

计算机与信息技术专业应用教材

C 程序

设计教程

(基于 Visual C++ 平台)

李春葆 编著



清华大学出版社

► 计算机与信息技术专业应用教材

C 程序设计教程

(基于 Visual C++ 平台)

李春葆 编著

清华大学出版社

北 京

内 容 提 要

本书以 Visual C++ 6.0 为环境, 讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧。

本书系统性、实用性强, 内容组织上层次分明, 结构清晰, 并强调用软件工程思想指导程序开发, 以培养学生编写“大程序”的能力。全书共 13 章, 分别介绍了 C 语言的基本概念、数据类型、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、编译预处理、结构体和共用体、位运算、文件以及软件开发过程。每章后均有小结, 并附有相当数量的练习题和上机实验题。

本书可作为大专院校各专业 C 语言程序设计课程的教材, 也可供准备参加计算机等级考试和考研的读者阅读参考。

版权所有, 盗版必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目 (CIP) 数据

C 程序设计教程/李春葆编著. —北京: 清华大学出版社, 2004

计算机与信息技术专业应用教材

ISBN 7-302-08618-4

I. C… II. 李… 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 (2004) 第 043613 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社总机: 010-62770175

客户服务: 010-62776969

组稿编辑: 夏非彼

文稿编辑: 陈洁

封面设计: 付剑飞

版式设计: 科海

印 刷 者: 北京市耀华印刷有限公司

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 23.5 字数: 572 千字

版 次: 2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷

书 号: ISBN 7-302-08618-4/TP·6177

印 数: 1~6000

定 价: 29.00 元

(如有印装质量问题, 我社负责调换)

前 言

C 语言程序设计是高等院校计算机专业及相关专业重要的专业基础课,其目的是培养学生的程序设计理念、掌握程序设计的基本方法,为后继课程(数据结构、面向对象程序设计、操作系统、编译原理和软件工程等)打下坚实的基础。

C 语言程序设计课程的前期课程通常有计算机导论(或计算机文化基础)。目前,各大专院校的计算机导论课程的教学内容都是基于 Windows 98/2000/XP 操作系统,但讲授到 C 语言程序设计课程时,又大多以 Turbo C 为环境,即回到了 DOS 操作系统环境下,这样造成两个层次脱节。有一部分学校使用的教材是基于 Turbo C 环境的,而学生上机实验环境又采用 Visual C++(VC++)等,虽然两者类似,但毕竟有较大的差异,不便于提高学生的动手能力。本教材完全以 VC++为环境讲授 C 语言程序设计的基本思想、方法和解决实际问题的技巧,并比较 Turbo C 和 VC++之间的一些差异,这样也兼顾以 Turbo C 为学习环境的读者。

另外,作者深切感受到学习 C 语言程序设计不仅要掌握 C 语言的语法要点和编程规范,更重要的是领会结构化程序设计思想,综合运用 C 语言知识开发“大型”程序,为此,特地编写了“软件开发过程”一章(第 13 章),通过一个较大的实例指导如何运用软件工程的思想进行程序开发。

本书具有以下特点:

- 内容组织上层次分明,结构清晰。
- 叙述深入浅出,表述严谨。如变量声明和变量定义是两个不同的概念,前者不分配内存空间,后者分配内存空间,有些教科书统称为变量说明,本书作了严格区别。
- 精选大量的习题,便于练习提高。
- 强调实验动手能力的提高,每章给出与内容相一致的上机实验题。与本书配套的《C 程序设计教程学习与上机指导》不仅给出全部练习题和上机实验题的参考答案,还详尽地给出了使用 VC++6.0 编写程序的操作步骤。
- 书中精编了大量的实例,这些实例体现了良好的程序设计风格。
- 以 VC++为环境,便于进一步学习面向对象程序设计,为掌握用 VC++进行实际软件开发打下基础。
- 强调用软件工程思想指导程序开发,以培养学生编写“大程序”的能力。

为了便于教学和自学,与本书配套的教学工具有:

- 《C 程序设计教程学习与上机指导》
- 全书的源程序(文件名为源程序开头注释中给定的文件名)

