

· 高等学校计算机基础教育教材精选 ·

C程序设计教程与实训 (第2版)

高敬阳 李芳 主编
朱群雄 主审

TP311.62-43

清华大学出版社

· 高等学校计算机基础教育教材精选 ·

C程序设计教程与实训 (第2版)

高敬阳 李芳 主编
马静 李国捷 尤枫 吴蕾 编著
朱群雄 主审

清华大学出版社
北京

内 容 简 介

本书通过案例教学的方式,由浅入深,让学生在模仿—训练—应用的过程中,快速掌握程序设计的基本思想和基本方法。

本书共9章,主要内容包括C程序概述、用C语言编写简单程序、分支结构、循环结构、数组、函数、指针、结构体与共用体、文件。各章均给出了内容丰富又有代表性的例题,全部程序都在Visual C++ 6.0中调试通过,同时也对Visual C++环境进行了介绍。书后配有各章习题分析及部分习题答案,供读者参考。此外,还提供了教学资源丰富的课程网站作为教学活动的课外补充。

本书可作为高等学校各专业C程序设计课程的教材,也可以作为各类计算机培训班的教材和成人教育同类课程教材及自学教材。

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

C程序设计教程与实训/高敬阳,李芳主编;马静等编著. —2版. —北京:清华大学出版社,2010.3

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-22204-0

I. ①C… II. ①高… ②李… ③马… 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(2010)第031449号

责任编辑:袁勤勇

责任校对:白蕾

责任印制:杨艳

出版发行:清华大学出版社

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

社 总 机:010-62770175

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

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:14.25 字 数:325千字

版 次:2010年3月第2版 印 次:2010年3月第1次印刷

印 数:1~3000

定 价:23.00元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:036920-01

出版说明

高等学校计算机基础教育教材精选

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,包括面向各高校开设的计算机必修课、选修课以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本,并保持不断更新),坚持宁缺毋滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是 jiaoh@tup.tsinghua.edu.cn。联系人:焦虹。

清华大学出版社

前言

C 程序设计教程与实训(第 2 版)

程序设计能力是计算机基础教育的重要组成部分,是高等学校学生应具备的基本技能之一。程序设计知识的学习有助于使学生真正理解计算机工作原理,了解计算机解决问题的方法,有效训练学生的逻辑思维和抽象思维,同时开阔学生的视野,培养丰富的想象力和创造力,最终帮助学生更好地使用计算机解决本专业科研、工作和生活中的相关问题。

程序设计是既有挑战性,又颇有成就感的过程。有经验的程序员在重新审视 C 语言的学习时,常常会感觉这门课程其实很简单。然而,在实际面对初学者的教学过程中,却面临着比想象中多得多的困难。通常的问题是,开课之初学生有很大的热情,但随着学习的深入,到了循环、数组部分,有些学生仍然迟迟不能入门,慢慢地失去了学习的兴趣,造成恶性循环,最终甚至放弃了该课程的学习。学生普遍反映对于抽象的 C 程序设计课程难于找到入门的捷径。这些情况的出现,原因是多方面的。其中很重要的原因就是长期以来,程序设计课程过多强调语言本身及其表达细节,忽视了程序设计的本质,造成很多学生过多地陷入具体细节的旋涡里,无法站在一定的高度欣赏程序设计的美。同时 C 程序设计又是一门实践性很强的课程,学生必须通过较多的编程训练才能掌握。因此,如果能让学生一开始就很清楚自己要做的事情,循序渐进地领会程序设计的精妙,在实践中形成良好的程序设计风格,并自始至终兴趣浓厚,相信 C 语言的教学工作将会收到事半功倍的效果。

鉴于此,我们决定从教材入手,转换思路。在教材的编写过程中,本着从始至终简化语法,培养学生动手编程能力的初衷,力争独辟蹊径,写出特色,让学生了解 C 程序的编写其实远没有传说中的那样困难。

本教材全书共分为 9 章,涵盖了 C 程序设计教程应包含的基本内容。并将文件的基本使用方法提前至数组一章,让学生提前了解文件的应用,并在后续知识的学习中反复使用,加深理解。

