

大学计算机基础教育规划教材

C程序设计教程

黄维通 鲁明羽 编著

1+X



清华大学出版社

大学计算机基础教育规划教材

C程序设计教程

黄维通 鲁明羽 编著

1+X

清华大学出版社
北京

内 容 简 介

本教材从C语言程序设计的基本原理及程序设计的基本思想出发,以“基础—应用”为主线,紧扣基础,循序渐进,面向应用。为方便授课和学习,本教材有配套的电子课件和习题解析。

本书首先讲述程序设计中的基本概念与应用基础,如数据类型、变量、数组、控制结构及判断结构等;在掌握了这些基本概念与应用的基础上适时引入函数的结构与应用、指针的概念及其应用、算法设计与实现、结构型数据的应用及文件的操作等面向应用的知识点介绍。

本书可作为高等院校、水平考试、各类成人教育的教材使用,也可供计算机爱好者自学。

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 程序设计教程/黄维通,鲁明羽编著. —北京:清华大学出版社,2005.11

(大学计算机基础教育规划教材)

ISBN 7-302-11465-X

I. C… II. ①黄… ②鲁… 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084340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:张 龙

文稿编辑:徐跃进

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

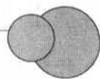
开 本:185×260 印张:14.5 字数:341千字

版 次:2005年11月第1版 2005年11月第1次印刷

书 号:ISBN 7-302-11465-X/TP·7522

印 数:1~5000

定 价:19.00元



进入 21 世纪,社会信息化不断向纵深发展,各行各业的信息化进程不断加速。我国的高等教育也进入了一个新的历史发展时期,尤其是高校的计算机基础教育,正在步入更加科学、更加合理、更加符合 21 世纪高校人才培养目标的新阶段。

为了进一步推动高校计算机基础教育的发展,教育部高等学校非计算机专业计算机基础课程教学指导分委员会近期提出了《关于进一步加强高校计算机基础教学的几点意见》(以下简称《意见》)。《意见》针对计算机基础教学的现状与发展,提出了计算机基础教学的指导思想;按照分类、分层次组织教学的思路,《意见》的附件提出了计算机基础课教学内容的知识结构与课程设置。《意见》认为,计算机基础教学的典型核心课程包括大学计算机基础、计算机程序设计基础、计算机硬件技术基础(微机原理与接口、单片机原理与应用)、数据库技术与应用、多媒体技术与应用、网络技术与应用。附件中介绍了上述六门核心课程的主要内容,这为今后的课程建设及教材编写提供了重要的依据。在下一步计算机课程规划工作中,建议各校采用“1+X”的方案,即“大学计算机基础”+若干必修/选修课程。

教材是实现教学要求的重要保证。为了更好地促进高校计算机基础教育的改革,我们组织了国内部分高校教师进行了深入的讨论和研究,根据《意见》中的相关课程教学基本要求组织编写了这套“大学计算机基础教育规划教材”。

本套教材的特点如下:

- (1) 体系完整,内容先进,符合大学非计算机专业学生的特点,注重应用,强调实践。
- (2) 教材的作者来自全国各个高校,都是教育部高等学校非计算机专业计算机基础课程教学指导委员会推荐的专家、教授和教学骨干。
- (3) 注重立体化教材的建设,除主教材外,还配有多媒体电子教案、习题与实验指导,以及教学网站和教学资源库等。
- (4) 注重案例教材和实验教材的建设,适应教师指导下的学生自主学习的教学模式。
- (5) 及时更新版本,力图反映计算机技术的新发展。



本套教材将随着高校计算机基础教育的发展不断调整,希望各位专家、教师和读者不吝提出宝贵的意见和建议,我们将根据大家的意见不断改进本套教材的组织、编写工作,为我国的计算机基础教育的教材建设和人才培养做出更大的贡献。

“大学计算机基础教育规划教材”丛书主编
教育部高等学校非计算机专业计算机基础课程教学指导分委员会主任委员

冯博琴

2004年8月



人类已经步入信息化的 21 世纪,信息时代的到来,使社会经济向知识经济发展。为了推进 21 世纪计算机基础教育改革、推进精品课程建设以及与之配套的精品教材建设,从素质教育的理念出发,结合信息化社会对高素质、复合型人才的需求,特出版此教材。此教材力求较为全面地介绍 C 语言的基本内容与程序设计思想。

C 语言是目前国内外广泛使用的程序设计语言之一,是国内外大学都开设的重要的基础课之一。C 语言功能丰富、表达能力强、使用方便灵活、程序执行效率高、可移植性好;既有高级语言的特点,又有汇编语言的特点,具有较强的系统处理能力。

