全国高职高专规划教材

C语言程序设计

Programming C

刘加海 主 编
齐幼菊 周文革 副主编

全国高职高专规划教材

科学出版社
www.sciencep.com

全国高职高专规划教材

C 语言程序设计

刘加海　主编

齐幼菊　周文革　副主编

科学出版社

北　京

内 容 简 介

本书根据C语言的特点，力求突出系统性、完整性、实用性，由浅入深地讲授C语言的基本概念及编程特点，全书贯穿C语言的精华部分——指针、数组、函数，使教材的整个体系融为一体。

本书共7章，讲述了C程序设计基础、程序的控制结构、函数、指针与数组、指针与函数、结构体与共用体、文件等内容。

本书内容精炼，结构合理，对学习C语言中可能遇到的难点作了系统、详尽的分析，极大地减轻了读者的困难。本书不仅适合作为高职、高专计算机专业的教材，也是各类本科、专科院校学生学习C语言程序设计的良师益友，是一本能提高学生程序设计能力的优秀教材。

图书在版编目(CIP)数据

C语言程序设计/刘加海主编；齐幼菊，周文革副主编. —北京：科学出版社，2003

（全国高职高专规划教材）

ISBN 7-03-011988-6

Ⅰ.C… Ⅱ.①刘…②齐…③周… Ⅲ.C语言—程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第065815号

策划编辑：李振格/责任编辑：于 娜

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

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003年8月第一版 开本：787×1092 1/16

2003年8月第一次印刷 印张：13 1/2

印数：1—5 000 字数：304 000

定价：19.00元

（如有印装质量问题，我社负责调换〈环伟〉）

全国高职高专规划教材编委会名单

主　任　俞瑞钊

副主任　陈庆章　蒋联海　周必水　刘加海

委　员　（以姓氏笔画为序）

王　雷　王筱慧　方　程　方锦明　卢菊洪　代绍庆

吕何新　朱　炜　刘向荣　江爱民　江锦祥　孙光弟

李天真　李永平　李良财　李明钧　李益明　余根墀

汪志达　沈凤池　沈安衢　张　元　张学辉　张锦祥

张德发　陈月波　陈晓燕　邵应珍　范剑波　欧阳江林

周国民　周建阳　赵小明　胡海影　秦学礼　徐文杰

凌　彦　曹哲新　戚海燕　龚祥国　章剑林　蒋黎红

董方武　鲁俊生　谢　川　谢晓飞　楼　丰　楼程伟

鞠洪尧

秘书长　熊盛新

本书编写人员名单

主　编　刘加海

副主编　齐幼菊　周文革

撰稿人　朱光忠　期　进　戚海燕　徐寿芳　郑正建　张　翔

前　言

C 语言是一种优秀的结构化程序设计语言，它结构严谨、数据类型完整、语句简练灵活、运算符丰富。自从应用 C 语言编写 UNIX 操作系统获得成功以来，它一直受到计算机业内人士的好评。即使在面向对象的语言广泛使用的今天，C 语言仍然是高校计算机专业的一门专业基础课。

几乎每一所高校都开设 C 语言这门课，然而真正易教易学的教材却不多，这是笔者在浙江大学教授近 100 轮次的 C 语言教学课后的深刻体会。目前大多数 C 语言教材先讲授数组，在课程快结束时才讲指针。很多学生对指针的概念感到难以理解，对于它的灵话应用更是困难。从 1996 年开始，笔者根据多年的教学经验，尝试在第一章就讲授指针与数组，把数组和指针有机地结合起来，并且全书贯穿指针与数组的概念，使学生一开始就对 C 语言有总体上的清晰认识，这种方法在后来的教学中取得了很大的成功，从心理学的角度来看这也是完全符合人类的心理学习规律。近来笔者对这种教学方法进行了反复的研究，并进行多次改进和进一步完善，结合高职高专的教学特点，编著了这本《C 语言程序设计》。

我们不敢说哪种方法最好，但必须尝试一种教学方法，它不但要符合此课程的教学特点，使学生在学习过程中感到轻松，更要符合人们学习的心理学习规律，花更少的时间，能更好地掌握知识，希望这本书能让广大师生达到此目的。

