郭陵之　编著

C语言基础教程

云南大学出版社

C 语言基础教程

郭陵之　编著

云南大学出版社

滇新登字 07 号

内容提要

本书是学习C语言程序设计的基础教材，全面系统、深入浅出地描述了C语言的基本知识及其程序设计方法。内容安排循序渐进，讲解方法通俗易懂，并辅以大量便于讲清概念的例题，使初学者容易理解和掌握，有利于培养实际编程能力。每章后面均附有习题，以帮助读者加深理解，巩固所学过的知识。

本书共分八章，主要包括：C语言的基本概念（第一章）；数据及其类型（第二章）；语句和控制流（第三章）；函数和变量的存储类（第四章）；数组（第五章）；指针（第六章）；结构和联合（第七章）；文件（第八章）。

本书可作为大专院校程序设计课程的教科书，也可作为计算机培训班的教材，并适合作为科技人员的自学材料和参考书。本书讲授50学时，配合25小时的上机练习，就能很好掌握全部内容。

书　　名：C语言基础教程
编　　著：郭陵之
责任编辑：周永坤
封面设计：丁群亚
出 版 者：云南大学出版社
地　　址：云南大学校内
邮政编码：650091
排 印 者：云南大学出版社微机室
　　云南大学出版社印刷厂
发 行 者：云南大学出版社
版本记录：787×1092　16开本　12.75印张　342.6千字
　　1995年9月第一版　1995年9月第一次印刷
印　　数：0001—2000册
书　　号：7-81025-586-X/TP·16
定　　价：12.50元

前　言

随着计算机科学知识的普及，越来越多的人认识、了解了C语言。C语言是一种结构化的计算机程序设计语言。它既具有高级语言的特点，又具有低级语言的功能；它既适合于编写应用程序，又适合于编写系统程序。它的应用领域十分广阔。编程人员一旦接触了C语言，并积累了一定的编程经验之后，就会对它爱不释手。其原因在于C语言是不折不扣的程序员语言，它允许程序员随心所欲地编程而几乎不受任何限制。

作者在美国留学期间，亲眼目睹了C语言在美国的广泛应用和诱人的发展前景。回国后即在本系率先开出《C语言程序设计》这门新课。这本教材就是作者在C语言教学活动中，精心备课、反复实践、不断发现新问题、逐步完善讲稿的基础上修订而成的。作者尝试着从初学者的角度来探讨C语言，形成了本书概念清楚、例题丰富、深入浅出、通俗易懂的特点。本书作为C语言的入门教材，适用于理工科大专学生和有一定计算机基础的科技工作者。对于普及C语言的各类培训班，本书也是一本很实用的教材。通过大约50学时的讲授和25小时的上机练习，就能很好地掌握本书的全部内容。

当前社会上学习C语言十分热门。有关C语言的各类书籍越出越"高级"。初学者和时间有限的科技人员，也许会在那些大部头的"经典"面前望而却步。而本书对于那些想在较短时间内了解C语言的全貌，进而学习利用C语言编程以解决实际问题的人来说，无疑是一本好的入门书。C语言由于其自身的特点，可写性强而可读性差，自学起来有一定的困难。因此选择一本内容适中、便于自学、又有实用价值的入门教材是很有必要的。本书作为学习C语言的入门钥匙，是当之无愧的。掌握了该书的基本内容后，将为读者进一步深入学习C语言提供坚实的基础和必备的条件。

本书得以出版，首先要感谢云南大学出版社的各位领导和同志们，感谢作者所在的云南大学地球科学系的领导和计算机重点学科的领导鼎力相助。计算机科学系的周堂福副教授审阅了本书的全稿，并提出了十分中肯而具体的修改意见；地球科学系的刘英实验师将本书除程序以外的全部内容录入计算机，付出了艰辛的努力；另外还有许多同志给予了作者极大的支持。所有这一切促成了本书的完成和出版。在此谨表示作者最诚挚的敬意和最深切的谢意。