同时,每章均由引例开始,引出该章将要引入的新知识,采用“提出问题—分析问题—引入新知识—解决问题—模仿编程—总结提高”这样一个循序渐进、螺旋式上升的教学模式。将一个个典型的、针对性强的、贴近现实或贴近专业的案例程序设计作为贯穿始终的主线,将课程内容抽茧拔丝般解析开来。学生可通过课堂练习题、课后习题和课后综合提高题等几个环节提升程序设计能力,达到由浅入深、举一反三进行程序设计实训的目的。

此外,本书重要章节(如循环、数组、函数、指针等章节)的课后习题均增加了面向各类

专业的应用与提高的部分习题,为各类专业学生了解计算机在本专业的应用提供感性认识。

本书还提供了教学资源丰富的课程网站。资源网站(<http://202.4.152.136/>)有电子教案、CAI 动画课件、自我测试题等供下载。

培养学生程序设计能力的方法仍在研究和探索中,最大限度地提高学生的学习效果是我们永恒的奋斗目标。

本书由从事了多年计算机基础课程教学、具有丰富教学实践经验的一线教师集体编写完成。第 1、2 章由李国捷编写,第 3 章及附录由高敬阳编写,第 4 章由吴蕾编写,第 5、9 章由马静编写,第 6、7 章由李芳编写,第 8 章由尤枫编写。全书由高敬阳、李芳组织编写并统稿,由朱群雄教授主审。

由于作者水平有限,书中难免有错误和不妥之处,恳请读者批评指正。

作者联系信箱:gaojy@mail.buct.edu.cn,lifang@mail.buct.edu.cn。

作者

2010 年 1 月

目录

C 程序设计教程与实训(第 2 版)

第 1 章 概述	1
1.1 引例	1
1.2 C 语言程序的基本结构	2
1.3 程序设计基本概念	3
1.3.1 程序	3
1.3.2 程序设计	3
1.3.3 程序设计语言	3
1.4 C 语言的发展与特点	4
1.4.1 C 语言的发展	4
1.4.2 C 语言的特点	4
1.5 C 语言的字符集	5
1.6 C 语言的词法符号	5
1.7 运行 C 程序的步骤和开发环境	7
1.7.1 运行 C 程序的步骤	7
1.7.2 集成开发环境	8
本章小结	12
习题 1	12
第 2 章 用 C 语言编写简单程序	13
2.1 引例	13
2.2 数据类型	14
2.2.1 C 语言的数据类型	14
2.2.2 常量和变量	14
2.2.3 整型数据	15
2.2.4 实型数据	16
2.2.5 字符型数据	17
2.2.6 变量赋初值	18

2.3	运算符与表达式	18
2.3.1	C语言运算符简介	18
2.3.2	算术运算符与算术表达式	18
2.3.3	赋值运算符与赋值表达式	19
2.3.4	逗号运算符与逗号表达式	20
2.4	各类数值型数据间的混合运算	20
2.4.1	自动类型转换	20
2.4.2	强制类型转换	21
2.5	数据的输入输出	21
2.5.1	标准字符输入输出函数	22
2.5.2	格式输出函数 printf()	22
2.5.3	格式输入函数 scanf()	23
2.6	顺序结构程序设计	25
2.6.1	C语言的语句	25
2.6.2	顺序结构程序设计举例	26
	本章小结	27
	习题 2	27
第 3 章	分支结构程序设计	29
3.1	引例	29
3.2	关系运算和逻辑运算	30
3.2.1	关系运算	30
3.2.2	逻辑运算	30
3.3	if 语句	31
3.3.1	if-else 形式	31
3.3.2	if 形式	33
3.3.3	if 语句的嵌套	34
3.3.4	if-else if 形式	37
3.3.5	条件运算符及条件表达式	39
3.4	switch 语句	39
	本章小结	42
	习题 3	43
第 4 章	循环结构程序设计	45
4.1	引例	45
4.2	while 语句	46
4.3	do-while 语句	49
4.4	for 语句	51