尽管笔者修改再三，但无论书的结构还是书的内容都难免有不足的地方，真切地希望广大师生批评指正，以便在一下版中订正。

本书由刘加海担任主编，齐幼菊、周文革任副主编，参加本书编写的还有戚海燕、朱光忠、张翔、期进、郑正建、徐寿芳，在这里对他们的辛苦工作表示感谢。

编　者

2003 年 6 月

目　录

第 1 章　C 程序设计基础

本章重点

- C 语言程序的结构
- 变量与常量的表示方法
- 运算符与表达式
- 指针的概念及变量地址与指针的关系
- 数组的概念
- 指针对数组元素的引用方法

本章难点

- C 语言中常量的表示方法
- 整型数在计算机中存储的形式
- 不同类型数据的转换
- 转义字符
- 各种运算符与表达式的正确理解
- 指针的概念及指针的移动
- 数组的概念及数组的赋值

1.1　C 程序的结构

1.1.1　程序的概念

程序是指人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中。当人们给出命令之后，它就按指令操作顺序自动进行，这种可以连续执行的一条条指令的集合称为“程序”。目前，正在使用的计算机程序设计语言有上百种，有些语言是面向机器的，如二进制语言，而多数是面向问题的语言。面向问题的语言都被称为计算机的“高级语言”，如 C 语言、Pascal 语言、FoxBase 等；Visual C++、Visual Basic 等被称为面向对象的语言。这些语言都是用接近人们习惯的自然语言和数学语言作为语言的表达形式，人们学习和操作起来感到十分方便。

由高级语言编写的程序称为“源程序”，把由二进制代码表示的程序称为“目标程序”。如何把源程序转换成机器能够接受的目标程序，软件工作者编制了一系列软件，通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制的机器指令。这种具

有翻译功能的软件称为“编译程序”。每一种高级语言都有与它对应的编译程序。目前的程序设计一般可分为非结构化程序设计、结构化程序设计和面向对象的程序设计。C 语言是结构化程序设计。程序设计的过程一般包括如下方面。

（1）问题的提出、要求及所采用的数据结构。

（2）算法的确定、程序的编制。

（3）程序的调试及修改。

（4）整理并写出文档资料。

结构化程序设计由三种结构组成：顺序结构、选择结构、循环结构。

我们所写的 C 语句序列称为 C 源程序，它的后缀为.c，C 源程序经过编译（compile）后生成一个后缀为.obj 的二进制文件，最后由连接程序（link）把此.obj 文件与 C 语言提供的各种库函数连接起来生成一个.exe 文件，它就是可执行文件。

因而程序的设计过程如图 1.1 所示。

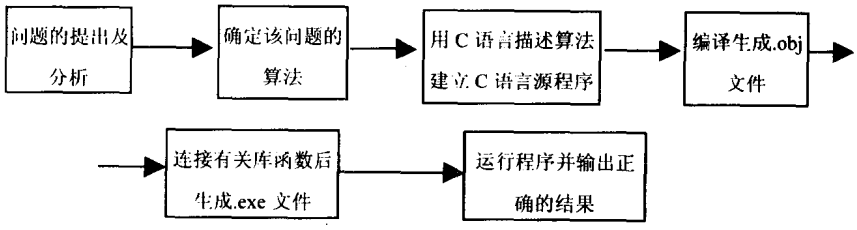

图 1.1 程序的设计过程

1.1.2 C 程序的构成

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组成，而一个文件可由一个或多个函数组成。C 程序中必须有一个函数名为 main 的函数，且只能有一个 main 函数。程序运行时从 main 函数开始，最后回到 main 函数。

C 语言源程序可由一个或一个以上文件组成，而每个文件至少有一个函数，函数是 C 语言的最基本的单位。

C 函数由语句构成，语句结束符用“;”表示，语句由关键字、标识符、运算符和表达式构成。其中“{”和“}”分别表示函数执行的起点与终点或程序块的起点与终点。

C 程序中书写格式自由，一行内可写几个语句，但区分大小写字母。用 C 语言写成的函数结构图如 1.2 所示。

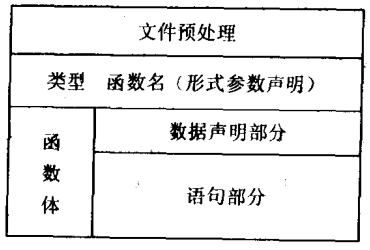

图 1.2 C 语言的结构示意图

例 1.1　函数结构的例子。

```
#include<stdio.h>                    /*文件预处理*/
int add(int x,int y)                 /*函数名及形式参数的声明*/
{                                    /*函数体开始*/
  int z;                             /* 数据定义部分，给变量分配内存空间*/
  z=x+y;                             /*执行语句*/
  return z;                          /*函数结束前返回一个整型值*/
}                                    /*函数体结束*/
```

