

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
计算机应用

C语言程序设计

案例教程(含实验)

仇芒仙 张丽华 主编
朱有根 刘小晶 杜选 副主编

清华大学出版社



高等学校教材
计算机应用

C 语言程序设计 案例教程(含实验)

仇芒仙 张丽华 主编
朱有根 刘小晶 杜选 副主编

清华大学出版社
北京

内 容 简 介

C 语言是计算机科学与技术、信息管理与信息系统等专业的一门专业基础课,也是计算机程序设计者的首选课程。

本书共分两篇内容。第一篇主要是 C 语言的基本知识,系统介绍了 C 语言程序设计的基本知识点: C 程序的基本构成、数据类型和表达式应用、程序的控制结构、指针、函数、数组、结构体与共用体、文件、位运算和高级编程知识等。为了帮助读者提高编程能力,第二篇内容中对常见的 45 个 C 程序典型案例进行了解析,并配合 C 语言基本知识点的学习阐述了 10 个上机实验项目,以巩固读者的所学知识,提高用 C 语言解决实际问题的能力。

全书文字通俗,简明易懂,便于自学。既可作为计算机、信息管理及其相关专业的本科教材,也可以作为计算机等级考试(C 语言)的辅导教材,又可以作为计算机业余爱好者的自学用书。

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 语言程序设计案例教程:含实验/仇芒仙,张丽华主编. —北京:清华大学出版社,2009.12
(高等学校教材·计算机应用)

ISBN 978-7-302-21069-6

I. C… II. ①仇… ②张… 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(2009)第 168403 号

责任编辑:闫红梅 薛 阳

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

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 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:23.5 字 数:585 千字

版 次:2009 年 12 月第 1 版 印 次:2009 年 12 月第 1 次印刷

印 数:1~3000

定 价:35.00 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:034374-01

C语言程序设计是计算机科学与技术、信息管理与信息系统等专业的一门专业基础课,也是数据结构、编译原理、操作系统、Java 语言程序设计等课程的前修课程。C 语言与其他高级语言相比有其显著的优点,它不仅是系统描述语言,而且是通用的程序设计语言。因此,它既适合编写系统软件,又适合编写应用软件,它的应用领域是很宽广的,是一种理想的结构化语言,适用于计算机及相关专业教学的需要。

有许多新型语言都是 C 语言的衍生,如 C++、Java 等。可以说掌握了 C 语言,就掌握了多门语言,经过简单的学习,学生就可以用这些新型语言去开发程序了。目前社会招聘程序员,很多单位也将 C 语言作为最基本的开发管理工具对应聘者进行考核。通过对本课程的学习,可以培养学生用计算机解决实际问题的能力,提高学生的编程和软件开发能力,为各相关专业后续课程的学习与实践打下良好的基础。

目前市场上有许多种 C 语言教材或参考书,但大多数都是传统的编写模式,首先是入门知识的介绍,然后是基础知识,最后是程序结构体系等。这种方法以知识点为主线,过于拘泥语法细节,而忽视了程序本身的重要性,因此学生学完后还是不能用 C 语言完整地解决一个实际问题,其缺点是:内容烦琐、理论知识讲得多、联系生活的实例少,内容严重滞后,不便于学生学习。

鉴于以上存在的问题,为了更好地培养学生的编程能力,配合计算机科学与技术、信息管理与信息系统专业课程建设的需要,在建设“C 语言程序设计”精品课程的同时,写作组的老师们根据多年教学实践与经验,提出了编写适合信息时代学生学习的《C 语言程序设计案例教程》,为课堂教学、实验指导、学生自学提供立体化的教学资源。

本书共分两篇,具体内容如下。

第一篇是 C 语言程序设计教程。第 1 章和第 2 章主要介绍 C 程序的基本组成与 C 语言的实验环境;第 3 章是数据类型及应用;第 4 章是 C 语言的三种程序结构,即顺序结构、选择结构与循环程序结构的应用;第 5 章是函数及应用,主要介绍了函数的定义、函数的调用、全局变量与局部变量的应用及指针作为函数参数的引用;第 6 章是数组及应用,主要介绍一维数组、二维数组及字符数组的定义、初始化及引用,着重对指针访问数组进行了举例;第 7 章是结构体与共用体的定义与引用,对单链表的建立、插入与删除进行了案例应用;第 8 章是文件的读写及应用;第 9 章是位运算操作;第 10、11 章是 C 语言的高级编程知识。

