



21世纪高职高专信息技术实训教材

C程序设计 实训教程

北京希望电子出版社 总策划
杨金龙 主 编
魏爱敏 副主编
杨兴明 陈彦许 崔会军等 编 著



科学出版社
www.sciencep.com



21世纪高职高专信息技术实训教材

C程序设计 实训教程

北京希望电子出版社 总策划
杨金龙 主 编
魏爱敏 副主编
杨兴明 陈彦许 崔会军等 编 著



 科学出版社
www.sciencep.com

内 容 简 介

本书是21世纪高职高专信息技术实训教材中的一本,本书可以帮助读者更好地掌握C语言程序设计课程的基本内容和学习方法,提高上机实验能力和程序设计能力。

本书讲述了程序的开发过程及常见的错误、Turbo C 2.0集成开发环境的使用、在各种计算机系统中常用的库函数及使用说明、Turbo C 2.0编译错误信息及处理方法和屏幕、图形函数及用法等内容。

本书内容丰富、系统全面、概念清晰、实用性较强,是学习 C 语言的一本很好的实践参考书。

本书可作为高等职业院校、高等专科学校、成人高等学校、软件职业技术学院专业学生学习用书,也可以供继续教育学校以及程序爱好者使用。

需要本书或需要得到技术支持的读者,请与北京中关村 083 信箱(邮编 100080)发行部联系,电话:010-82702660 010-82702658, 010-62978181 转 103 或 238, 传真:010-82702698, E-mail: tbd@bhp.com.cn

图书在版编目(CIP)数据

C 程序设计实训教程/杨金龙主编. —北京:科学出版社,
2005.2

21 世纪高职高专信息技术实训教材

ISBN 7-03-014655-7

I. C... II. 杨... III. C 语言—程序设计—高等
学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 122598 号

责任编辑:王玉玲

/ 责任校对:佳 宜

责任印刷:媛 明

/ 封面设计:王 焱

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 2 月第 一 版 开本:787×1092 1/16

2005 年 2 月第一次印刷 印张:13 7/8

印数:1-5000 册 字数:318 000

定价:20.00 元

21世纪高职高专信息技术教材编委会名单

(排名不分先后)

主 任 高 林

副主任 谢玉声

袁启昌

胡伏湘

陆卫民

委 员

阮东波

龙 超

杨丽群

王东红

罗映峰

侯晓华

连晋平

冯矢勇

杨章静

唐伟奇

徐 萍

尹 静

慕东周

李 淼

田 更

朱作付

李超燕

吴 军

杨旭东

景鹏森

杨金龙

米 昶

陈孟建

崔俊杰

孙 杰

宗小翀

陈翠娥

唐燕青

韦 伟

陈 春

刘 毅

袁海宁

徐建华

邱建国

曹冬梅

郑明红

蒋建强

陈彦许

韩素华

王趾成

崔会军

张光瑞

总 序

高等职业教育目前已成为我国高等教育的重要组成部分,对于推动我国社会主义现代化建设起着不可忽视的作用。计算机教育在整个高职教育中有着举足轻重的地位,因为计算机的普及已经涉及到各个行业。对于传统的学习计算机知识的方法即理论为主、应用为辅的教学模式,相对高职教育来说有些不太适合,针对这种情况,就需要一些符合高职教育特点的教材来满足这种需求。

为解决教材供需不平衡的矛盾,北京希望电子出版社与全国高等学校计算机基础教育研究会高职高专专业委员会联合组织国内十几所高职院校,聘请“双师”型教师共同编写针对高职特点的教材 30 多种,以及实训类教材 10 多种,并请专家论证了本套教材的体系、风格、结构、内容等方面的可行性与可操作性。该系列教材体现“重在能力素质培养”的目标,结合教育部的教学大纲要求,在实用性、新颖性、可读性几个方面都有所突破。

高职教材建设是教学改革重要的环节,高等职业技术教育专业设置要与劳动力市场需求相结合,教学内容与国家职业标准相衔接。采取“订单教学”的校企合作培养模式,实行学业文凭和职业资格两种证书制度,使一线技术人才培养实现教学与市场“零距离”、毕业生上岗“零适应期”。这种以市场为导向实行的订单教学,能够直接为用人单位培养实用型人才,是一条富有特色的职教之路,可以保证同学们将来在就业和升学两条渠道上有最大的发展空间。所以,高校就要突出应用技能培养的办学特色,按照人才市场供求信号进行学科、专业和教学内容的调整,以适应社会需要。在培养学生的知识、能力、技能方面都要与其他综合性本科院校有所区别。