由于本人水平有限，加之时间仓促，书成稿后仍留下不少的遗憾，缺点、错误在所难免。恳切地欢迎广大读者不吝指教。

编者

一九九四年十二月五日

目　录

第一章 概 述

§1.1 C语言的发展简况

一、C语言的由来

C语言是目前最流行的国际通用的计算机程序设计语言。

C语言是七十年代初由美国电话电报公司(AT&T)所属的贝尔实验室(BELL Lab.)的D·里奇(D·M·Ritchie)设计的,于1978年正式发表。人们称D·里奇为"C语言之父"。C语言的形成经历了以下几个阶段:

ALGOL 60　(1960年国际委员会设计)

↓

CPL　(Combined Programming Language 的缩写,1963年伦敦大学设计)

↓

BCPL　(Basic CPL,1967年剑桥大学的 Martin Richards 设计)

↓

B　(1970年 BELL Lab. 的 Ken Thompson 设计)

↓

C　(1972年 BELL Lab. 的 D·M·Ritchie 设计)

ALGOL 60是一种面向问题的语言,结构严密、注重语法,对后来的程序设计语言有很大的影响,但与计算机硬件相距太远,不适宜写软件特别是系统软件。

CPL是仿照ALGOL 60设计的,由于过于庞大和复杂,影响了它的广泛应用。

BCPL对CPL进行了精简,保持了基本特点,目前仍有顽强的生命力。

B对BCPL进一步简化,非常接近计算机硬件。

BCPL和B在简化方面走得太远,限制了语言能力,使它们只能用于特殊用途而不能解决一般问题。D·Ritchie把一些缺少的功能加入B,并作了全盘规划和整理,提出了C。

二、C语言与UNIX操作系统

C语言与UNIX操作系统有着非常密切的关系。当初D·Ritchie设计C语言就是为了更好地开发UNIX操作系统。可以说,UNIX系统是C语言发展的摇篮,而C语言又为UNIX的进一步发展铺平了道路。

1983年,UNIX系统和C语言的主要开发者K·Thompson和D·Ritchie获得了计算机科学技术领域的最高荣誉——图灵奖。这一方面充分肯定了两人的巨大贡献,同时也足以说明UNIX系统和C语言在计算机发展史上的重大作用和地位。

现在UNIX系统已经成为多用户微机、小型机、中型机,甚至某些大型机的标准操作系统,而C语言也已独立于UNIX系统,成为所有计算机上通用的高级语言。

三、标准C、ANSI C、C++

标准C是指由Brian W·Kernighan和Dennis M·Ritchie(简称K&R)在1978年所著的《The

C Programming Language》一书的基础上确立的C语言版本。

1983年美国国家标准局(简称ANSI)语言标准化委员会对C语言问世以来的若干发展、补充和修改作了总结，公布了一个C语言国际标准草案，称为83 ANSI C。1987年正式实施这个标准，称为ANSI C。ANSI C推出后，就把原来的C语言版本称为标准C。

ANSI C推出后，各C语言的编译系统都向它靠拢。原来K&R的标准C也就不再标准了，不过，习惯上仍称为标准C。

1988年，《The C Programming Language》一书出版十周年之际，K&R再次修改了他们的著作，完全按照ANSI C的标准来介绍和编程了。

我们这本教程，就是以ANSI C为基础，同时兼顾了各种版本的通用性和一致性来介绍C语言的。

读者翻阅C语言的参考书时，可能会发现有《C++程序设计语言》这一类的书，于是会产生这样的疑问"C++是什么东西?","C++是C语言吗?"。

C++是一种在C语言基础上发展起来的，比C语言功能更强、更高级的计算机语言。我们掌握了C语言，要进一步学习C++语言并不困难。事实上，所有用C语言编写的程序，在C++的环境中都可以通行无阻。不过，C++语言不是我们这本教程要讨论的内容。