其中,书中的源程序可以登录到 <http://www.khp.com.cn> 网站下载。

书中的绝大多数程序（除少数以 Turbo C 为环境的程序外）在 VC++6.0 系统中已编译调试通过，这些程序均冠以“VC++6.0”标志。

为了便于讲授和阅读，在版式上进行了如下安排：

注意 ► 指出在学习中的注意点。

说明 ► 标识一些重要说明信息。

特别说明 ► 给出了 VC++ 不同于 Turbo C 的语法等规定。

目前，C 语言程序设计课程的课时安排大多为 54 课时和 72 课时两种，下表给出了作者推荐的课时分配方案，供任课教师和自学的学生参考。

章次	内容	54 课时			72 课时		
		讲授	实验	小计	讲授	实验	小计
第 1 章	C 语言概述	2		2	2		2
第 2 章	数据类型及其运算	2		2	2		2
第 3 章	顺序结构程序设计	2	2	4	2	2	4
第 4 章	选择结构程序设计	2		2	2	2	4
第 5 章	循环结构程序设计	2	2	4	4	2	6
第 6 章	数组	4	2	6	4	2	6
第 7 章	函数	4	2	6	6	4	10
第 8 章	指针	4	2	6	6	4	10
第 9 章	编译预处理	2	2	4	2		2
第 10 章	结构体和共用体	4	2	6	6	4	10
第 11 章	位运算	2		2	2		2
第 12 章	文件	4	2	6	4	2	6
第 13 章	软件开发过程	2	2	4	4	4	8
合计		36	18	54	46	26	72

尽管作者教学多年并试图努力编写好每个知识要点，但由于水平有限，书中难免存在错误和不足之处，敬请读者和同行批评指正。

李春葆
2004 年初春于珞珈山下

目 录

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	2
1.1.1 C 语言的发展	2
1.1.2 C 语言的特点	2
1.2 C 语言程序的结构和语法规则	3
1.2.1 C 语言程序的基本结构	3
1.2.2 C 语言函数的一般结构	6
1.2.3 C 语言函数的语法规则	7
1.3 C 程序编译和开发过程	8
1.3.1 C 程序编译过程	8
1.3.2 C 程序开发过程	8
1.4 小结	9
练习题 1	9
上机实验 1	10
第 2 章 数据类型及其运算	12
2.1 标识符	13
2.2 C 语言的数据类型	14
2.3 常量	16
2.3.1 整型、实型及符号常量	16
2.3.2 字符型常量	17
2.4 变量	19
2.4.1 变量的概念	20
2.4.2 变量的定义与初始化	20
2.5 运算符与表达式	21
2.5.1 C 运算符概述	21
2.5.2 算术运算符与算术表达式	22
2.5.3 表达式中数据间的混合运算与类型转换	25
2.5.4 赋值运算符与赋值表达式	28
2.5.5 条件运算符和条件表达式	30
2.5.6 逗号运算符和逗号表达式	31
2.6 小结	31
练习题 2	32

上机实验 2	35
第 3 章 顺序结构程序设计	36
3.1 程序的三种基本控制结构	37
3.2 顺序执行语句	38
3.3 格式化输出函数 printf()	40
3.3.1 printf()函数的一般格式	40
3.3.2 格式指示符	40
3.3.3 使用说明	42
3.4 格式化输入函数 scanf()	43
3.4.1 scanf()函数的一般格式	43
3.4.2 格式指示符	44
3.4.3 使用说明	45
3.5 单个字符的输入/输出函数	48
3.6 小结	49
练习题 3	49
上机实验 3	55
第 4 章 选择结构程序设计	56
4.1 关系运算及其表达式	57
4.1.1 关系运算符及其优先次序	57
4.1.2 关系表达式	57
4.2 逻辑运算及其表达式	58
4.2.1 逻辑运算及其优先次序	58
4.2.2 逻辑表达式	59
4.3 if 语句	60
4.3.1 单分支 if 语句	60
4.3.2 双分支 if 语句	60
4.3.3 多分支 if 语句	63
4.4 switch 语句	64
4.5 小结	68
练习题 4	69
上机实验 4	74
第 5 章 循环结构程序设计	75
5.1 while 语句	76
5.2 do-while 循环语句	78
5.3 for 语句	81
5.4 break 和 continue 语句	83

5.5 循环结构的嵌套	89
5.6 goto 语句	91
5.7 循环结构程序设计应用	92
5.7.1 穷举法	92
5.7.2 迭代	96
5.8 小结	97
练习题 5	98
上机实验 5	104
第 6 章 数组	105
6.1 一维数组	106
6.1.1 一维数组的定义	106
6.1.2 一维数组元素的引用	106
6.1.3 一维数组的初始化	106
6.1.4 一维数组的应用	108
6.2 二维数组	116
6.2.1 二维数组的定义	116
6.2.2 二维数组元素的引用	117
6.2.3 二维数组的初始化	117
6.3 字符数组和字符串数组	120
6.3.1 字符数组	120
6.3.2 字符串数组	126
6.4 小结	128
练习题 6	129
上机实验 6	135
第 7 章 函数	136
7.1 函数的定义与调用	137
7.1.1 函数定义	137
7.1.2 函数调用	138
7.1.3 函数的返回值与函数类型	140
7.1.4 被调函数声明	142
7.1.5 函数应用举例	143
7.2 外部函数和内部函数	149
7.2.1 外部函数	149
7.2.2 内部函数	149
7.3 内部变量和外部变量	151
7.3.1 内部变量	151
7.3.2 外部变量	152