本系列教材就是遵循这种订单式教学的需要,一方面是设定系统理论知识的教材,这种教材的内容按照“必需、够用”的原则,构筑坚实的具有高职特色的理论体系基础;另一方面是训练职业动手能力的实训教材,按照“切实、实用”的原则,培养动手能力强的人才。以上两种教材相互配合,既可以单独使用,也可以配套使用。

高职教材建设还在探索中,如何能满足企业对人才的需求,跟上时代发展的步伐,这些都是亟需解决的问题。本丛书旨在抛砖引玉,希望更多的优秀教师参与到教材建设中来,真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议,为下一次的修订与改版做准备,使本丛书日臻完美。

若有投稿或建议,请发至本丛书出版者电子邮件: textbook@bhp.com.cn。

21 世纪高职高专信息技术教材编委会

前 言

C 语言是近年来在国内得到迅速推广的现代编程语言。为了满足计算机爱好者和学习者的需要，我们编写了这本书。本书可以帮助读者更好地掌握 C 语言程序设计课程的基本内容和学习方法，提高上机实验能力和程序设计能力。

本书由五部分构成：

第一部分为习题和参考答案，包括了很多比较典型的 C 程序设计习题。为了便于读者阅读和理解程序，在程序中给出了较多的汉字注释，字符串也多用汉字表示。对于编程题，参考答案并不是惟一正确的答案，同学们可以根据参考答案，拓展思路，编写出更好、更适合自己风格的程序。本部分所有程序都经过严格地调试，并全部调试通过。

第二部分为 C 语言实践环境，包括 Turbo C 2.0 集成开发环境的使用，程序的开发过程，程序编制过程中的常见错误。

第三部分为上机实验内容，共安排了 12 个实验。读者应在学习 C 语言程序设计时，重视提高自己的动手实践能力，培养独自编写程序及上机调试程序的能力。实验内容中既有直接给出源程序供读者上机调试的，也有给出问题让读者独立编写程序和上机调试的。这些实验由浅入深，循序渐进，内容丰富，结构合理。

第四部分为 C 库函数，包括在各种计算机系统中常用的库函数及使用说明。

第五部分为附录内容，包括 Turbo C 2.0 集成环境中编译错误信息及处理方法，屏幕、图形函数及其用法。

本书由杨金龙主编，魏爱敏老师编写了第四部分，参与本书编写的还有杨兴明、孙杰、苑伟、刘萍、刘春秋、高雪霞、邹岚、杨丽群、崔俊杰、赵明洁、杜海波、陈彦许、崔会军等。由于作者水平有限，书中难免会有不足之处，希望广大读者批评指正。

需要本书课件的老师请与本社联系。

编 者

目 录

第1章	C语言概述	1
第2章	C语言程序设计基础	3
第3章	选择结构程序设计	10
第4章	循环控制程序设计	17
第5章	数组	31
第6章	函数	44
第7章	编译预处理	63
第8章	指针	68
第9章	结构体、共用体和枚举类型	84
第10章	位运算	101
第11章	文件	105
第12章	概述	113
12.1	C程序的开发过程	113
12.1.1	源程序的编辑	113
12.1.2	源程序的编译	113
12.1.3	目标模块的链接与装配	114
12.1.4	程序的执行和调试	115
12.1.5	程序开发中库的维护与管理	117
12.2	程序开发过程中的常见错误	117
12.2.1	C程序调试中的常见错误分析	118
12.2.2	Turbo C 2.0 编译与连接时的常见错误	119
12.3	C语言常用语法提要	135
12.3.1	标识符	135
12.3.2	常量	135
12.3.3	表达式	136
12.3.4	数据定义	137
12.3.5	函数定义	138
12.3.6	变量的初始化	138
12.3.7	语句	138
12.3.8	预处理命令	139
第13章	Turbo C 2.0 集成开发环境	140
13.1	Turbo C 2.0 简介	140
13.2	Turbo C 2.0 集成开发环境	140
13.2.1	File (文件) 菜单	141
13.2.2	Edit (编辑) 菜单	142
13.2.3	Run (运行) 菜单	143
13.2.4	Compile (编译) 菜单	143
13.2.5	Project (项目) 菜单	144
13.2.6	Options (选择菜单)	145
13.2.7	Debug (调试)	149