本教材从 C 语言程序设计的基本原理及程序设计的基本思想出发,以“基础—应用”为主线,紧扣基础,面向应用,循序渐进地引导读者学习程序设计的思想和方法。每一章内容都力求做到基础与应用的有机结合,在例题的设计上也充分考虑这一问题。为方便授课和学习,本教材还配套有电子教案和习题解析。

本书的特点之一是通俗易懂,突出“三基”(基本概念、基本原理与基本应用)的介绍与应用。

本书的特点之二是在介绍“算法设计与实现”这部分内容时,重点介绍经典的排序与查找的算法及其实现,同时通过几种不同算法的比较,讨论算法的效率及代码实现的效率。计算机程序设计,很大一部分工作是为了实现某一特定的模型或算法,通过介绍算法设计并通过计算机程序加以实现,可以使读者更好地学习程序设计的思想、体系结构和方法,尤其是优化的程序设计方法。

本书的特点之三是基础与应用相结合,力求做到基础与应用相统一。

本书的特点之四是教材中有代码的详细注释,为了用最简洁的语言讲解代码结构及功能,本教材中的例题代码中给出了详细的代码注释,以利于读者更好地理解代码,同时,教材中的所有代码均在 TC3.0 下调试通过。

本书的特点之五是重点突出、难点分散,这样从学习者对知识的掌握角度来说是有利的,而且这样也有利于自学。

本书的特点之六是结合教学经验与认知特点安排教学内容,本教材中将“编译预处理”内容放到附录中去,由于此部分内容跨越了很多知识点,如果单独作为一章内容,那么只能将这部分内容放到较靠后的位置,但基础知识的学习中也牵涉到编译预处理的内容,因此,将这部分内容放到附录中,可以根据需要及时查询。读者可以根据实际情况安排学习编译预处理内容的进度。

本书可供高等院校、计算机水平考试培训、各类成人教育学校作为开设程序设计课程

的教材,也可供计算机爱好者自学。

本书由黄维通、鲁明羽、关敬敏、关继来、邱春风等编写,同时参与程序调试的有关发达、张光、王春明,参加代码校验的有顾雷等,感谢清华大学出版社为编写本书给予的大力支持。

由于作者水平有限,加上时间仓促,书中缺点和错误在所难免,恳请读者批评指正并通过作者电子信箱告知。

作者联系信箱: huangweitong@tsinghua.org.cn

黄维通

2005年8月



第 1 章 C 语言的基本概念	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	1
1.2 几个基本概念	2
1.2.1 程序与程序设计	2
1.2.2 算法	3
1.2.3 数据结构	3
1.3 软件工程的观念	3
1.4 C 语言程序的基本标识符	4
1.5 C 程序的几个简单实例	5
1.6 C 语言程序的编译和执行	7
习题	8
第 2 章 C 语言程序的基本数据类型及其运算	10
2.1 C 语言的数据类型	10
2.1.1 数据类型的一般概念	10
2.1.2 常量	11
2.2 数据类型及变量	14
2.2.1 基本数据类型	14
2.2.2 变量及变量的定义	15
2.2.3 变量的初始化	15
2.2.4 数据类型转换	16
2.3 运算符和表达式	18
2.3.1 运算符和表达式概述	18
2.3.2 赋值运算符和赋值表达式	19
2.3.3 算术运算符和算术表达式	20
2.3.4 关系运算符和关系表达式	22
2.3.5 逻辑运算符和逻辑表达式	23

2.3.6	条件运算符	23
2.3.7	其他运算符	24
2.4	位运算符.....	25
2.4.1	按位取反运算符	25
2.4.2	移位运算符	26
2.4.3	按位“与”、按位“或”、按位“异或”	26
2.5	C语言基本输入输出函数	28
2.5.1	字符输入输出函数	28
2.5.2	格式化输入输出函数	29
习题		34
第3章	基本控制结构及其应用	36
3.1	算法及结构化程序设计.....	36
3.1.1	算法及其特征	36
3.1.2	算法的类型与结构	38
3.2	顺序结构程序设计.....	39
3.3	分支结构程序设计.....	40
3.3.1	if分支	40
3.3.2	if-else分支	40
3.3.3	多分支 if...else if...else 形式	41
3.3.4	条件分支的嵌套	42
3.3.5	开关分支	44
3.4	循环结构程序设计.....	47
3.4.1	while语句	47
3.4.2	do-while语句	47
3.4.3	for语句.....	48
3.4.4	三种循环的比较	49
3.4.5	多重循环	50
3.4.6	循环和开关分支的中途退出	51
3.4.7	goto语句	52
3.4.8	结构化程序设计注意事项	53
3.5	结构化程序设计及应用举例.....	54
习题		55
第4章	数组及其应用	59
4.1	一维数组.....	59
4.1.1	一维数组的定义	59
4.1.2	一维数组的初始化	60