7.4 变量的存储类型	153
7.4.1 内部变量的存储类型	154
7.4.2 外部变量的存储类型	156
7.5 函数的数据传递	157
7.5.1 传值调用	157
7.5.2 传址调用	159
7.5.3 外部变量传递数据	160
7.6 数组作为函数参数	161
7.6.1 数组元素作为函数参数	161
7.6.2 数组名作为函数参数	161
7.7 递归函数	164
7.7.1 递归模型	164
7.7.2 递归的执行过程	164
7.8 小结	167
练习题 7	169
上机实验 7	177
第 8 章 指针	178
8.1 指针和指针变量	179
8.2 指针变量的定义及其运算	180
8.2.1 指针变量的定义	180
8.2.2 指针运算符	181
8.2.3 指针变量的初始化	183
8.2.4 指针变量的运算	184
8.3 指针变量和数组	187
8.3.1 指针变量与一维数组	187
8.3.2 指向多维数组元素和指向分数组的指针	192
8.4 字符串指针变量和字符串	198
8.4.1 字符串的表示和引用	198
8.5 指针数组	202
8.6 多级指针变量	205
8.7 指针与函数	207
8.7.1 指针型函数	207
8.7.2 指向函数的指针变量	209
8.8 命令行参数	212
8.9 小结	213
练习题 8	215
上机实验 8	224

第 9 章 编译预处理	225
9.1 宏定义	226
9.1.1 无参宏定义	226
9.1.2 带参宏定义	228
9.2 条件编译	230
9.3 文件包含	233
9.4 小结	234
练习题 9	235
上机实验 9	238
第 10 章 结构体和共用体	239
10.1 结构体类型声明和结构体变量定义	240
10.1.1 结构体类型声明	240
10.1.2 结构体变量的定义	241
10.1.3 结构体变量的引用和初始化	242
10.1.4 结构体变量作为函数参数	244
10.2 结构体数组	245
10.2.1 结构体数组的定义	245
10.2.2 结构体数组的引用	246
10.2.3 结构体数组的初始化	246
10.3 结构体指针变量	248
10.3.1 结构体指针变量定义	248
10.3.2 结构体数组指针	249
10.3.3 结构体指针变量作为函数参数	250
10.3.4 结构体数组作为函数参数	251
10.4 结构体的应用: 链表	253
10.4.1 C 语言动态分配函数	253
10.4.2 单链表及其基本运算的实现	254
10.5 共用体	260
10.5.1 共用体类型的定义	260
10.5.2 共用体变量的定义	261
10.5.3 共用体变量的引用和初始化	262
10.6 枚举类型	266
10.6.1 枚举类型的声明和变量定义	266
10.6.2 有关枚举类型数据的操作	268
10.7 用户定义类型	270
10.8 小结	272
练习题 10	273

上机实验 10	282
第 11 章 位运算	283
11.1 位运算符和位运算	284
11.1.1 位运算符	284
11.1.2 位运算符的运算功能	284
11.2 位段结构	288
11.3 小结	291
练习题 11	291
上机实验 11	293
第 12 章 文件	294
12.1 文件概述	295
12.1.1 文件的分类	295
12.1.2 文件的操作流程	296
12.1.3 文件缓冲区	297
12.1.4 流和文件指针	297
12.2 文件的打开和关闭	298
12.2.1 文件的打开	298
12.2.2 文件的关闭	300
12.3 文件的顺序读写	300
12.3.1 文件的字符读/写函数	300
12.3.2 文件的字符串读/写函数	304
12.3.3 文件格式化读写	306
12.4 文件的随机读写	308
12.4.1 文件定位操作	308
12.4.2 文件的随机读写	310
12.5 文件检测函数	315
12.6 小结	316
练习题 12	317
上机实验 12	324
第 13 章 软件开发过程	325
13.1 软件开发步骤	326
13.1.1 软件计划	326
13.1.2 需求分析	326
13.1.3 软件概要设计	327
13.1.4 软件详细设计	333
13.1.5 编码	334