第二篇是 C 语言典型程序解析与 C 语言实验指南。为配合 C 语言程序设计课程的教学,进一步提高学生用 C 语言解决实际问题的能力,便于学生参加各种程序设计大赛,第 12 章对常用的 45 个典型实例进行了解题分析。每个实例都有算法分析、流程图、程序代码与运行结果。第 13 章是 C 语言实验指南,为配合《C 语言程序设计案例教程》教材中的知识点,对上机实验的 10 个项目进行了描述,实验内容也是分层次,题目难易程度不同,根据不同学生学习情况可以选择。

本书的特点如下。

(1) 以案例驱动知识。打破传统的教材写作与讲授模式,从问题案例、任务驱动入手,在提出解决问题算法的基础上,引出各个知识点,即以“案例—知识—总结”的模式进行编写,并举一反三,配有习题与实验项目,帮助读者巩固所学知识。

(2) 精简内容、缩短章节。把握“使用”与“够用”的尺度,以往的教材内容比较烦琐,语法较细,而程序编写思路不太明确。我们本着“精讲多练”的原则,删掉一些不必要的、应用滞后的内容,把重点放在培养学生“编程技巧与能力”上。

(3) 循序渐进,由浅入深。各章节从问题入手,并通过大量的思考题延伸知识点与新问题,做到“环环相扣,层层推进”,充分体现解析法的精髓,达到通俗易懂、由浅入深的效果,培养读者迁移知识的能力。

(4) 教学、实验相互配合,内容全面。为教师与学生提供了教学、自学、实践的配套资料,为读者掌握本课程内容与技能奠定良好的基础。本教材编写者都是多年在第一线教学的教师,他们总结多年教学的实践经验,将 C 语言课程设计的案例与设计项目融入本教材中。

(5) 将“指针”内容的知识点通过实例贯穿全书(从第 3 章到第 10 章的内容中),从第 3 章就引入“指针”的概念,以强调它的重要性,而且结合每章的内容都有不同层次的应用,这样避免了将“指针”放在一章中介绍,学生不好“消化”、“一头雾水”的现象。

(6) 案例分层次,便于不同学习水平的学生掌握。在介绍知识点时,尽量列举日常生活中的常见案例,而在实验中由浅入深分层次布置实验内容,使学习较好的学生能够做一些提高性的实验内容。在典型题解中再对难度较大的问题进行解析,这样既利于学生掌握一般知识点又可以让喜欢编程的学生更深入地学习,从而逐步提高程序设计的技巧。

本教材全部作者都是高等院校的教师。本书的出版是“C 语言程序设计”精品课程建设中所催生的产物,在课程建设过程中,编写组成员对教材内容、教学方法进行了多次的研讨,最后确定了以“案例驱动知识”的写作方法。

教材的讲授学时为 64~80,也可根据学生的实际情况进行选讲,甚至可以将第 9~11 章的内容作为自学,或作为课程设计时的参考内容。

本书为老师提供电子课件等教学资料,可从清华大学出版社的网站(www.tup.com.cn)下载。

本书第 1、2 章由仇芒仙教授编写;第 3、8 章由杜选副教授编写;第 4 章由魏远旺编写;第 5 章由龚迅炜、朱有根编写;第 6 章由殷海明编写;第 7 章由刘小晶副教授编写;第 9~11 章由李永刚编写。第 12 章由张丽华、朱有根编写;第 13 章由仇芒仙教授编写。全书由仇芒仙教授和张丽华副教授统稿。在本书出版之际,对在编写过程中给予帮助并提出了建设性意见的金镇、丁辉、郭步、殷联甫、邓昶、李海明等专家表示感谢。

本书参考了大量国内外的文献与资料,对它们的作者不能一一列出,在此对他们表示深深的谢意。

由于编写水平有限,书中疏漏之处在所难免,敬请各位读者及同仁不吝赐教,便于以后进一步的修订。

编 者

2009 年 6 月

第一篇 C 语言程序设计教程

第 1 章 C 程序概述	3
1.1 C 程序结构	3
1.1.1 C 程序实例	3
1.1.2 C 程序构成	4
1.1.3 书写程序时应遵循的原则	7
1.2 C 语言的发展	7
1.2.1 C 语言发展	7
1.2.2 C 语言的特点	8
1.3 数据在计算机中的存储形式与表示方法	9
1.3.1 内存的组织形式	9
1.3.2 原码、反码和补码	10
第 2 章 程序与算法	13
2.1 利用计算机解决问题的基本过程	13
2.2 程序的算法	15
2.2.1 算法的概念	15
2.2.2 算法的表示	15
2.3 结构化程序设计	18
2.3.1 结构化程序的基本结构	18
2.3.2 结构化程序设计的特点与方法	19
2.3.3 结构化程序设计的过程	19
2.3.4 程序设计的原则	20
2.4 C 程序的运行环境	22
2.4.1 Turbo C 2.0 实验环境	23
2.4.2 Visual C++ 6.0 实验环境	29

