



普通高等教育“十二五”规划教材·公共课系列

C语言程序设计

辅导与实践

刘加海 陈忠宝 朱云芳 等◎编著



科学出版社

普通高等教育“十二五”规划教材·公共课系列

C 语言程序设计辅导与实践

刘加海 陈忠宝 朱云芳 等 编著

科学出版社

北 京

内 容 简 介

本书是《C 语言程序设计》(第二版)一书的配套教材,主要内容包括以下三部分。

第一部分为辅导与练习,与教材相对应地介绍了每章的重点,“本章要点”部分总结了学生学习过程中经常遇到的一些疑难和容易出错的问题,对读者的学习能起到指点迷津的作用。

第二部分为程序设计实验,以 Visual C++ 6.0 为编程环境,共给出 14 个实验,包含多文件系统的程序设计、输入/输出函数应用、循环程序设计,分支程序设计、数组的应用、函数的应用及文件操作等方面的内容,实验的安排符合人们学习程序设计的心理活动,实验的设计采用了“认识—模仿—思考—创造”的设计过程。

第三部分为自测试题选编,提供了 4 套试题,可供不同需求的学生结合自身水平进行测试。

本书可作为高等学校非计算机专业计算机程序设计课程和计算机专业程序设计课程的实验教学用书,也可作为自学 C 语言程序设计的实验参考书或培训用的实验教材。

图书在版编目(CIP)数据

C 语言程序设计辅导与实践/刘加海,陈忠宝,朱云芳等编著. —北京:科学出版社,2011

(普通高等教育“十二五”规划教材·公共课系列)

ISBN 978-7-03-031295-2

I. ①C… II. ①刘… ②陈… ③朱… 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(2011)第 102299 号

责任编辑:孙露露 / 责任校对:耿 耘

责任印制:吕春珉 / 封面设计:东方人华平面设计部

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>

北京路局票据印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2011 年 6 月第 一 版 开本:787×1092 1/16

2011 年 6 月第一次印刷 印张:13 1/4

印数:1—3 000 字数:302 000

定价:23.00 元

(如有印装质量问题,我社负责调换<路局票据>)

销售部电话 010-62142126 编辑部电话 010-62138978-8212

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

本书是《C 语言程序设计》(第二版)的配套教学用书,可用于各本科院校非计算机专业计算机程序设计课程和计算机专业程序设计课程的实验教学。

C 语言是目前使用最为广泛的计算机高级语言,很多高校的计算机或非计算机专业都是选用 C 语言作为学生学习程序设计的基础语言。对于大多数学生而言,初次接触计算机程序设计,他们还不了解计算机工作的原理和特点,因此要学好 C 语言并不是件容易的事情。有很多学生,即使能够理解课本里的每个知识点,但是真正要写程序的时候,仍然一筹莫展。帮助学生理解并掌握教材中每章的重点和难点,并提高他们实际动手编写程序的能力,是作者编写本书的初衷。

本书共分为三个部分,其内容安排大致如下:

第一部分为辅导与练习,在这一部分,总结教材中每章的重点和难点,并进行讲解,在讲解的过程中配以适当的例题,让学生能更好地理解问题,掌握该知识点。除此之外,在这部分的最后,还给出了一部分练习题和参考答案,学生可在课外自己进行练习。

第二部分为程序设计实验,本书介绍的实验例子主要在 Visual C++ 6.0 环境下运行,也适用于 Turbo C 等编译器。

与教材的内容相配合,本书共安排了 14 个实验,除第 3 章循环程序设计和第 5 章数组及应用各有两个实验之外,其余每章各安排一个实验。在每个实验中,给出了本次实验的基本任务、与实验内容相对应的基本知识点、实验的基本目的,指明了在该次实验中,学生需要掌握的内容;每个实验安排了 5~10 个小实验,其内容的编排按照由易到难的顺序,让学生先学习例程,从“读程序”做起,然后再开始自己模仿编写、调试程序。这样安排的目的,一方面要培养学生对程序的“语感”及编写程序的正确思路;另一方面,也希望学生在读程序的过程中,能学会并养成编写程序的良好风格及好习惯,这对他们日后编写大型程序是十分有帮助的。在这一部分中,我们还根据学生对本章的知识点理解容易出错的部分,提出了相应的问题思考,让学生自己动手通过上机实验解决问题,在这个过程中进一步加深理解并加强对课本上知识点的印象。此外,要求学生在每次实验中都记录实验过程,写出实验小结与体会。通过精心构造的实验内容及安排,让学生在提高自己动手编程能力的同时,进一步加深对课本上知识点的理解,与理论课相辅相成,取得更好的学习效果。实验可在 Windows 操作系统下 Turbo C 2.0、Turbo C 3.0、Visual C++6.0 集成环境或 Linux 环境中 gcc 编译。