13.1.6 测试.....	335
13.2 用 C 语言开发软件实例.....	335
13.2.1 系统功能.....	335
13.2.2 系统分析.....	336
13.2.3 系统设计.....	336
13.2.4 编码.....	338
13.2.5 测试.....	359
上机实验 13.....	360
附录 A C 语言运算符及优先级.....	361
附录 B 部分字符与 ASCII 代码对照表.....	362
参考文献.....	363

第 1 章

C 语言概述

计算机语言是人们描述计算过程即程序的规范书写语言。程序是计算机处理对象和计算规则的描述。语言的基础是一组记号和规则，根据规则由记号构成记号串的总体就是语言。C 语言是一种面向过程的高级程序设计语言。C 语言的语法源自 ANSI C 标准（ANSI 即美国国家标准协会）。本章主要介绍 C 语言程序的结构、书写规则和 C 程序开发过程等。

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

1.1.2 C 语言的特点

1.2 C 语言程序的结构和语法规则

1.2.1 C 语言程序的基本结构

1.2.2 C 语言函数的一般结构

1.2.3 C 语言函数的语法规则

1.3 C 程序编译和开发过程

1.3.1 C 程序编译过程

1.3.2 C 程序开发过程

1.4 小结

练习题 1

上机实验 1

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

在 C 语言诞生以前,系统软件(例如操作系统)主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件,其可读性和可移植性都很差,而一般的高级语言又难以实现对计算机硬件的直接操作(对硬件直接操作正是汇编语言的优势),于是人们盼望有一种兼有汇编语言和高级语言特性的新语言。C 语言就是在这种背景下应运而生的。

C 语言是贝尔实验室于 20 世纪 70 年代初期研制出来的,并随着 UNIX 操作系统的日益广泛使用而迅速得到推广。后来,C 语言又被多次改进,并出现了多种版本。20 世纪 80 年代初(1983 年),ANSI 根据 C 语言问世以来的各种版本对 C 语言进行了改进和扩充,制定了 ANSI C 标准(并于 1989 年再次做了修订)。本书以 ANSI C 标准为基础介绍 C 语言。

目前,在微机上广泛使用的 C 语言编译系统有 Borland C++(简称 BC++)、Turbo C(简称 TC)、Microsoft Visual C++(简称 VC++)等。虽然它们的基本部分都是相同的,但还是有一些差异,所以读者应注意自己所使用的 C 编译系统的特点和相关规定(可参阅相应编译系统配备的使用手册和用户指南了解这些信息)。本书选定的上机环境是 Microsoft Visual C++ 6.0 系统(简称 VC++6.0)。

1.1.2 C 语言的特点

C 语言是近年来较流行的高级程序设计语言之一,许多大型软件均是用 C 语言编写的(如 UNIX 操作系统)。C 语言同时具有汇编语言和高级语言的双重特性。具体来说,C 语言的主要特点如下:

- C 语言是一种模块化的程序设计语言。模块化的基本思想是将一个大的程序按功能分割成一些模块,使每一个模块都成为功能单一、结构清晰、容易理解的函数。
- 语言简洁,结构紧凑,使用方便、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。
- 运算符极其丰富,数据处理能力强。C 语言一共有 45 种运算符,例如,自增(++)和自减(--)运算符、复合赋值运算符、位运算符及条件运算符等。同时,C 语言可以实现其他高级语言较难实现的功能。
- 可移植性好。C 语言程序基本上可以不作任何修改,就能运行于各种不同型号的计算机和各种操作系统环境上。
- C 语言提供了某些接近汇编语言的功能,例如可以直接调用系统功能,实现对硬件的操作,从而有利于编写系统软件。

总之,由于 C 语言的上述特点,使得 C 语言越来越受到广泛的重视。

在 C 语言的基础上,1983 年贝尔实验室又推出了 C++ 语言。C++ 语言进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。

C语言是C++语言的基础，C++语言和C语言在很多方面是兼容的。因此，掌握了C语言，再进一步学习C++语言就更容易和便利，并能达到事半功倍的效果。

1.2 C语言程序的结构和语法规则

1.2.1 C语言程序的基本结构

先看看下面几个简单的示例，读者对C语言程序会有一个基本的了解。在此基础上，再进一步了解C语言程序的语法和书写规则。

【例 1.1】求三个数的平均值的C语言程序。

【解】程序如下：

```
/*文件名: exam1_1.cpp*/
/*功能: 求三个数的平均值*/
#include <stdio.h>
main()
{
    float a,b,c,avg;
    a=85.5;
    b=92.5;
    c=71;
    avg=(a+b+c)/3;
    printf("avg=%f\n",avg);
}
```