第3章 数据类型及其应用	36
3.1 常量	37
3.1.1 数值型常量	37
3.1.2 字符型常量	39
3.1.3 符号常量	40
3.2 变量	41
3.2.1 变量的要素	41
3.2.2 变量的分类	42
3.2.3 变量的地址——指针	45
3.3 运算符和表达式	46
3.3.1 运算符	47
3.3.2 算术运算符和算术表达式	48
3.3.3 赋值运算符和赋值表达式	49
3.3.4 复合赋值符和表达式	51
3.3.5 逗号运算符和逗号表达式	51
3.3.6 关系运算符和关系表达式	52
3.3.7 逻辑运算符和逻辑表达式	53
3.3.8 数据类型转换	54
3.4 数据的输出	56
3.4.1 printf()函数	56
3.4.2 格式字符串	57
3.4.3 字符输出函数 putchar()	59
3.5 数据的输入	59
3.5.1 scanf()函数	59
3.5.2 格式字符串	60
3.5.3 字符输入函数 getchar()	63
第4章 控制结构	67
4.1 顺序结构	67
4.2 选择结构	68
4.2.1 关系运算符及其表达式	68
4.2.2 逻辑运算符及其表达式	69
4.2.3 if 语句与条件运算符	70
4.2.4 switch 语句	74
4.3 循环结构	77
4.3.1 用 while 语句实现循环	77
4.3.2 用 do-while 语句实现循环	79
4.3.3 用 for 语句实现循环	81

4.3.4 循环嵌套	83
4.3.5 break 语句和 continue 语句	85
4.3.6 goto 语句	86
4.4 综合实例	87
第 5 章 函数	93
5.1 函数的定义与调用	93
5.1.1 程序解析	93
5.1.2 函数的定义	94
5.1.3 函数的调用	96
5.1.4 函数调用的过程	98
5.1.5 函数编程举例	101
5.2 变量与函数	103
5.2.1 局部变量和全局变量	103
5.2.2 变量生命周期和静态局部变量	106
5.3 函数的组织	110
5.3.1 程序解析——计算常用圆形体体积	110
5.3.2 函数的嵌套调用	112
5.3.3 文件包含	114
5.3.4 函数与程序文件模块	115
5.4 递归函数	116
5.4.1 程序解析	116
5.4.2 递归函数基本概念	116
5.5 指针与函数	120
5.5.1 指针作为函数参数	120
5.5.2 指针函数与函数指针	120
5.6 宏定义	125
5.6.1 宏基本定义	125
5.6.2 带参数的宏定义	126
5.7 编译预处理	128
第 6 章 数组	132
6.1 一维数组	132
6.1.1 一维数组的定义和引用	133
6.1.2 一维数组的初始化	134
6.1.3 一维数组的应用	135
6.2 字符串与字符数组	136
6.2.1 字符数组的定义和引用	137
6.2.2 字符串处理函数	137

6.2.3 字符数组综合举例	141
6.3 二维数组	142
6.3.1 二维数组的定义和引用	142
6.3.2 二维数组的初始化	143
6.3.3 二维数组的应用	143
6.3.4 多维数组	145
6.4 数组与指针	145
6.4.1 通过指针访问数组元素	145
6.4.2 指针数组	148
6.4.3 指向数组的指针	150
6.4.4 多级指针	151
6.4.5 main()函数的参数	151
6.4.6 与指针有关的一些问题	152
6.5 数组与函数	154
第7章 结构体与共用体	159
7.1 提出问题	159
7.2 结构体变量的定义和引用	160
7.2.1 结构体类型的定义	160
7.2.2 结构体变量的定义	161
7.2.3 结构体变量的引用	166
7.3 结构体与数组	168
7.3.1 结构体数组的定义及初始化	169
7.3.2 结构体数组的使用	169
7.4 结构体与指针	173
7.4.1 指向结构体变量的指针	173
7.4.2 指向结构体数组的指针	175
7.5 链表及其操作	176
7.5.1 链表概述	176
7.5.2 单链表的表示及实现	177
7.5.3 动态内存管理函数	179
7.5.4 单链表的基本操作	179
7.5.5 链表的应用举例	184
7.6 共用体与枚举	185
7.6.1 共用体	186
7.6.2 枚举类型	190
第8章 文件	197
8.1 文件的基本概念	197