在第三部分,提供了 4 套自测试题和答案,前两套选自浙江大学远程试题,后两套选自浙江大学本科试题,供学生进行自我练习。根据作者多年从事 C 语言课程的教学经验,这几套试题题量合适,难度适中,内容覆盖了全书所有章节,能够较准确地评测出学生对整个课程的掌握程度。



本书的编写人员有浙江大学城市学院刘加海、陈忠宝、严冰，浙江工商大学朱云芳，浙江商业职业技术学院孔美云，浙江大学宁波理工学院唐云廷。作者对 C 语言有丰富的实践与教学经验。

由于时间仓促及作者水平所限，书中难免存在疏漏和不足之处，敬请广大读者批评指正。

目 录

第一部分 辅导与练习	1
第 1 章 C 语言程序的基本构成	1
第 2 章 输入/输出程序设计	6
第 3 章 循环程序设计	10
第 4 章 分支结构的程序设计	18
第 5 章 数组及其应用	25
第 6 章 函数及其应用	31
第 7 章 变量存储类型与多文件系统	40
第 8 章 文件的操作及其应用	44
第 9 章 指针与数组	56
第 10 章 指针与函数	68
第 11 章 结构体及其应用	77
第 12 章 链表及其应用	88
第二部分 程序设计实验	103
实验 1 C 程序的构成	103
实验 2 输入/输出程序设计	109
实验 3 循环程序设计	113
实验 4 循环嵌套程序设计	116
实验 5 分支程序设计	119
实验 6 数组及其应用 (1)	126
实验 7 数组及其应用 (2)	129
实验 8 函数及其应用	135
实验 9 变量存储类型与多文件系统	143
实验 10 文件操作	149
实验 11 指针与数组的应用	153
实验 12 指针与数组、指针与函数的应用	156
实验 13 结构体变量及应用	161
实验 14 队列与堆栈的应用	170
第三部分 自测试题选编	178
试卷 1	178
试卷 2	182
试卷 3	186
试卷 4	193

第一部分 辅导与练习

第 1 章 C 语言程序的基本构成

本章要点

1. C 程序中的文件。用 C 语句所写的程序称为 C 源程序，它的后缀为.c。C 源程序经过编译（compile）后生成目标文件（.obj），链接库函数后生成可执行文件（.exe）。
2. 结构化程序的结构。结构化程序的 3 种结构：顺序结构、分支结构和循环结构。
(1) 顺序结构的程序是一条语句接一条语句顺序地往下执行，一直到程序结束。

例如：

```
int x,y;
scanf("%d",&x);
y=x*x;
printf("y=%d\n",y);
```

以上 4 条语句，顺序地往下执行。

- (2) 分支结构指在程序执行过程中，程序的流程可由多路分支组成，根据不同的条件去执行不同的任务。

例如：

```
int x;
if(x>0)
    printf("%d 的值大于 0\n",x);
else
    printf("%d 的值小于 0\n",x);
```

以上 if-else 语句根据 x 是否大于 0，只执行一条分支。

- (3) 循环结构是指在程序中需要根据某项条件重复地执行某项任务若干次，直到满足或不满足某条件为止。

例如：

```
int x=100,sum=0;
while(x>=1)
{
    sum=sum+x;
    x--;
}
```

3. 算法+数据结构=程序。算法是指解决问题所使用的一系列合乎逻辑的解题步骤，

程序可以认为是对算法的具体实现，即算法+数据结构=程序。

例如，有如图 1-1 所示的环，剪开一个环作为一种操作，做怎样的最小操作，能呈现出最多的状态？

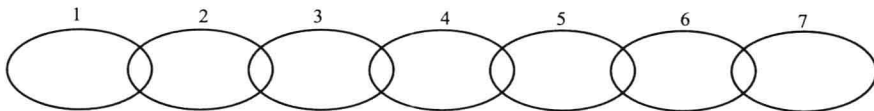


图 1-1 环

图中是 7 个环环相扣的环，如果把第 3 个环剪断，可呈现 7 种状态。

一个环：3

二个环：1, 2

三个环：1, 2, 3

四个环：4, 5, 6, 7

五个环：3, 4, 5, 6, 7

六个环：1, 2, 4, 5, 6, 7

七个环：1, 2, 3, 4, 5, 6

通过剪断第 3 个环（即最小操作），呈现最多状态，这是解决问题的方法，即算法。

例如，如何在数列 23,4,56,-9,100,2,45,0,87,10 中查找最大数？

具体的算法如下。

第 1 步：把第 1 个数 23 作为当前最大值；

第 2 步：当前最大值与第 2 个数比较，获取当前最大值 23；

第 3 步：当前最大值与第 3 个数比较，获取当前最大值 56；

第 4 步：当前最大值与第 4 个数比较，获取当前最大值 56；

第 5 步：当前最大值与第 5 个数比较，获取当前最大值 100；

.....