13.2.8 Break/watch (断点及监视表达式)	149
13.3 Turbo C 2.0 的配置文件	150
上机实验目的和要求	151
实验 1 C 程序的运行环境和运行方法	152
实验 2 数据类型、运算符和表达式	153
实验 3 简单的 C 程序设计	156
实验 4 选择结构程序设计	158
实验 5 循环结构程序设计	159
实验 6 编译预处理	160
实验 7 数组	160
实验 8 函数	161
实验 9 指针	162
实验 10 结构体、共用体和枚举类型	163
实验 11 位运算	163
实验 12 文件	164
附录 A Turbo C 2.0 编译错误信息及处理方法	165
附录 B 屏幕、图形函数及用法	177
B.1 字符屏幕函数	177
B.1.1 文本窗口的定义	177
B.1.2 文本窗口颜色的设置	177
B.1.3 窗口内文本的输入/输出函数	179
B.1.4 有关屏幕操作的函数	180
B.2 图形函数	182
B.2.1 图形模式的初始化	182
B.2.2 独立图形运行程序的建立	185
B.2.3 屏幕颜色的设置和清屏函数	185
B.2.4 基本图形函数	187
B.2.5 封闭图形的填充	190
B.2.6 有关图形窗口和图形屏幕操作函数	194
B.2.7 图形模式下的文本输出	196
附录 C C 库函数	201
C.1 数学函数	201
C.2 字符函数和字符串函数	203
C.3 输入/输出函数	204
C.4 动态存储分配函数	208
C.5 时间函数	208
C.6 其他函数	210

第一部分

C 程序基础知识及习题

第 1 章 C 语言概述

1.1 请参照本章例题，编写一个程序，能显示以下信息：

My first C program!

I like it!

解：

```
main()
{
    /*两个简单的输出语句*/
    printf("My first C program!\n");
    printf("I like it! ");
}
```

运行结果：

My first C program!

I like it!

1.2 编写一个程序，已知其中变量 $a=2$ ， $b=5$ ，计算它们的和并赋值给 c ，然后输出 a ， b ， c 的值。

解：

```
main()
{
    int a,b,c;
    a=2;b=5;
    c=a+b;
    printf("a=%d,b=%d,c=%d\n",a,b,c); /*有参数的简单输出*/
}
```

运行结果：

$a=2, b=5, c=7$

1.3 编写一个程序，通过数学函数的调用来计算 625 的正平方根。

解：

```
#include <math.h> /*用 include 命令把函数所在的头文件(math.h)包含进来*/
main()
{
```

```

long int a;
a=sqrt(625);    /*用 sqrt 函数计算平方根*/
printf("625 的平方根=%d\n",a);
}

```

运行结果:

625 的平方根=25

1.4 从键盘输入两个整数，通过函数的调用来求得其中的大者，在屏幕上输出其值。

解:

```

main()
{
    int a,b,c;
    scanf("%d, %d", &a, &b);
    c = max(a, b);
    printf("max = %d", c);
}

int max(x,y)
int x, y;
{
    int z;
    if (x>y) z = x;
    else z = y;
    return z;
}

```

从以上 4 个习题我们可以对 C 语言的特点做以下几点总结。

(1) C 程序由函数构成，C 程序中有以下 3 种类型的函数。

1) **main()**: 主函数，开发系统提供的特殊函数，每一个 C 程序必须且只有一个 **main()** 函数。它代表程序开始执行的起始位置。

2) 发系统提供的函数，如 **printf()**、**scanf()** 等。Turbo C 开发系统提供 300 多个函数。程序员（或用户）自己设计的函数，如 **max()**。

3) 程序由函数构成的特点有利于实现程序的结构化（结构清晰）。

(2) 一个函数由以下两部分构成：

1) 函数的说明部分，包括函数名、函数类型（返回值类型）、函数属性（前面的例子中未使用函数的属性。属性包括 **near** 和 **far**、形式参数名、形式参数类型。