§1.2 C语言的特点

C语言是一种通用程序设计语言，它具有丰富的表达式、现代控制结构与数据结构，它既具有高级语言的优点，又兼有低级语言的特点。

一、优点

C语言的优点和特点，归纳起来主要有如下几点：

1. 表达能力强

主要体现在它可以直接处理地址、字符和数字，可以完成很低级的机器级的算术、逻辑运算，可用于编写操作系统及有关的系统软件与支撑软件。在这方面，它的功能相当于汇编语言。

众所周知，早期进行系统程序设计均使用汇编语言。这主要是由于汇编语言能够体现计算机硬件指令级的特征，其表达能力足以描述系统软件各方面的特性，且由于汇编语言程序形成的代码有较高的质量，使得大系统运行的资源开销为计算机所能承受。但是汇编语言存在着它本身不能克服的缺点，它的可读性、可移植性以及描述问题的性能远不如目前已广泛使用的高级语言。用C语言取代汇编语言，既有高级语言的特点，又有低级语言的功能，可移植性好，使单纯用于编写系统软件的系统语言发展成为又可实用于计算领域和应用领域的多功能计算机语言。

2. 数据类型与控制结构丰富

C既提供了字符、整数、浮点数等基本纯量类型，也具有诸如数组、结构、联合等构造类型以及指针类型。变量、函数可具有多种存储类(如自动类、寄存器类、外部类、静态类)，有助于实施数据隐蔽和模块化程序设计。

C具有丰富的顺序、选择和循环控制结构。break语句和continue语句的使用可以减少程序中goto语句的使用，降低程序的复杂性。

3. 简洁、短小、规模适中，编译程序小得可在内存很小的机器上运行

例如，UNIX操作系统用C语言改写后，除核心部分的1000行左右(不到10%)仍用汇编语言

书写外，其余 90%以上都是用 C 语言描述的。新的操作系统的规模，只是原来全部用汇编语言编写的操作系统规模的三分之一。

C 语言的输入输出功能是通过调用标准函数库中的函数来实现的。库函数属于 C 语言的环境，而不是 C 语言本身的一部分，这样做也减小了 C 语言的规模。

4. 目标代码质量高

这是指时空质量，通俗地说，就是执行速度快，占用内存少。C 语言的各种结构（如指针、数组）力图反映机器指令与编码的结构，编译程序不用做太多的工作就可以生成时空效率高的代码。据经验统计，C 的代码效率只比汇编语言低 10～20%，这是一般高级语言无法比拟的。

5. 灵活性好

C 将大部分决策留给程序员去做，对诸如类型转换、类型兼容性检查等工作限制很少，对数组下标越界也不作检查。书写格式自由，这对于学过 FORTRAN 又来学 C 的人来说，感受尤深。

6. 可移植性好

这是 C 语言的一个突出优点。大家都知道，汇编语言由于过份依赖机器而很难移植。其它一些高级语言如 Pascal 和 FORTRAN，现有各种机器上的编译程序基本上都是按照标准重新实现的，很少有哪一个是移植其它机器上的编译程序的。而目前许多机器上配置的 C 都是移植得到的。统计资料表明，各种机器上的 C 编译程序中有 80%的代码是公共的。

我们这本书上所列举的程序，都是在 PC 机上 Turbo C 环境下实现的。由于 C 的可移植性，这些程序几乎不加改动就可以在 VAX 机的 VMS 系统下实现。

需要特别说明的是，虽然所有程序都是在机器上运行通过后直接拷贝过来的，但由于排版上的原因，现在程序中有些符号，例如“=”、“&”、“%”等以两个字符的宽度出现，实际上均应只占一个字符位，敬请读者注意并谅解。