在程序中可以对程序进行注释，注释的目的主要是为了人们能读懂您的程序。注释部分必须用符号“/*”和“*/”来界定，“/”和“*”之间不可以有空格。注释可以用西文，也可以用中文。注释可以出现在程序中任意合适的地方，注释部分对程序的运行不起作用。在注释中可以说明变量的含义、程序段的功能，以便帮助人们阅读程序。因此一个好的程序应该有详细的注释。

在上例中，如要在机器上运行必须要有一个 main()函数，通过 main()调用函数 add()，例如完整的程序代码为：

```
#include<stdio.h>
int add(int x,int y)
{
  int z;
  z=x+y;
  return z;
}
void main( )
{
  int a=1,b=2,z;                     /* 变量的定义*/
  z=add(a,b);                        /* 函数调用 */
  printf("%d+%d=%d\n",a,b,z);        /* %d为格式控制符，表示以整数形式输出结果
*/
}
```

此程序的执行结果为：1+2=3。

请读者仔细思考，有关问题将在以后学到。

良好的编程风格有助于对程序的阅读、理解和记忆，因而编程时应养成良好的习惯，并注意以下各点。

（1）用“;”作为语句的结束记号，但 main（）、#include< >、#define 不是语句，后面不能用“;”号。例如：

```
int i;
for(i=1;i<=5;i++)                   /*此位置无";"号*/
printf("Hello\n");
```

（2）“/*”和“*/”及“{”和“}”必须成对使用。

（3）程序中代码区分大小写。

（4）要有预处理，如#include<stdio.h>、#include<math.h>、#include<graphics.h>等。

（5）程序中必须有一个 main（）函数，且只能有一个 main（）函数。

1.2 最简单的 C 程序

C 语言程序可由表达式、系统函数及自定义函数组成，程序可以很复杂，也可以很简单，但即使很简单的程序也应由 main()及“{”、“}”组成，例如：

例 1.2 最简单的 C 程序。

```
/*最简单的C程序*/
#include<stdio.h>
void main( )
{
    ; }
```

此程序没有什么意义，执行后什么也没做，可以认为是最简单的 C 程序了。

例 1.3 字符串的输出。

```
/*功能  打印字符，在屏幕上输出:Hello，world */
#include<stdio.h>
void main()
{
printf("Hello,world\n");          /*\n为换行符*/
}
```

此程序执行后在屏幕上输出：

```
Hello，world
```

在 C 语言中，printf 为格式输出函数，它是 C 语言的库函数。包含在 stdio.h 库中，因而凡是用到 printf 函数都要包含函数库 stdio.h。在此函数中，除了格式控制符外，其他字符原样输出，因而此程序的运行结果为：Hello，world。下面我们将用一些例子来说明此函数的应用。

例 1.4 整型数输出的例子。

```
/*功能 输出整型数 */
#include<stdio.h>
void main()
{
  int  x=2;                       /* 定义一个整型x，并赋予初值2  */
  printf("%d  %d\n",x,5);         /* printf为输出函数，%d为整型数的输出格式  */
 }
```

此程序执行后在屏幕上输出：

```
2   5
```

在此例子中，语句 int x=2; 表示定义一个整型 x，并赋予初值 2。为什么在程序设计中，我们要定义变量后才能使用呢？这是因为变量的数值存放在内存空间，因而必须给变量分配存储空间，而变量的定义就是给变量分配存储空间。

在程序设计中，经常要通过键盘输入，显示器输出。在 C 语言中用于格式输入、输出函数为 scanf、printf ，它们常用格式如下。

%d　用于十进制的输入、输出。

%f　用于实型数的输入、输出。

%lf　用于双精度数的输入、输出。

%c　用于字符的输入、输出。

%s　用于字符串的输入、输出。

更详细的一些细节见附录。

例 1.5　从键盘输入一个双精度数，然后输出。

```
/*功能 从键盘输入 一个双精度数，然后输出 */
#include<stdio.h>
void main( )
{
  double  x ;
  scanf("%lf",&x);                    /* &x 表示变量x的地址  */
  printf("%lf\n",x);
 }
```

程序在运行时会出现一个空屏，此时您应输入一个数据，然后按回车键后在屏幕上输出结果。

例 1.6　从键盘输入一个字符串，然后输出的程序。

解析　在程序中先定义一个字符数组，然后提示输入，输入一个字符串后，最后输出在屏幕上。

```
/*功能：从键盘输入 一个字符串，然后输出字符串的程序 */
#include<stdio.h>
void main()
{
  char  name[10] ;              /* name表示存放字符串的首地址．可存放10个字符*/
  printf ("请输入您的姓名\n") ;/* 屏幕输出 */
  scanf ("%s",name) ;          /* 键盘读入字符串，以空格分隔，格式符为%s */
  printf ("您的姓名是：%s\n",name) ;
                               /*表示输出以name为首地址上的内容，直到字符串结束*/
}
```