2) 函数体，大括号 **{ }** 中的部分。函数体中包含变量定义部分和执行部分。

(3) 一个 C 程序总是从 **main()** 处开始执行，而不管 **main()** 在源程序中的位置。

(4) C 程序书写格式自由，一个语句可以占多行，一行也可以有多个语句。

(5) 语句和数据定义后必须要有分号。如：**c = a + b ;** 。

(6) C 语言用函数进行输入输出，如 **scanf()**、**printf()**。

(7) C 语言用 **/* */** 作注释。

第2章 C语言程序设计基础

2.1 字符常量和字符串常量有什么区别？

解：

字符常量是一个字符，用单引号括起来，如'A'就是字符常量。

字符串常量是由0个或多个字符组合而成的，用双引号括起来，而且在存储时系统会自动在字符串最后加上一个结束符号'\0'，如""和"abc 123"都是字符串常量，一个是空字符串常量，另一个是非空字符串常量，它们在存储时末尾处会加上'\0'。

2.2 写出下列程序的运行结果。

```
main()
{
    char ch1='a',ch2='b',ch3='c';
    char ch4="\104",ch5="\101",ch6="\115";
    printf("%c\t%c\t%c",ch1,ch2,ch3);
    printf("\n\t\b%c%c%c",ch4,ch5,ch6);
}
```

解：

运行结果：

```
    a          b          c
          DAM
```

2.3 阅读下列程序写出输出结果。

```
main()
{ char ch1,ch2;
  ch1='a'; ch2='B';
  printf("ch1=%c,ch2=%c\n",ch1-32,ch2+32);
  printf("ch1+200=%d\n", ch1+200);
  printf("ch1+200=%c\n", ch1+200);
  printf("ch1+256=%d\n", ch1+256);
}
```

运行结果：

```
ch1=A,ch2=b
ch2+200=297
ch1+200=)
ch1+256=353
```

2.4 写出下列程序的输出结果。

```
main()
```

```

{ int x=6, y;
  printf("x=%d\n",x);    /*输出 x 的初值*/
  y=++x;                 /*前置运算: x 先增 1(=7), 然后再赋值给 y(=7)*/
  printf("y=++x: x=%d,y=%d\n",x,y);
  y=x--;                 /*后置运算: 先将 x 的值(=7)赋值给 y(=7), 然后 x 再减 1(=6)*/
  printf("y=x--: x=%d,y=%d\n",x,y);
}

```

解:

运行结果:

```

x=6
y=++x:x=7,y=7
y=x--:x=6,y=7

```

2.5 写出下面程序执行之后的输出结果。

```

main()
{
  float x,y;
  int k;
  x=3.5;
  k=(int)x+10;
  printf("x=%f,k=%d",x,k);
}

```

解:

运行结果:

```

x=3.500000,k=13

```

注意: 在 $k=(int)x+10$; 中 $(int)x$ 对 x 进行了强制类型转换, 使 $k=3+10=13$, 但是 x 本身的数据类型并没有改变。

2.6 写出下面程序执行后的输出结果。

```

main()
{
  int i=6,j,k;
  j=++i;
  k=i++;
  printf("i=%d,j=%d,k=%d",i,j,k);
}

```

解:

运行结果:

```

i=8,j=7,k=7

```

2.7 写出下面程序执行后的输出结果。

```
main()
{
    int i=5,j=10;
    i=i-j++;
    printf("i=%d,j=%d",i,j);
}
```

解:

运行结果:

i=-5,j=11

注意: ++和--运算符的结合方向是自右向左而且优先级比-运算符高, i=i-j++; 语句先进行 j++ 的运算, 然后在进行其他运算。

2.8 写出下面程序执行后的输出结果。

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

```
void main()
```

```
{
    /* 换行符'\n', 用于输出换行 */
    printf("How are you?\n");
    printf("I am fine.\n\n");
    /* 横向跳格符'\t', 使跳到下一个输出区 */
    printf("How are you?\t");
    printf("I am fine.\n\n");
    /* 退格符'\b', 使当前的输出位置退一格, 即输出的起始位置左移一位 */
    printf(" How are you?\n");
    printf(" \bI am fine.\n\n");
    /* 回车符'\r', 使当前输出位置回到本行开头 */
    printf("          I am fine.");
    printf("\rHow are you?\n\n");
    /* 多个转义字符的混合运用 */
    printf("note:\n  a s\ti\b\bk\rp\n");
}
```

解:

运行结果:

How are you?