对于以上优点，开始大家不一定都能理解，也不要求逐条记忆，等学到以后的章节，再回过头来想一想，印象就深刻了。

二、缺点

世间一切事物都不是完美无缺的，对于拥有众多优点的 C 也不例外。C 语言的主要缺点表现在：

1. 对程序中的类型转换不加任何限制并且编译程序也不作检查，这自然大大提高了程序的执行速度，但是却带来了很大的不安全性。

2. 表达式的运算符有 40 余种，运算优先级有 15 个之多，不便于记忆。

3. C 语言可写性强而可读性差。这也是 C 语言比较起其它高级语言，自学有一定困难的一个原因。

§1.3 示范程序

C 语言程序一般是由若干个函数组成，每一个函数完成相对独立的功能。这些函数可以保存在一个或几个源程序文件中，这些源文件都以.c 为文件扩展名。在组成一个程序的若干函数中必须有一个且只能有一个名为 main 的函数，称为主函数。不论 main 函数在程序中的位置如何，程序总是从 main 函数开始执行。一个函数在其名字之后一定要有一对圆括号，这是函数的标志。圆括号中的参数可有可无。下面请看两个简单的 C 程序。

例 1.1 在计算机终端屏幕上显示下列字符串：

The C Programming Language.

```
/* The C Programming Language. to print */
main()
{ printf("The C Programming Language.\n");
}
```

这是一个完整的 C 程序，现解释如下：

① /* 和 */之间是注释，只起说明、提示的作用，提高程序的可读性，编译时不产生代码。

② 这个程序只有 main 函数。

③ 函数名后的圆括号中没有参数，此时圆括号仍不能省略。

④ C 程序中无函数和子程序的区别，这一点与 FORTRAN 语言不同。在 FORTRAN 中有返回值的程序称为函数，无返回值的程序称为子程序。在 C 中，不论有无返回值，一律称为函数。

⑤ main()后面有一对花括号，花括号内的部分称为函数体。本程序的函数体只有一个语句，它调用库函数 printf 对输出进行形式加工。输出的字符串要括在双引号之间，其中的"\n"是一个控制代码，代表换行操作，程序执行到这里，将光标移到下一行的开始位置。

⑥ 函数体中也可以没有语句，构成一个空函数。例如：

```
main()
{
}
```

这仍是一个合法的函数，只是没有什么意义罢了。

⑦ C 的每一个语句之后，必须带分号(;)。分号是语句的终止符，不可省略。

例 1.2 计算一个数的正弦值。

```
#include <math.h>
main()
{ float x;
  x=sin(0.19199);
  printf("sin(0.19199)=%f\n",x);
}
```

程序说明：

这也是一个完整的 C 程序。

① 第一行是预处理行，功能是将含有数学函数相关信息的文件包含进来。"sin"是 C 系统的库函数，凡调用数学库中的函数，都必须加上这一行。"math.h"中的".h"是标头文件的扩展名，其中"h"代表 head 即标头的意思，因为包含文件一般放在源文件的最前面。

有两点值得注意：

第一，当程序运行时，系统自动地打开三个标准文件，即标准输入、文件标准输出和文件标准出错输出文件。所以第一个例子中用到标准输出函数 printf 时，不需要使用预处理行。

第二，预处理行的格式在各机器上不完全相同。

本例中的#include <math.h> 是在 PC 机上 Turbo C 环境下的形式。在 VAX 机上该行应写

成：#include <math> 或#include math，即去掉扩展名".h"。关于预处理行，我们后面还有专门的章节(§4.3)详细介绍。

② 第三行是说明语句，说明 x 是浮点型变量。在C语言中所有变量都要先说明，然后才能使用。变量的说明包括两个方面：数据的类型和存储类(即存储方式)。这里数据类型是浮点型，存储类是自动类，缺省。关于存储类的问题，以后还要专门讨论(§4.2)。

③ 第四行是赋值语句，调用 sin 函数求出 0.19199 弧度的正弦值赋给变量 x。

④ 第五行调用库函数 printf，对输出进行形式加工。这里"%f"称格式字符串，指定按实数的格式进行输出，即小数点后保留 6 位数字，小数点前的数字位数不指定，根据实际值的位数输出。这个程序的运行结果是：

sin(0.19199)=0.190813

程序中双引号内的"sin(0.19199)="原样输出，双引号外的 x 的值取代%f 以实型格式输出。

§1.4 常用的输入/输出函数(或宏)