4.1.3 一维数组的引用	61
4.1.4 一维数组的应用举例	62
4.2 多维数组	64
4.2.1 多维数组的定义	64
4.2.2 多维数组的存储形式	64
4.2.3 多维数组的引用	65
4.2.4 多维数组的初始化	65
4.2.5 数组应用举例	67
4.3 字符型数组与字符串	68
4.3.1 字符型数组的概念	68
4.3.2 字符型数组的初始化	69
4.3.3 字符型数组的输入输出	70
4.3.4 字符型数组的应用举例	71
4.4 综合应用举例	72
习题	74
第5章 函数及其应用	76
5.1 函数的定义与调用	76
5.1.1 C源程序的结构	76
5.1.2 函数的定义	77
5.1.3 函数的调用	79
5.2 函数间的信息传递方式	82
5.2.1 实参-形参之间的信息传递	83
5.2.2 函数调用结果的返回	86
5.3 函数与数组	89
5.3.1 数组元素作实参	89
5.3.2 一维数组名作实参	89
5.3.3 多维数组名作参数	91
5.3.4 字符数组作函数的参数	92
5.4 递归函数与递归调用	93
5.4.1 递归的概念	93
5.4.2 递归程序的设计	94
5.5 变量的存储类型及作用域	95
5.5.1 auto 型变量	96
5.5.2 extern 型变量	96
5.5.3 静态变量	99
习题	101

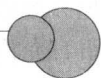
第 6 章 指针	104
6.1 指针的基本概念及定义方式	104
6.1.1 指针的基本概念.....	104
6.1.2 指针的定义.....	105
6.1.3 指针的初始化.....	105
6.1.4 指针运算符.....	107
6.2 指针的运算	107
6.2.1 指针的算术运算.....	107
6.2.2 关系运算.....	109
6.2.3 指针的赋值运算.....	109
6.3 指针与数组	110
6.3.1 指向一维数组的指针.....	110
6.3.2 指向多维数组的指针.....	112
6.4 字符指针和字符串	114
6.5 指针数组	115
6.5.1 指针数组的概念.....	115
6.5.2 指针数组的应用.....	117
6.5.3 指针数组在带形参的 main 函数中的应用	117
6.6 指针在函数参数传递中的应用	119
6.7 指针型函数	120
6.7.1 指针型函数的定义和引用.....	120
6.7.2 指针型函数的应用举例.....	120
6.8 多级指针	121
6.8.1 多级指针的概念及定义.....	121
6.8.2 多级指针应用举例.....	122
6.9 指向函数的指针	123
6.9.1 指向函数的指针的概念.....	123
6.9.2 指向函数的指针的应用.....	124
6.10 动态指针.....	126
6.10.1 动态内存分配的概念.....	126
* 6.10.2 动态内存分配的应用.....	127
习题.....	129
第 7 章 结构体、联合体和枚举	132
7.1 结构体的说明和定义	132
7.1.1 什么叫结构体.....	132
7.1.2 结构体的说明及结构体变量的定义.....	132

7.2	结构体成员的引用与结构体变量的初始化	135
7.2.1	结构体成员的引用	135
7.2.2	结构体变量的初始化	136
7.3	结构体数组	137
7.3.1	结构体数组的定义及初始化	137
7.3.2	结构体数组的应用举例	137
7.4	结构体指针	138
7.4.1	结构体指针及其定义	138
7.4.2	通过指针引用结构体成员	139
7.5	结构体在函数间的传递	141
7.5.1	结构体变量的传递	141
7.5.2	结构体数组在函数间的传递	144
7.6	结构体指针型和结构体型函数	145
7.6.1	结构体指针型函数	145
7.6.2	结构体型函数	147
7.7	结构体嵌套	148
7.7.1	什么是结构体嵌套	148
7.7.2	嵌套结构体类型变量的引用	149
7.7.3	结构体嵌套应用举例	150
7.8	联合体	151
7.8.1	联合体的说明及联合体变量的定义	151
7.8.2	使用联合体变量应注意的问题	154
7.9	枚举类型	156
7.9.1	枚举类型数据的概念及其定义	156
7.9.2	枚举型变量的使用	157
7.10	自定义类型	159
7.10.1	自定义类型及其表示形式	159
7.10.2	自定义类型的优点	160
	习题	161
第8章	排序及查找算法及其实现	164
8.1	排序概述	164
8.1.1	排序的概念	164
8.1.2	排序的方法	164
8.2	冒泡排序法的设计及其实现	165
8.2.1	冒泡算法设计思想	165
8.2.2	冒泡算法的实现	165
8.3	选择排序法的设计及其实现	167

