,它既具有高级语言的功能,又兼有汇编语言(低级语言)的大部分功能。有时,也称它为“中间语言”或“中级语言”。可以说,它是高级语言中的低级语言。

⑦ 生成的目标代码质量高。它比一般的高级语言生成的目标代码质量高约20%,但还是要比汇编语言低10%~20%。这在高级语言中已是出类拔萃的了。

⑧ 可移植性好。同其他高级语言一样,C语言程序不作大的修改就可以很容易地移植到其他类型的机器上去。

目前较为流行的高级语言主要有: BASIC、Pascal、Fortran、C及面向对象的和可视化的编程语言。BASIC是初学者指令代码,Pascal是一种教学语言,Fortran是用于数值计算的高级语言。C语言能完成各种功能,是其他高级语言无法比拟的,所以,越来越多的人开始学习和使用C语言进行程序设计。

1.1.3 C语言中32个关键字和语句形式

1. 32个关键字

① 程序控制语句关键字(12个):

if else for do while continue
switch break case default goto return

② 类型定义说明关键字(12个):

int char float double long short
signed unsigned enum struct union void

③ 存储类别定义说明关键字(4个):

auto register static extern

④ 常量、变量定义和自定义类型关键字(3个):

const volatile typedef

⑤ 字节测试关键字(1个):

sizeof

2. 语句形式

① 控制语句:


```

if( ~ ) ~ else ~      /* 选择分支语句 */
switch( ~ ) case...;  /* 多分支语句 */
for( ~ ; ~ ; ~ ) ~    /* for 循环语句 */
while( ~ ) ~          /* while 循环语句 */
do ~ ; while( ~ );    /* do while 循环语句 */
continue;             /* 无条件进行下一次循环语句 */
break;               /* 无条件结束当前层循环或跳出 switch 语句 */
goto ~               /* 无条件跳转到程序指定处语句 */
return ~            /* 函数调用结束返回语句 */

```

② 函数调用语句:

函数名(实参表达式表);

③ 表达式语句:

表达式;

④ 空语句:

;或{} /* 不作任何操作 */

⑤ 复合语句(分程序):

{ ~ }

⑥ 注释语句:

/* ~ */ /* 注释的部分不参加程序编译和运行 */

其中“~”代表一个或多个表达式或语句,括号内为表达式,括号外为语句。

1.1.4 程序的三种基本结构与流程图简介

1966 年,Bohra 和 Jacopini 提出了程序的三种结构,即顺序结构、分支结构和循环结构。流程图是用一些图框来表示程序或算法的运行走向的一种图示。用图形表示算法或程序的走向,直观形象、容易理解。美国国家标准化协会(ANSI)规定了一些常用的流程图符号(如图 1.1 所示),已被程序工作者所普遍采用。

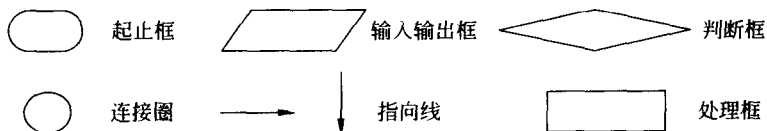


图 1.1 常用的流程图符号

1. 三种基本结构与其对应的流程图

(1) 顺序结构

程序由上至下每个语句顺序执行的结构称为顺序程序结构。与其对应的流程图见图 1.2。

(2) 分支结构