C语言中没有输入、输出语句，输入、输出功能是通过调用库函数来实现的。输入、输出函数很多，本节只讨论最常用的，即

函数： scanf 和 printf

宏： getchar 和 putchar

一、格式化 I/O 标准函数 scanf 和 printf

功能类似于 FORTRAN 语言中的 READ 和 WRITE 语句。

1. 格式化输出标准函数 printf

它的主要功能是在标准输出设备(即终端屏幕)上输出，在输出过程中进行格式化处理。它的一般形式是：

printf("格式控制字符串",参数 1,参数 2,…,参数 n);

其中：参数 1、参数 2 等可以不出现，例如：

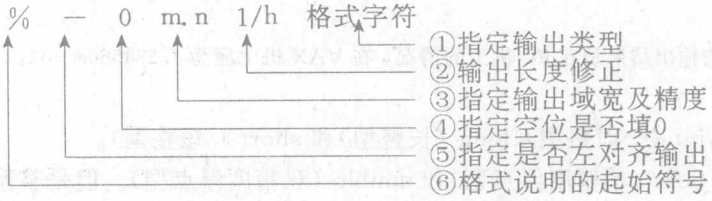
printf("Programming is fun.\n");

格式控制字符串包括两种：

普通字符(包括控制代码)——原样输出(或执行)。

格式说明——说明参数的输出格式，并按格式输出。

格式说明以字符"%"开始，以格式字符结束，中间可以夹有一些任选的附加说明项，其完整形式为：



(1) 格式字符

常用的有：

d——带符号的十进制整数（decimal）

x——无符号的十六进制整数（hexadecimal）

o——无符号的八进制整数（octal）

u——无符号的十进制整数（unsigned）

c——单个字符（character）

s——字符串（string）

e(或E)——指数形式的浮点数（exponent）

其形式为：[－]m.nnnnnne[±]xxx，称为科学计数法。

小数点前只输出不为0的一位数字(m)，小数点后输出的位数(nnn…)由精度值说明，通常缺省值为6。指数部分以e作为前导，e后的指数(xxx)在PC机上为三位数字，在VAX机上为两位数字。

f——小数形式的浮点数（float）

小数点后输出的位数由精度值说明，缺省值为6。

g(或G)——选用%e或%f格式中输出位数较短的一种，不打印无意义的0。

%——百分号本身。

表1.1给出了printf函数中的主要的格式字符的意义及其用法示例。

表1.1 格式字符

格式字符	输出形式	举例	输出结果
d(或i)	十进制整数	int a=123;printf("%d",a);	123
x(或X)	十六进制整数	int a=123;printf("%x",a);	7B
o	八进制整数	int a=123;printf("%o",a);	173
u	不带符号的十进制整数	int a=80;printf("%u",a);	80
c	单个字符	char a=69;printf("%c",a);	E
s	字符串	static char a[]="CHINA";printf("%s",a);	CHINA
e(或E)	指数形式的浮点小数	float a=123.456;printf("%e",a);	1.234560e+002
f	小数形式的浮点数	float a=123.456;printf("%f",a);	123.456000
g(或G)	e和f中较短的一种，不印无效0	float a=123.456;printf("%g",a);	123.456
%	百分号本身	printf("%%");	%

注：表1.1中格式字符为e(或E)时的输出结果是在PC机上的情况，在VAX机上应为1.234560e+02。

(2) 长度修正符

C语言的整数可分为三种类型：int（一般整型）、long（长整型）和short（短整型）。

同样的，实数可分为两种类型：float（单精度浮点型）和double（双精度浮点型）。但是整型的格式字符d、x、o、u并没有区分int、long和short。实型的格式字符f也没有区分float与double。