8.3.1	选择排序法设计思想	167
8.3.2	选择排序法设计的实现	167
8.4	插入排序法的设计及其实现	169
8.4.1	插入排序法设计思想	169
8.4.2	插入排序法的实现	169
8.5	SHELL 排序法的设计及其实现	171
8.5.1	SHELL 排序法设计思想	171
8.5.2	SHELL 排序法的实现	172
8.6	查找概述	173
8.6.1	顺序查找及其应用	173
8.6.2	折半查找及其应用	175
	习题	176
第 9 章	文件操作	177
9.1	文件概述	177
9.2	文件的基本操作	178
9.2.1	文件的打开	178
9.2.2	文件的关闭	179
9.2.3	文件的重命名	180
9.3	文件的读写操作	180
9.3.1	文件的非格式化读写	180
9.3.2	文件的格式化写操作	185
9.3.3	文件的格式化读操作	187
9.4	文件的定位	189
9.4.1	fgetpos 函数	189
9.4.2	fsetpos 函数	189
9.4.3	ftell 函数	189
9.4.4	fseek 函数	189
9.4.5	feof 函数	191
	习题	191
附录 A	预处理命令的应用	193
附录 B	Turbo C 3.0 的上机过程	205
附录 C	Turbo C 3.0 程序的调试	210
附录 D	ASCII 码表	217
	参考文献	218

第1章

C语言的基本概念



1.1 C语言的发展与特点

1.1.1 C语言的发展

C语言是目前国际上广泛流行的一种结构化的程序设计语言,它不仅是开发系统软件的很好的工具,而且也是开发应用软件的很好的程序设计语言。

C语言是在20世纪70年代初由美国贝尔实验室的Dennis M. Ritchie设计的,当时只是为了描述和实现UNIX操作系统而提供一种工作语言而设计的,到了1973年,K. Thompson和Dennis M. Ritchie两人合作把UNIX的90%以上内容用C语言进行了改写,即大家熟知的UNIX第五版。多年来,UNIX V系统配备的C语言一直是C的公认标准,在Brian W. Kernighan和Dennis M. Ritchie合著的《C Programming Language》中对此也进行了介绍。随着微型计算机的普及,出现了较多的C语言系统,其中多数系统接受的源程序都能高度兼容,然而,由于没有统一的标准,兼容总会存在一定的差异,这种差异对于计算机应用技术的发展,显然不利。为了克服此不利局面,ANSI于1983年成立了专门定义C语言标准的委员会,花了6年时间使C语言迈向了标准化。随着UNIX第六版的推出和面向对象程序设计技术的出现,C语言的突出优点引起了人们的普遍关注,ANSI C标准于1989年被采用,该标准一般称为ANSI/ISO Standard C,于是,1989年定义的C标准定义为C89,到了1995年,出现了C的修订版,其中增加了一些库函数,出现了初步的C++,在此基础上,C89成为C++的子集。此后,C语言不断发展,在1999年又推出了C99,C99在基本保留了C的特性的基础上增加了一系列新的特性,随后又几经修改和完善,它也从而面向过程的编程语言发展到面向对象的程序设计语言,目前可在微型计算机上运行的C语言版本主要有Turbo C,UNIX C,Visual C/C++等版本。

1.1.2 C语言的特点

C语言之所以能被推广并被广泛使用,概括地说主要有如下特点。

1. 中级语言

C语言通常被称为中级语言,这并不意味着C语言的功能不如高级语言,而是因为它

把高级语言的先进思想与汇编语言的控制和灵活性有机地结合起来。作为中级语言,C语言允许对位、字节和地址这些计算机功能中的基本成分进行操作。

2. 结构化语言

C语言是便于进行模块化程序设计的语言,C语言程序是由一系列函数组成的,这种结构便于把一个大型程序划分为若干个功能相对单一的、相对独立的模块,模块间通过函数调用来实现相互连接,函数允许一个程序中的各个任务被分别定义和编码,从而使程序模块化。因此,对于大型程序的设计来说,函数的设计至关重要,因为函数是C语言的独立子程序,是一种构件,程序的所有操作都在其中发生,同时要求独立的函数其代码不能意外地影响他人的程序。