4.4.1	for 语句格式	51
4.4.2	for 语句实例	52
4.4.3	三种循环语句的比较	54
4.5	循环嵌套	54
4.6	break 和 continue 语句	57
4.6.1	break 语句	57
4.6.2	continue 语句	59
4.7	goto 语句	60
4.8	循环应用	61
	本章小结	66
	习题 4	66
第 5 章	数组	69
5.1	引例	69
5.2	一维数组	70
5.2.1	数组的概念	70
5.2.2	一维数组的定义	71
5.2.3	一维数组的引用	71
5.2.4	一维数组的初始化	72
5.2.5	一维数组的应用	73
5.3	二维数组	76
5.3.1	二维数组的定义	76
5.3.2	二维数组的引用和初始化	76
5.3.3	二维数组的应用	78
5.4	字符数组	80
5.4.1	字符数组的定义	80
5.4.2	字符数组的初始化和引用	81
5.4.3	字符数组和字符串	82
5.4.4	字符串处理函数	84
5.5	用文件处理数据	87
5.6	综合应用实例	88
	本章小结	90
	习题 5	91
第 6 章	函数	94
6.1	引例	94
6.2	函数的定义及调用	95
6.2.1	函数的定义	95

6.2.2	函数的调用	96
6.2.3	函数声明	98
6.2.4	两种特殊的函数	100
6.3	函数的递归调用	100
6.4	数组作为函数的参数	103
6.4.1	一维数组作为函数的参数	104
6.4.2	函数间的参数传递	105
6.4.3	二维数组作为函数的参数	106
6.4.4	字符数组作为函数的参数	108
6.5	程序的多文件组织	109
6.5.1	多文件组织	109
6.5.2	VC 6.0 集成环境中多文件组织的应用	110
6.6	作用域和存储类型	112
6.6.1	变量的作用域	112
6.6.2	变量的存储类型	114
6.6.3	函数的存储类型	118
6.7	函数的应用	119
6.7.1	函数应用实例	119
6.7.2	函数的通用性	121
	本章小结	122
	习题 6	122

第 7 章 指针 125

7.1	引例	125
7.2	指针变量的定义和引用	126
7.2.1	指针变量的定义	126
7.2.2	指针变量的引用	127
7.2.3	指针变量的应用	128
7.3	指针与数组	129
7.3.1	指向数组元素的指针	129
7.3.2	指针与字符串	131
7.3.3	指针与二维数组的关系	132
7.4	指针与函数	135
7.4.1	指针作为函数的参数	135
7.4.2	返回值为指针的函数	136
7.4.3	函数指针	137
7.5	指针数组和指向指针的指针	138
7.5.1	指针数组	138

7.5.2 指向指针的指针	140
7.5.3 命令行参数与字符指针数组	141
本章小结	142
习题 7	142
第 8 章 结构体与共用体	145
8.1 引例	145
8.2 结构体类型的声明和结构体类型变量的定义	146
8.2.1 结构体类型的声明	146
8.2.2 结构体类型变量的定义	147
8.2.3 结构体变量的引用	148
8.2.4 结构体变量的初始化	149
8.3 结构体数组	150
8.3.1 定义结构体数组	150
8.3.2 结构体数组的初始化	151
8.3.3 结构体数组的引用	151
8.4 结构体指针	153
8.4.1 结构体指针变量的定义与引用	153
8.4.2 指向结构体数组的指针	155
8.4.3 结构体变量和指向结构体的指针作为函数参数	156
8.5 动态存储分配	157
8.6 链表	158
8.6.1 链表的概念	158
8.6.2 动态链表	159
8.6.3 单向链表中的插入与删除	161
8.7 共用体	164
8.7.1 共用体的概念	164
8.7.2 共用体变量的引用	165
8.8 枚举类型	166
8.8.1 枚举类型的定义	166
8.8.2 枚举变量的定义	166
8.8.3 枚举变量的赋值和使用	166
8.9 用 typedef 命名类型	168
本章小结	169
习题 8	170
第 9 章 文件	174
9.1 文件概述	174

9.1.1	文件的概念	174
9.1.2	缓冲文件系统	175
9.1.3	文件结构和文件类型指针	175
9.2	文件的打开和关闭	176
9.2.1	文件打开函数 fopen	176
9.2.2	文件关闭函数 fclose	177
9.3	文件的读写	177
9.3.1	文件的字符输入输出函数	178
9.3.2	文件的字符串输入输出函数	179
9.3.3	文件的格式化输入输出函数	180
9.3.4	文件的数据块输入输出函数	182
9.4	其他文件函数	184
9.5	应用举例	185
	本章小结	188
	习题 9	188
附录 A	ASCII 码表	189
附录 B	运算符的优先级和结合性	190
附录 C	常用库函数	192
附录 D	预处理命令	197
D.1	宏定义	197
D.2	文件包含	199
D.3	条件编译	199
附录 E	各章习题解析与提示	201
	参考文献	211