对整型来说，d、x、o、u是指int型；对实型来说，f是指float型。

为了适应不同长度的数据，可在格式字符前面加一个长度修正符 l 或 h。

l（l 表示“long word”，长字）：对整型指 long 型，例如：%ld、%lx、%lo、%lu；

对实型指 double 型，例如：%lf。

h（h 表示“half－word”，半字）：只用于将整型格式字符修正为 short 型，例如：%hd、%hx、%ho、%hu。

(3) 域宽及精度描述符 m.n

m——域宽，输出的数据所占的字符位数（小数点也占一位）。

n——精度，输出的实型数的小数位数，缺省值为 6。

例如：

%4d 表示输出的十进制整数至少要占 4 位；

%6f 表示输出的浮点数至少要占 6 位（包括小数点）；

%.2f 表示小数点后面的数占 2 位；

%6.2f 表示浮点数共占 6 位（包括小数点），其中小数点后占 2 位。

表 1.2 给出了使用域宽与精度描述符的一些例子。

```
int i=79;
float x;
double y,z;
x=y=333.12345678901234567890;
z=－555.123456789012345678;
```

表 1.2 域宽与精度说明应用实例

格式项	输出项	实际输出	说明
%d	i	79	缺省域宽值
%5d	-i	-79	负数
%5o	i	115	八进制
%5x	i	4f	十六进制
%1d	i	79	域宽值太小
%f	x	333.123444	实型的缺省精度值为 6
%f	z	-555.123457	实型的缺省精度值为 6
%e	z	-5.551235e+002	实型的缺省精度值为 6
%.4f	x	333.1234	域宽缺省只指定精度值
%.8f	y	333.12345679	域宽缺省只指定精度值
%.3e	z	-5.551e+002	域宽缺省只指定精度值
%.18f	y	333.123456789012340000	域宽缺省只指定精度值
%10.3e	y	3.331e+002	精度值小于域宽值
%2.5f	y	333.12346	精度值大于域宽值
%2e	z	-5.551235e+002	域宽值太小

请大家考虑以下几个问题：

① 域宽值小于实际宽度时如何处理？

② 缺省的域宽值与精度值各是多少？

③ 精度说明大于、小于实际精度时，如何处理？

④ float 与 double 的最大精度是多少？

⑤ 符号位如何处理？

⑥ 多余的小数是被截断？还是四舍五入？

需要指出的是：输出数据的实际精度并不主要决定于格式项中的域宽与精度，也不决定于输入数据的精度，而主要决定于数据在机器内的存储精度。例如，在 VAX 机上，对 float 只能提供 7 位有效数字，double 大约有 16 位有效数字。所以增加域宽与精度并不能提高输出数据的实际精度。

(4) 数 0 用以指定数字前的空位是否用 0 填补。有此项则空位以 0 填补，无此项则空位用空格填补。

例如：
```
⋮
float a=1.23;
printf("%08.1f\n",a);
printf("%8.1f\n",a);
⋮
```
输出结果：0 0 0 0 0 1.2

1.2

(5) 负号用以指定输出项是否左对齐输出。不加负号或加正号为右对齐输出，加负号为左对齐输出。

例如：
```
⋮
float a=1.23;
printf("%-8.2f\n",a);
printf("%8.2f\n",a);
⋮
```
输出结果：1.23

1.23

2. 格式化输入标准函数 scanf

它的主要功能是从标准输入设备(即终端键盘)上，按规定的格式输入数值或字符，并将输入的内容放到参数所指定的单元(即地址)中。

它的一般形式是：

scanf("格式控制信息",地址 1,地址 2,…,地址 n);

要特别注意的是，在 printf 函数中的参数是要输出数据的变量的名字，而在 scanf 函数中的参数是变量的地址。执行该语句时，将接收的数据存入变量所在的地址单元中。为了明确，我们直接写成地址 1,地址 2,…,地址 n。

例如：
```
scanf("%d%d",&x,&y);
```
这个语句调用 scanf 函数，实现从终端键盘上输入两个十进制整数，分别放到变量 x、y 的存储单元(地址)中，实际上就是把输入的数值，分别赋给变量 x 和 y。