/*main() 称为主函数*/
/*定义 a,b,c,avg 为实型变量*/
/*计算平均值*/
/*在屏幕上输出 avg 的值*/

程序运行结果:

avg=83.000000

【例 1.2】输出两个数中的较大值的C语言程序。

【解】程序如下：

```
/*文件名: exam1_2.cpp*/
/*功能: 输出两个数中的较大值*/
#include <stdio.h>
int max(int,int);
main()
{
    int n1,n2;
    scanf("%d,%d",&n1,&n2);
    printf("Max=%d\n",max(n1,n2));
}
int max(int x,int y)
{
```

/*max() 函数声明*/
/*由键盘输入 n1 和 n2 的值*/
/*用户设计的函数 max()*/

```

    if (x>y)
        return x;
    else
        return y;
}

```

程序运行结果:

4,8✓ (注:“✓”表示按回车键,下划线部分表示由用户通过键盘输入,全书统一采用这种表示法。)

Max=8

【例 1.3】用于计算 $n!$ 之值的 C 语言程序。

【解】程序如下:

```

/*文件名: exam1_3.cpp*/
/*功能: 求 n!*/
#include <stdio.h>
int fun(int);          /*fun() 函数声明*/
main()
{
    int n,m;            /*定义变量 n,m 为整型*/
    printf("n:");       /*输出"n:"*/
    scanf("%d",&n);     /*输入 n 值*/
    m=fun(n);           /*调用函数 fun()*/
    printf("%d!=%d\n",n,m); /*输出结果*/
}
int fun(int x)          /*定义函数 fun(),形参 x 为整型,返回整型值*/
{
    int i,y;            /*定义变量 i 和 y 为整型*/
    y=1;                /*变量 y 赋初值 1*/
    for (i=1;i<=x;i++)  /*通过循环求 y 值*/
        y=y*i;
    return y;           /*返回 y 的值*/
}

```

程序运行结果:

n:5✓
5!=120

在以上三个示例中,例 1.1 所示的 C 语言程序仅由一个 main()函数构成,它相当于其他高级语言中的主程序;例 1.2 和例 1.3 所示的 C 语言程序由一个 main()和一个其他函数(用户自己设计的函数)构成。函数 max()相当于其他高级语言中的子程序。由此可见,一个完整的 C 语言程序结构有以下两种表现形式。

(1) 仅由一个 main()函数(又称主函数)构成,如下所示:

```
main()
{
    :
}
```

(2) 由一个 `main()` 函数和若干个其他函数结合而成, 如下所示。其中, 自定义函数由用户自己设计。

自定义函数 1, 自定义函数 2, ... 的声明

```
main()
{
    :
}
自定义函数 1
自定义函数 2
:

```

结合以上示例, 可以看出 C 语言程序结构有以下基本特点:

- C 语言程序是由函数 (如 `main()` 函数和 `max()` 函数等) 组成的, 每一个函数完成相对独立的功能, 函数是 C 语言程序的基本模块单元。`main()` 是函数名, 后面的一对圆括号 “()” 是用来写函数的参数的, 参数可以有也可以没有 (前面例子程序中的 `main()` 函数都没有参数), 但圆括号不能省略。
- 一个 C 程序总是从 `main()` 函数开始执行的, 而不论 `main()` 函数在整个程序中的位置如何。
- C 语言编译系统区分字母大小写。C 语言把同一字母的大小写视为两个不同的字符, 并规定每条语句均以分号 (;) 结束。分号是语句不可缺少的组成部分。
- C 语言程序中所调用的函数, 既可以是由系统提供的库函数, 也可以是由设计人员自己根据需要而设计的函数。例如, 在前面各例中, `printf()` 函数是 C 语言编译系统库函数中的一个函数, 它的作用是在屏幕上按指定格式输出指定的内容; 例 1.2 中, `max()` 函数是由用户自己设计的函数, 它的作用是计算两个数中的较大值。

说明 对于自定义函数, 若其定义放在调用语句之前时, 可以不需要进行函数声明; 若其定义放在调用语句之后时, 还需要进行函数声明。例如, 例 1.2 程序中, `max()` 函数在 `main()` 函数之后定义, 但在 `main()` 中调用, 所以程序前面包含以下函数声明语句:

```
int max(int, int);
```

C 语言编译系统中有许多以 .h 为扩展名的文件 (称为头文件)。在这些头文件中, 对相应的库函数及符号常量等进行了声明和定义。例如, 一般的 C 源程序都需要用库函数 `scanf()`、`printf()` 等进行输入/输出操作, 因此在源程序的最前面一般需要使用语句

```
#include <stdio.h>
```