本章主要内容：

- C 语言程序的基本结构；
- 程序设计基本概念；
- C 语言的发展与特点；
- C 语言的字符集；
- C 语言的词法符号；
- 运行 C 程序的步骤和方法。

1.1 引 例

首先让我们看两个用 C 语言编写的程序。

例 1-1 在屏幕上显示一行信息“This is the first C program!”。

程序代码如下：

```
#include <stdio.h>                /* 编译预处理命令 */
void main()                        /* 定义主函数 main() */
{
    printf ("This is the first C program!\n"); /* 调用 printf() 函数输出文字 */
}
```

运行结果：

```
This is the first C program!
```

程序中的 `#include <stdio.h>` 是编译预处理命令，因为后面调用的 `printf()` 函数是 C 语言提供的标准输出函数，在系统文件 `stdio.h` 中声明。

程序中 `/* …… */` 是程序的注释，用来说明程序的功能。

程序中的 `void main()` 定义了一个名称为 `main()` 的函数，关键字 `void` 表示函数无返回值。

一对大括号把构成函数的语句括起来，称为函数体。例 1-1 的函数体只有一条语句。

语句“printf(“This is the first C program!\n”);”由函数调用和分号两部分组成。“printf(“This is the first C program!\n”)”是一个函数调用,它的作用是将双引号中的内容原样输出;“\n”是换行符,即在输出“This is the first C program!”后换行;而分号表示该语句的结束。

例 1-2 求两数之和。

程序代码如下:

```
#include <stdio.h>                /* 编译预处理命令 */
void main()                        /* 定义主函数 main() */
{
    int a,b,sum;                   /* 定义变量 a、b、sum 为整型 */
    a=66;                          /* 为变量 a 赋值 */
    b=88;                          /* 为变量 b 赋值 */
    sum=a+b;                       /* 将 a 与 b 的和赋值给变量 sum */
    printf ("sum is %d\n",sum);    /* 调用 printf() 函数输出 sum 的值 */
}
```

运行结果:

```
sum is 154
```

程序的第 4 行定义三个变量 a、b、sum 为整型(int)变量;程序的第 5、6 行都是赋值语句,分别使 a 的值为 66,b 的值为 88;程序的第 7 行也是一条赋值语句,使 sum 的值为 a+b;程序的第 8 行是输出函数调用语句,其中的“%d”是输入输出格式说明,表示“以十进制整数类型”输入输出相应的数据(详见第 2 章);括号内逗号的右端 sum 是要输出的变量,现在它的值为 154(即 66 与 88 之和);此函数调用后,双引号括起来的“sum is”按原样输出,在“%d”的位置上显示变量 sum 的值 154。

1.2 C 语言程序的基本结构

通过 1.1 节中的两个例子,我们可以看到 C 语言程序以下的结构。

① C 程序由函数组成,函数是程序的基本单位。main 是一个特殊的函数名,一个程序总是从 main()函数开始执行。

② 函数由函数首部和函数体两部分组成。

函数首部用于定义函数的名称、函数的返回值类型和各种参数名称及数据类型(也可能没有参数及数据类型)。例如 void main()即是函数首部。

③ 函数体一般包括数据定义部分和执行部分,它们都是 C 语句。

④ 每条语句用分号“;”作结束符,分号是 C 语句必不可少的组成部分。

⑤ 在 C 语言中,一行可以写多条语句,一条语句也可写成几行。如将例 1-2 的三条语句合并成如下形式:

```
a=66;b=88;sum=a+b;
```

其结果和输出格式均不改变。

⑥ 可以对 C 程序中的任何部分做注释。一个好的、有使用价值的程序应当加上必要的注释,以改善程序的可读性和可维护性。注释可以占一行的一部分,也可以单独占一行,还可以占若干行。

1.3 程序设计基本概念

初学者应对下面几个有关程序设计的基本概念有所了解。

1.3.1 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)。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规则。其中关于对象及它们之间的关系涉及数据结构的内容,而处理规则却是求解某个问题的算法。因此,对程序的描述,经常有如下等式:

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

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。

1.3.2 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,一个好的设计思想方法能够大大提高程序的效率和合理程度。通常程序设计有一套完整的方法,也称为程序设计方法学,因此有人提出如下关系:

$$\text{程序设计} = \text{数据结构} + \text{算法} + \text{程序设计方法} + \text{语言工具和环境}$$

程序设计方法学在程序设计中提到比较高的位置,尤其对于大型软件更是如此。

1.3.3 程序设计语言

为了描述程序所制定的一组规则,即语法规则(主要包括词法规则与句法规则)。就像汉语与英语都有各自一整套的语法规则一样,众多的计算机语言,如 Basic 语言、Fortran 语言以及我们将要学习的 C 语言也都有各自一整套的语法规则。

1.4 C语言的发展与特点

1.4.1 C语言的发展

C语言是一种国际上广泛流行的、深受程序员喜爱的程序设计语言。

1967年,英国剑桥大学的 Martin Richards 在 ALGOL 60 的 CPL 语言基础上推出了 BCPL (Basic Combined Programming Language) 语言。

1970年,美国贝尔实验室的 Ken Thompson 在 BCPL 语言的基础上,设计了既简单又接近硬件的 B 语言(以 BCPL 首字母命名)。

1972年,贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上发明了 C 语言(以 BCPL 第二个字母命名)。

1978年, Brian W. Kernighan 与 Dennis M. Ritchie 合作写出了著名的 *The C Programming Language*。该书成为后来广泛使用的 C 语言版本的基础,称为标准 C。

1983年,美国国家标准协会(ANSI)为 C 语言制定了一套标准,称为 ANSI C。

1987年,ANSI 又公布了 87 ANSI C(新标准)。

1990年,国际标准化组织(ISO)接受 87 ANSI C 为 ISO 标准。目前流行的 C 语言版本都以此为基础。

1980年,贝尔实验室的 Bjarne Stroustrup 及其同事对 C 语言进行了改进,并将类的概念扩充到 C 语言中,在 1983年由 Rick Maseitti 提议正式命名为 C++ 语言。

C++ 是 C 的超集,C 是 C++ 的基础,用 C 语言编写的许多程序不经修改就可以在 C++ 环境下运行。因此,学习 C 语言可为进一步学习 C++ 及其他相关语言打下坚实的基础。

本教程以 ANSI C 标准为基础,书中的例题均在 Microsoft Visual C++ 6.0 集成环境下运行测试过。

1.4.2 C语言的特点

与其他高级语言相比,C语言的主要特点如下。

1. C语言是结构化、模块化的程序设计语言

C语言通过 9 种结构控制语句可描述各种结构的程序;以函数作为程序的基本单位,从而可实现模块化的程序设计。

2. C语言有强大的处理能力,适用面广

C语言既具有高级语言的功能,又能像低级语言一样对计算机最基本的工作单元(位、字节和地址)进行直接操作。因此,它既适宜编写大型系统程序,又适宜编写小型控

制程序，也适用于科学计算，并具有强大的图形处理功能。

3. C 语言语句简洁、紧凑，使用方便、灵活

C 语言一共只有 32 个保留字和 9 种控制语句，程序书写形式自由，压缩了一切不必要的成分。

4. 目标代码的效率

用 C 语言程序生成的目标代码的效率可达到汇编语言目标代码效率的 80% ~ 90%。

5. 可移植性强

C 语言的输入输出不依赖于计算机硬件，使之能适应多种操作系统，如 DOS、UNIX、Windows 等，也适应多种机型。从而便于在各种不同的机器间实现程序的移植。

由于 C 语言具有上述特点，因此它得到了广泛的应用。

1.5 C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空白符、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可显示的图形符号。

① 英文字母：小写字母 a~z、大写字母 A~Z。

② 阿拉伯数字：0~9。

③ 空白符：空格符、制表符、换行符等统称为空白符。

空白符只在字符常量和字符串常量中起作用。在其他地方出现时，只起间隔作用，编译程序忽略它们。因此在程序中使用空白符与否，对程序的编译不发生影响，但在程序中适当的的地方使用空白符将提高程序的清晰性和可读性。

④ 标点和特殊字符：

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

1.6 C 语言的词法符号

词法符号是最小的词法单元。

C 语言的词法符号分为以下几类：关键字、标识符、运算符、分隔符、常量和注释符。

1. 关键字

关键字是 C 语言规定的具有特定意义的字符串，也称为保留字。C 语言有以下 32 个