8.2	文件的打开与关闭	198
8.2.1	文件打开函数 <code>fopen()</code>	198
8.2.2	文件关闭函数 <code>fclose()</code>	200
8.3	文件的读写	200
8.3.1	读字符函数 <code>fgetc()</code>	201
8.3.2	写字符函数 <code>fputc()</code>	202
8.3.3	读字符串函数 <code>fgets()</code>	204
8.3.4	写字符串函数 <code>fputs()</code>	205
8.3.5	数据块读写函数 <code>fread()</code> 和 <code>fwrite()</code>	206
8.3.6	格式化读写函数 <code>fscanf()</code> 和 <code>fprintf()</code>	207
8.4	文件的随机读写	208
8.5	文件检测函数	210
第9章	位操作	212
9.1	位运算符	212
9.1.1	按位与运算	213
9.1.2	按位或运算	213
9.1.3	按位异或运算	213
9.1.4	求反运算	214
9.1.5	左移运算	214
9.1.6	右移运算	215
9.2	位段	215
第10章	C语言图形编程	219
10.1	图形模式的初始化	219
10.2	屏幕文本模式和图形模式之间的切换	222
10.3	图形的坐标	224
10.4	屏幕和视口的设置与清除	226
10.5	颜色设置	227
10.6	图形文本	229
10.7	图形函数	230
10.7.1	像素函数	230
10.7.2	直线和线型函数	232
10.7.3	多边形函数	234
10.7.4	圆、弧和曲线函数	236
10.7.5	填充函数	239
10.8	动画与图像函数	241

第 11 章 C 语言综合应用	247
11.1 巧用数组设计算法	247
11.2 活用指针设计特效算法	249
11.3 用算法提高程序的速度	252

第二篇 C 语言典型程序解析与 C 语言实验指南

第 12 章 C 语言典型程序解析	257
第 13 章 C 语言实验指南	310
实验项目 1 C 程序的编辑、编译、链接和运行	310
实验项目 2 数据类型、运算符、表达式	312
实验项目 3 顺序结构程序设计	316
实验项目 4 分支结构程序设计	317
实验项目 5 循环结构程序设计	322
实验项目 6 函数及其应用	326
实验项目 7 数组及其应用	330
实验项目 8 指针及其应用	335
实验项目 9 结构体及其应用	338
实验项目 10 文件的应用	343
附录 A 常用的字符与 ASCII 码	345
附录 B C 语言中的关键字及其用途	347
附录 C 运算符的优先级和结合性	348
附录 D 常用库函数	350
参考文献	355

C语言程序设计教程

第1章 C程序概述

第2章 程序与算法

第3章 数据类型及其应用

第4章 控制结构

第5章 函数

第6章 数组

第7章 结构体与共用体

第8章 文件

第9章 位操作

第10章 C语言图形编程

第11章 C语言综合应用

C 程序概述

本章主要知识点：

- ◆ C 程序结构
- ◆ C 语言的发展
- ◆ 数据在计算机中的存储形式与表示方法

从 1946 年世界上第一台电子计算机诞生到现在,计算机的发展经历了 4 个阶段,迄今为止,仍然采用计算机之父“冯·诺依曼”的思想——存储程序的原理。即计算机要想做任何事情,都要按照一定的程序去做,也就是我们先编写好程序,然后输入到计算机中,计算机才能按照程序的要求去完成。因此想做一个能够驾驭计算机的高手,就一定要掌握用计算机语言编程的本领,而 C 语言是目前国际上广泛流行的计算机语言,它是步入程序设计殿堂的敲门砖,也是学好计算机相关专业后续课程的专业基础课,目前很多游戏软件、查杀病毒软件、工具软件、控制软件等都是用它编写的。

那么 C 语言能干什么?用 C 语言又如何编写程序呢?下面通过实例来了解。

1.1 C 程序结构

在了解 C 语言程序之前,我们先通过几个实例对 C 程序结构有一个初步认识。本节主要介绍 C 语言程序的基本结构。

1.1.1 C 程序实例

程序是计算机完成确定任务的具体步骤。

【实例 1-1】 在屏幕上输出“I am a student !”

```
/* 实例 1-1 */                                /* 注释语句 */
#include <stdio.h>                             /* 预处理命令 */
main()                                         /* 主函数 */
{
    printf("I am a student !");              /* 输出字符串 "I am a student !" */
}
```

分析说明：以上程序只用到输出库函数 printf,因此在程序开头一定要有 #include <stdio.h> 预处理命令,此命令的功能是将“stdio.h”包含在当前 C 语言源程序中,因为库函数 printf 属于“stdio.h”文件的一部分,换句话说就是建立了头文件“stdio.h”与当前 C 程序的链接。