第 10 步：当前最大值与第 10 个数比较，获取当前最大值 100。

程序代码如下：

```
int a[10]={23,4,56,-9,100,2,45,0,87,10};
int i,max;
max=a[0];
for(i=1;i<10;i++)
{
    if(a[i]>max)
        max=a[i];
}
printf("max=%d\n",max);
```

4. 流程图。流程图是一种用图解方式来说明解决一个方案所需完成的一系列操作。

例如，查找最大数的流程图如图 1-2 所示。

5. C 语言是结构化的程序设计语言。C 程序由一个或多个文件构成，而文件可由一个或多个函数组成，函数由语句构成，“;”表示语句结束，语句由表达式构成，表达式由保留字、标识符、运算符构成。

(3) 常见的保留字。

保留字是指 C 语言中有特殊含义的字, 只能应用于 C 语言中的特定场合, 例如:

int: 用于定义整型变量。

float: 用于定义实型变量。

double: 用于定义双精度变量。

char: 用于定义字符变量或字符串变量。

if: 用于判断。

for: 用于循环。

while: 用于循环。

(4) 标识符。

标识符分为系统命名的标识符与用户自定义标识符。通常, 用户自定义标识符以字母或下划线开头, 由字母、数字和下划线组成。

6. 常量。

(1) 整型常量的表示方法有 023、0x23、23 等。

(2) 实型常量的表示方法有 2.71828、0.0、.54、0.3e2 等。

(3) 字符常量的表示方法有 'A'、'\n' 等。

注意: 转义字符常量, 如 '\n'、'\101'、'\141' 只代表一个字符。反斜线后的八进制数可以不用 0 开头, 如 '\101' 代表的就是字符常量 'A'。反斜线后的十六进制数只可由小写字母 x 开头, 不允许用大写字母 X, 也不能用 0x 开头, 如 'x41' 代表字符常量 'A'。

7. 变量。在程序设计中, 使用变量前都需要先定义, 定义变量意味着给变量分配存储空间, 变量的定义格式如下:

类型 变量名;

例如, 定义两个整型变量 x、y:

```
int x,y ;
```

在程序中变量的地址是用 "&变量名" 来表示的。

8. 字符串。字符串常量是由双引号括起来的一串字符, 如 "string" 就是字符串常量。在 C 语言中, 系统在每个字符串的最后自动加入一个字符 '\0' 作为字符串的结束标志。

(1) 字符串的结束符 '\0' 占内存空间, 但在测试字符串长度时不计在内, 也不输出。

(2) '\0' 为字符串的结束符, 但遇到 '\0' 不一定是字符串的结束, 可能是八进制数组成的转义字符常量, 如字符串 "abc\067de" 表示 6 个字符, 并非为 3 个, 因 '\067' 为一个转义字符。

9. 运算符。C 程序中有极丰富的运算符, 如算术运算符、关系运算符、逻辑运算符、位运算符等。

(1) 算术运算符有 +、-、*、/、%、++、-- 等。

注意: % 只能适用于整型数运算, 表达式 $y=(x++)$ 与 $y=x++$ 等同。

(2) 关系运算符主要用于比较, >、>=、<、<= 的优先级高于 ==、!=。

(3) 逻辑运算符有逻辑与 (&&)、逻辑或 (||) 和逻辑非 (!)。

(4) 位运算符有 ~、&、|、^、<<、>>。

10. 编辑 C 程序时应注意的问题如下:

- (1) C 程序中区分大小写字母, 书写格式自由, 一行内可写几个语句。
- (2) 用 “;” 作为语句的结束标记。
- (3) 函数定义如 `main()`、头文件如 `#include<stdio.h>`、宏定义如 `#define PI 3.14159` 都不是语句, 后面不能用 “;” 号。
- (4) 注释用符号 “/*” 和 “*/” 来界定, “/” 和 “*” 之间不可以有空格。
- (5) 可运行程序中必须有一个 `main` 函数, 且只能有一个 `main` 函数。
- (6) 一个好的程序应该有详细的注释。