I am fine.

How are you? I am fine.

How are you?

I am fine.

How are you? I am fine.

note:

p a s k i

2.9 写出下面程序执行后的输出结果。

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

```
void main()
```

```
{
```

```
/* 定义一个整数类型的变量，用来存放后面算式的值 */
```

```
int logic;
```

```
int a = 1;
```

```
int b = 2;
```

```
int c = 3;
```

```
logic = a+b>c&&b<=c;
```

```
printf("logic = %d\n", logic);
```

```
logic = a>=b+c||b==c;
```

```
printf("logic = %d\n", logic);
```

```
logic = !(a<c)+b!=1&&(a+c)/2;
```

```
printf("logic = %d\n", logic);
```

```
}
```

解:

运行结果:

```
logic = 0
```

```
logic = 0
```

```
logic = 1
```

2.10 写出下面程序执行后的输出结果。

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

```
void main()
```

```
{
```

```
int i, j, k;
```

```
int m, n, p;
```

```
i = 8;
```

```
j = 10;
```

```
k = 12;
```

```
/* 自增在操作数之前 */
```

```
m = ++i;
```

```
printf("i = %d\n", i);
```

```
printf("m = %d\n", m);
```

```

/* 自减在操作数之后 */
n = j--;
printf("j = %d\n", j);
printf("n = %d\n", n);
/* 自增、自减的混合运算 */
p = (++m)*(n++)+(--k);
printf("k = %d\n", k);
printf("p = %d\n", p);
}

```

解:

运行结果:

```

i = 9
m = 9
j = 9
n = 10
k = 11
p = 111

```

2.11 写出下面程序执行后的输出结果。

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

```
void main()
```

```

{
    /* 定义了一个无符号字符型变量, 此变量只能用来存储无符号数 */
    unsigned char result;
    int a, b, c, d;
    a = 2;
    b = 4;
    c = 6;
    d = 8;
    /* 对变量进行“按位与”操作 */
    result = a & c;
    printf("result = %d\n", result);
    /* 对变量进行“按位或”操作 */
    result = b | d;
    printf("result = %d\n", result);

    /* 对变量进行“按位异或”操作 */
    result = a ^ d;
    printf("result = %d\n", result);
    /* 对变量进行“取反”操作 */

```

```

    result = ~a;
    printf("result = %d\n", result);
}

```

解:

运行结果:

```

result = 2
result = 12
result = 10
result = 253

```

2.12 设计一个顺序结构程序, 求方程的根。

解:

```

#include "math.h"          /*为使用求平方根函数 sqrt(), 包含 math.h 头文件 */
main()
{
    float a,b,c,disc,x1,x2;
    printf("Input  a, b, c: ");
    scanf("%f,%f,%f",&a,&b,&c); /*输入方程的 3 个系数的值*/
    disc=b*b-4*a*c;           /*求判别式的值赋给 disc*/
    x1=(-b+sqrt(disc))/(2*a);
    x2=(-b-sqrt(disc))/(2*a);
    printf("\nx1=%6.2f\nx2=%6.2f\n",x1,x2);
}

```

运行结果:

```

Input  a, b, c:
1,7,5 ✓

```

```

x1= -0.81
x2= -6.19

```

2.13 设计一个顺序结构程序, 将小写字母转换成大写输出。

解:

```

#include "stdio.h"
main()
{
    char c1,c2;
    printf("Input a lowercase letter: ");
    c1=getchar();
    putchar(c1);printf("\n",c1);
    c2=c1-32;          /*将大写字母转换成对应的小写字母*/
    printf("%c,%d\n",c2,c2);
}

```

```
}
```

运行结果:

Input a lowercase letter:a ✓

a,97

A,65

2.14 下面程序段是说明 putchar()和 getchar()函数的格式和作用的, 试着使用并理解。

```
#include "stdio.h"      /*文件包含*/
```

```
main()
```

```
{
```

```
    char ch;
```

```
    printf("Please input two character: ");
```

```
    ch=getchar();          /*从键盘输入一个字符并赋给变量 ch */
```

```
    putchar(ch);putchar('\n');
```

```
    putchar(getchar());    /*从键盘输入一个字符并输出*/
```

```
    putchar('\n');
```

```
}
```

运行结果:

Please input two character: ac ✓

a

c