点拨

在 Turbo C 2.0 环境下,可以省略此命令。

【实例 1-2】 求两个数中的最大数。

```
/* 实例 1-2 */
#include <stdio.h>
void main()                                /* 主函数 */
{
    int x,y,t;                             /* 说明部分,定义整型变量 x,y,t */
    int max(int,int);                     /* 函数声明 */
    printf(" please input x,y: ");        /* 在屏幕上显示字符串 "please input x,y: " */
    scanf("%d,%d",&x,&y);                 /* 输入 x,y 的值 */
    t = max(x,y);                         /* 赋值语句 */
    printf("x= %d,y= %d,max= %d",x,y,t); /* 输出 x,y,t 的值 */
}
int max(int a,int b)                      /* 函数 max */
{
    if(a>b)                               /* 条件语句 */
        return a;                       /* 条件成立,返回值 a */
    else
        return b;                       /* 条件不成立,返回值 b */
}
```

运行结果如图 1-1 所示。

分析说明：实例 1-2 是由两个函数组成,一个是主函数 void main(),一个是 int max(int a,int b)。void 是函数的类型,表示此函数执行后不会返回函数值,即“空函数”。int max 函数是求 a、b 两个数中最大者,函数返回值是整型的。

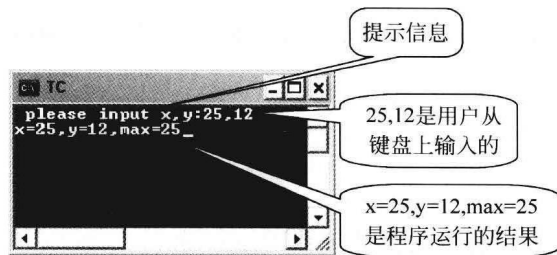


图 1-1 实例 1-2 的程序运行结果

1.1.2 C 程序构成

以上两个程序都是 C 语言的“源程序”。不难看出 C 语言程序的构成如下。

(1) C程序由函数组成。一个C程序至少包含且仅包含一个主函数 `main()`，也可以是包含一个主函数 `main()` 和若干个其他函数。C程序的运行是从 `main` 函数开始，结束在 `main` 函数。`main` 函数后面的 `()` 是表示函数的属性，是函数的标识。

函数由两部分组成：函数首部和函数体。实例 1-2 中的 `void main()`、`int max(int a, int b)` 都是函数首部，它表示函数的类型、函数名、函数参数及参数的类型。

例如 `max` 的首部为：

<code>int</code>	<code>max</code>	<code>(int</code>	<code>a,</code>	<code>int</code>	<code>b)</code>
↑	↑	↑	↑	↑	↑
函数的类型	函数名	参数的类型	函数参数	参数的类型	函数参数

`max` 的首部表示函数 `max` 返回值是整型的 `int`，并且有两个整型参数 `a` 和 `b`。

大括号“`{}`”中是函数的内容，也称函数体。函数体由一系列语句组成。

(2) C程序的语句，每个语句以分号结束。C程序的语句大概分为注释语句、说明语句、执行语句和空语句。

注释语句：由“`/ *`”与“`* /`”括起来的一串字符，字符串可以是中文字符也可以是英文字符。注释语句是非执行语句，其作用是方便别人阅读程序所加的说明。它可以出现在程序的任何地方。

说明语句：说明语句一般放在程序体的前面，比如“`int x, y, t;`”和“`int max(int, int);`”语句，前者是说明 `x, y, t` 是“`int`”类型，后者是说明 `max` 函数是“`int`”类型并带有两个“`int`”的参数，也可以写成“`int max(int a, int b);`”。

点拨

“`int max(int, int);`”语句也称函数声明语句。因 `max` 是被调函数，即它可以被主函数 `main` 或其他函数来调用。在上例中，主函数 `main` 在前，被调函数的定义在后，所以在程序主函数 `main` 中要先对 `max` 进行说明。

执行语句：是表示能执行特定的操作。

C语言的执行语句有表达式语句、函数调用语句、控制语句、复合语句、赋值语句。

(1) 表达式语句

表达式语句由表达式加上分号“`;`”组成。

其一般形式为：

表达式；

执行表达式语句就是计算表达式的值。例如：`a=3` 为一个赋值表达式，若在后面加上一个分号则构成赋值语句 `a=3;`；即一个表达式的最后加一个分号就成为一个语句。

(2) 函数调用语句

由函数名、实际参数加上分号“`;`”组成。

其一般形式为：

函数名(实际参数)；

执行函数语句就是调用函数体并把实际参数赋予函数定义中的形式参数，然后执行被调函数体中的语句，计算函数值并返回。