课本习题参考答案

一、选择题

- | | | | | | |
|------|------|------|-------|-------|------|
| 1. A | 2. B | 3. A | 4. A | 5. D | 6. A |
| 7. A | 8. A | 9. B | 10. B | 11. D | |

二、程序调试题

1. 思考题答案:

- 1) 文件 `ex1-1.c` 由两个函数构成: `add` 和 `main`。
- 2) 程序调用了两个系统函数: `printf` 和 `scanf`。
- 3) 程序中用到得保留字为 `include`、`int` 和 `return`, 用到得用户自定义标识符为 `a`、`b`、`c` 和 `add`。

2. 思考题答案:

- 1) 程序的功能是从键盘读入两个整数, 调用函数 `and` 进行逻辑与运算, 调用函数 `or` 进行逻辑或运算。
- 2) 程序由 `ex1-2.c` 和 `ex1-2-2.c` 两个文件构成。
- 3) 程序调用了两个系统函数: `printf` 和 `scanf`。
- 4) 程序中用到得保留字为 `include`、`int` 和 `return`, 用到得用户自定义标识符为 `a`、`b`、`c`、`and` 和 `or`。

第 2 章

输入/输出程序设计

本章要点

1. 字符串的格式输入与输出。在计算机程序设计中最基本的就是输入与输出,输入与输出与变量的类型有关,如用 `printf`、`scanf` 的格式符 `%d`、`%lf`、`%c`、`%s` 控制整型变量、双精度变量、字符变量、字符串的输入与输出。

1) 字符串的输出。字符串的输出分直接输出和格式输出两种情况。

(1) 直接输出。

格式:

```
printf("输出字符串");
```

例如:

```
printf("*****");  
printf("请您输入");
```

(2) 格式输出。

格式:

```
printf("%s.....", 字符串变量地址, .....);
```

例如:

```
char str[20]= "abcd";  
printf("字符 str 的%s %s\n", "内容是", str);
```

2) 字符串的输入。

格式:

```
scanf("%s.....", 字符串变量地址, .....);
```

例如:

```
char str[20];  
scanf("%s", str);
```

注意: `scanf` 输入以空格为分隔。在上式中,如输入: 123 4567 abcd (回车), 字符串 `str` 的内容为 123, 数组名代表此数组的首地址。

2. 整型数的格式输入与输出。

1) 整型数的格式输出。

格式:

```
printf("%d.....", 整型变量或整型数, .....);
```

例如:

```
int x=0x20;  
printf("x=0%o, x=0x%x, x=%d y=%-5d\n", x, x, x, 200);
```

2) 整型数的格式输入。

格式:

```
scanf("%d.....", 整型变量地址, .....);
```

例如:

```
int x, y;  
scanf("%d %d", &x, &y);
```

此时要注意输入的两个整数用空格分隔, 如果语句写成

```
scanf("%d , %d", &x, &y);
```

则输入的两个整数用“,”分隔。

3. 双精度数的格式输入与输出。

1) 双精度数的格式输出。

格式:

```
printf("%lf.....", 双精度变量或双精度数, .....);
```

例如:

```
double y=0.51347e-3;  
printf("%-6.3lf\n", y);
```

2) 双精度数的格式输入。

格式:

```
scanf("%lf.....", 双精度变量地址, .....);
```

例如:

```
double y;  
scanf("%lf", &y);
```

注意: 在双精度数输入时不能控制精度, 例如, 下列语句是错误的:

```
double y;  
scanf("%6.2lf", &y);
```

4. 字符的格式输入与输出。

1) 字符的格式输出。

格式:

```
printf("%c.....", 字符变量或字符, .....);
```

2) 字符的格式输入。

格式:

```
scanf("%c.....", 字符变量地址, .....);
```

例如:

```
char ch1, ch2;  
scanf("%c%c", &ch1, &ch2);  
printf("%c %c %d\n", ch1, ch2, ch1-ch2);
```

注意:

- (1) printf 可以有修饰符, 如%6.2f、%-5d、%2c 等;
- (2) scanf 输入变量时, 输入格式必须与控制格式一一对应;
- (3) scanf 输入时%f 或%lf 不可以有精度控制, 在%s 时输入以空格分割;
- (4) 字符串的输入可以使用函数 gets, 字符串的输出可以使用函数 puts。