程序运行结果如图 1.3 所示。

图 1.3　例 1.6 的程序运行结果

注意 当用格式符%s 时，对应的参数为字符变量的地址，表示输出以此地址上的内容开始，直到字符串结束。

1.3 标识符与保留字

在 C 语言中，标识符可用作变量名、符号名、函数名、数组名、文件名以及一些具有专门含义的名字。合法的标识符由字母、数字和下划线组成，并且第一个字符必须为字母或下划线，不能跨行书写，自定义标识符不能与关键字同名。下面的标识符都是合法的：

are、PI、liu_123、liu、e_array;

以下都是非法的标识符：

456P、cade-y、w.w、a&b

C 语言的标识符可以分为 3 类：保留字、预定义标识符、用户自定义标识符。

1.3.1 保留字

保留字也叫关键字，在C语言中用来表示某种特点含义的英文单词，如 include、int、char、for 等，关键字含义特定，不允许随意书写或拼写错误 。它们不能再用作变量名或函数名。

例如，C 语言中的关键字如下。

auto、break、case、char、const、continue、default、do、double、else、enum、extern、float、for、goto、if、int、long、register、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while。

1.3.2 标识符

1. 预定义标识符

有些标识符在C 语言中都有特定的含义，如C 语言提供的库函数的名字（printf）和预编译处理命令（如 define）等。C 语言语法允许用户把这类标识符另作它用，但这将使这些标识符失去系统规定的原意。为了避免误解，建议用户不要把这些预定义标识符另作它用。

2. 用户标识符

由用户根据需要定义的标识符称为用户标识符。一般用来给变量、函数、数组或文件等命名。程序中使用的用户标识符除要遵循命名规则外，还应注意做到“见名知义”，即采用具有相关含义的英文单词或汉语拼音，以增加程序的可读性。

如果用户标识符与关键字相同，程序在编译时将给出出错信息；如果与预定义标识符相同，系统并不报错，只是该预定义标识符将失去原定含义，代之以用户确认的含义；或者会引发一些运行时错误。

例 1.7 下列各组字符序列中，可用作 C 语言程序标识符的一组字符序列是（ ）。

（A）S.b，sum，average，_above　　（B）Class，day，lotus_1，2day

（C）#md，&12x，month，student_n1　　（D）D56，r_1_2，name，_st_1

解析　C 语言规定标识符只能由字母、数字和下划线三种字符组成，而且第一个字符必须是字母或下划线。因而答案 D 是正确的。

1.4　常量

在程序运行过程中，其值不能被改变的量，称为常量。在 C 语言中，常量有不同的类型，有整型常量（int），实型常量（float 或 double），字符常量（char）和字符串常量。即使是整型常量也还有短整型（short int）、长整型（long int）和无符号型（unsigned int）等。

1.4.1　整型常量

整型常量的定义格式：

int　const 常量名=常量值；或 const　int　常量名=常量值；

例如，int　const　x=10；

基本整型常量只用数字表示，不能带小数点，如 12、-1、0 等都是合法的整型常量。应该注意常量的值一旦指定，在程序中就不能再改变，在上例程序中如果您试图改变 x 的值，编译器就会报错。

整型常量通常可分为十进制、八进制和十六进制常量。

十进制常量用一串连续的数字来表示，如 32767、-32768、0 等。

八进制常量用数字 0 开头。如：　05、012、01 都是八进制数，它们分别代表十进制数 5、10、1。

十六进制常量用数字 0 和字母 x（或大写字母 X）开头。如：0x10、0xff、0X8 均是十六进制数，它们分别代表十进制数的 16、255、8。

注意　（1）十六进制数只能用合法的十六进制数字表示，字母 a、b、c、d、e、f 既可用大写字母也可用小写字母。

（2）常量的长度及表示数据的范围通常与机器类型有关。

例 1.8　在 C 语言中，错误的 int 类型的常数是（　）。

（A）32　　（B）078　　（C）037　　（D）0xAF

解析　选项 A 为 int 型数，选项 B 应该为八进制数，而八进制数中无 8，选项 C 为八进制，选项 D 为是十六进制，所以答案为 B。

1.4.2　实型常量

实型常量定义：**float　const 常量名=常量值；**

例如，float const　x=2.1；

实型常量也称数值型常量，它们有正值和负值的区分。实型常量通常用带小数点的数字表示，例如，3.14159、2.71828、0.0、.54 等。