3. 可移植性

C语言程序具有较高的移植性。可移植性指的是,可以把为某种计算机编写的软件运行在另一种机器或操作系统上,如在UNIX操作系统下写的程序,能够方便地在Windows 2000下运行,这个程序就是一个可移植的程序。C语言程序的输入输出功能是由独立于C语言的库函数来实现,这样就使C语言程序本身不依赖于硬件系统,也便于在不同的机器系统间移植。

1.2 几个基本概念

在介绍C语言程序设计之前,先来认识几个有关程序设计的基本概念。

1.2.1 程序与程序设计

学习C语言程序设计,就一定会涉及程序编写。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)。通常,一个计算机程序主要描述两部分的内容,其一是描述问题的每个对象及它们之间的关系,其二是描述对这些对象进行处理的处理规则。其中关于对象及它们之间的关系涉及到数据结构的内容,而处理规则却是求解某个问题的算法。因此,对程序的描述,经常可以理解为:

$$\text{程序} = \text{数据结构} + \text{算法}$$

一个设计合理的数据结构往往可以简化算法,而且一个好的程序有可靠性、易读性、可维护性等良好特性。

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,然后再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。通常,程序设计是很讲究方法的,一个好的设计方法能够大大提高程序的高效性、合理性。通常程序设计有一整套完整的算法,也称为程序设计方法学,因此有人提出如下关系:

$$\text{程序设计} = \text{数据结构} + \text{算法} + \text{程序设计方法学}$$

程序设计方法学在程序设计中被提到比较高的位置,尤其对于大型软件,更是如此,

它是软件工程的组成部分。

1.2.2 算法

所谓算法,就是问题的求解方法。通常,一个算法由一系列求解步骤来完成。正确的算法要求组成算法的规则和步骤的意义是惟一确定的,不能存在二义性,而且这些规则指定的操作是有序的,必须按算法指定的操作顺序执行,并能够在有限的执行步骤后给出正确的结果。

算法是不允许存在二义性和模糊概念的,否则编码工作将无法进行。如“加一吨水”有明确的概念,这在程序的计算或控制中是可以操作的,但如果说“加一些水”,那就无法操作了,日常生活中,多少认为是“一些”呢?没有明确的定义,毕竟程序是根据人的具体要求来完成相应工作的。

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,细化不可能一步到位,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。

1.2.3 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,程序中的数据结构描述了程序中的数据间的组织形式及相互关系。

数据结构与算法密不可分,一个良好的数据结构,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,因此两者是相辅相承的。

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完全不同。许多时候可以“殊途同归”,也就是说,对同一个问题的求解,允许有不同的算法,也允许存在不同的数据结构,而依不同算法编写的操作代码,其执行效率也就不一样。

1.3 软件工程的概

软件工程是开发、运行、维护和修复软件的系统方法。著名学者 Boehm 认为,软件工程就是运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料。软件工程包括方法、工具和过程三个要素。

软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务,如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等。

软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。目前,已经开发出了许多软件工具,而且已经有人把诸多软件工具集成起来,使得一种工具产生的信息可以为其他工具所使用,这样就建立起一种称之为计算机辅助软件工程(CASE)的软件开

发支撑系统。CASE 将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

软件工程的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。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理及软件开发各个阶段完成的内容。

开发一个大型应用程序,就必须从软件工程的思想出发,对需求进行系统性的分析与设计,而不是匆匆编码。今天大家开始学习 C 语言,只是学习编程开发的第一步,要把自己培养成合格的程序员或系统分析员,就必须牢固树立软件工程的思想,系统、全面地看待每一个细小的问题,这样才能更好地掌握编程技术和方法。有关软件工程的进一步知识,请读者阅读相应的资料。

1.4 C 语言程序的基本标识符

任何一种高级语言,都有自己的基本词汇符号和语法规则,程序代码都是由这些基本词汇符号(有时也称为标识符)并根据该语言的语法规则编写而成,C 语言也不例外。C 语言规定了其所需的基本字符集和标识符。

1. 字符集

满足 C 语言文法要求的字符集如下:

- (1) 英文字母 $a \sim z, A \sim Z$;
- (2) 阿拉伯数字 $0 \sim 9$;
- (3) 特殊符号如表 1-1 所示。

表 1-1 C 语言的特殊符号

+	-	*	/	%	=
{	}	(	)	[	]
_(下划线)	'(单引号)	.	:	?	~
<	>	&	;	"	
!	#	空格		^	

2. 标识符

C 语言的标识符主要用来表示常量、变量、函数和类型等的名字,它由下划线或英文字母构成,它包括如下三类。

1) 保留字

拥有特定含义的标识符称为保留字。C 语言不允许用户把保留字当作变量名使用,C 语言的保留字都用小写英文字母表示,如表 1-2 所示。