例如:

```
char str[100];
gets(str);
puts(str);
```

课本习题参考答案

1. 用函数 printf 输出一个中空三角星。

参考答案:

```
#include<stdio.h>
void main()
{
    printf("      *      ");
    printf("    *  *      ");
    printf("  *  *  *      ");
}
```

2. 从键盘输入两个整型数, 实现两个整型数的互换。

参考答案:

```
#include<stdio.h>
void main()
{
    printf("Enter two integer:\n");
    int x1,x2,x;
    scanf("%d%d",&x1,&x2);
    x=x1;
    x1=x2;
    x2=x;
    printf("%d,%d",x1,x2);
}
```

3. 从键盘输入一个三位的整数, 分别输出它的个位数、十位数和百位数。

参考答案:

```
#include<stdio.h>
void main()
{
    printf("Enter one integer:\n");
    int x,y1,y2,y3;
    scanf("%d",&x);
    y1=x%10;
    x=x/10;
    y2=x%10;
    x=x/10;
    y3=x%10;
    printf("个位: %d,十位: %d,百位: %d\n",y1,y2,y3);
}
```

4. 写出以下程序的输出结果。

```
#include<stdio.h>
void main()
{
    int i=010,j=10;
```

```
printf("%d,%d\n",++i,j-- );  
}
```

答案: 9,10

5. 编写程序, 屏幕提示“您是一个学生吗”, 可输入'y'或'n'进行回答, 屏幕可输出您的选项。

参考答案:

```
#include<stdio.h>  
void main()  
{  
    char x;  
    printf("您是一个学生吗?您输入'y'或'n'进行回答:\n");  
    scanf("%c",&x);  
    if(x=='y')  
        printf("您是一个学生\n");  
    else  
        printf("您不是一个学生\n");  
}
```

6. 从键盘输入一个角度 x , 求 $10*\sin(x)$ 的值。

```
#include<stdio.h>  
#include<math.h>  
void main()  
{  
    printf("Enter one integer:\n");  
    float x;  
    scanf("%f",&x);  
    printf("10*sin(x)=%f\n",10*sin(x));  
}
```

第 3 章

循环程序设计

本章要点

1. while 循环。while 循环的格式如下：

```
while(条件表达式)
{
    循环体
}
```

while 循环是前判断循环。首先，判断循环的条件是否满足，如条件满足则执行循环体，如条件不满足则退出循环体，这种循环有可能循环一次也不执行。

例如：

```
while((ch=getchar())!='#')
{
    if(ch>='A'&&ch<='Z')
        putchar(ch);
}
```

注意：循环体不止一条语句时要形成语句块，在(ch=getchar())中，括号不能省略。

2. do-while 循环。do-while 循环是后判断循环，无论循环的条件是否满足，循环体至少应执行一次，do-while 循环的语句格式如下：

```
do
{
    循环体
}while(表达式);
```

例如：

```
int sum=0,x;
do
{
    scanf("%d",&x);
    sum+=x;
}while(x);
```

while 与 do-while 在循环条件不满足时是有区别的，如语句段：

```
int n;
scanf("%d",&n);
while(n)
{
    printf("*");
}
```

```
n--;  
}
```

语句段:

```
int n;  
scanf("%d",&n);  
do  
{  
    printf("*");  
    n--;  
}while(n);
```

do-while 在条件不满足时还要执行一次,如上例中 n 输入 0 时, do_while 还要执行循环体。

3. for 循环。C 语言中 for 循环语句应用比较灵活,虽然主要用于定数循环,但也可以不定数循环,它完全可以替代 while 循环。

for 语句格式如下:

```
for(初始化表达式;条件表达式;增量表达式)  
{  
    循环体  
}
```

例如:

```
for( ;(ch=getchar())!='#' ; )  
    if(ch>='A'&&ch<='Z')  
        putchar(ch);
```

4. break 语句和 continue 语句。break 语句可应用于循环体或 switch 语句中。在循环中,当程序执行到 break 语句后,立即退出循环;在分分支语句中,程序遇到 break 语句时,停止以下分支的 执行。

continue 语句只能位于循环体内,执行 continue 语句后,continue 语句到循环体末尾之间的语句会被跳过,转到循环条件判断处,根据循环条件决定是否执行下一次循环。break 语句和 continue 语句往往与 if 一起使用。

如语句段:

```
int n;  
scanf("%d",&n);  
while(n)  
{  
    if(n%2==0)  
        continue;  
    printf("*");  
    n--;  
}
```

课本习题参考答案

1. 执行语句 for($i=0;i++<5$;)后,变量 i 的值是 ()。

A. 6

B. 5

C. 4

D. 0

答案: A