21世纪高等院校规划教材·计算机类

主　编　吴国凤
副主编　宣善立

C语言程序设计教程

YUYAN
CHENGXU SHEJI
JIAOCHENG

中国科学技术大学出版社

21世纪高等院校规划教材·计算机类

C语言程序设计教程

主　编：吴国凤

副主编：宣善立

主　审：胡学钢　王　浩

中国科学技术大学出版社

2003·合肥

内 容 简 介

本书是面向 21 世纪高等院校规划教材，也是高等院校非计算机专业第一门程序设计课程教材。全书共分 10 章，主要内容包括：C 语言概述、数据类型与运算规则、程序控制结构、数组与字符数据处理、函数、指针、结构体与联合、位运算、文件、面向对象及 C++基础知识等。

全书内容丰富，系统性强，深入浅出。在结构上突出了以程序设计为中心，以语言知识为工具的思想，对 C 语言的语法规则进行了整理和提炼；在内容上注重知识的完整性；在写法上追求循序渐近，通俗易懂。本书各章均有习题，另配《C 语言程序设计教程上机指导》书，适合作为工科高等院校计算机专业及非计算机专业程序设计语言教科书。

图书在版编目（CIP）数据

C 语言程序设计教程/吴国凤主编. —合肥：中国科学技术大学出版社，2003.2

ISBN 7-312-01164-0

I.C… II.吴… 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（2003）第 005441 号

中国科学技术大学出版社出版发行

（安徽省合肥市金寨路 96 号，邮编：230026）

合肥学苑印刷厂印刷

全国新华书店经销

开本：787×1092/16 印张：15 字数：380 千

2003 年 2 月第 1 版 2003 年 2 月第 1 次印刷

印数：1—6000 册

ISBN 7-312-01164-0/TP・311 定价：20.00 元

前 言

C语言是一种在世界范围内被普遍采用的优秀的程序设计语言。C语言之所以流行，主要是由于C语言具有许多优点，表现在C语言可以用于开发系统软件，它既具有高级语言的可理解性又能直接对硬件操作；C语言功能完善，可以适用于各种需要；C语言自身的运算丰富、表达简洁、便于移植，其生成的代码效率高等。

作为培养高素质人才的高等院校均已将计算机基础教育放到了很重要的位置。计算机基础教育不仅包括计算机的应用基础，更重要的是还包括程序设计基础。在程序设计语言的教学中C语言是较好的选择。通过程序设计的学习，可以使学习者掌握程序设计的基本概念和一般方法；掌握算法的概念和要求；并建立模块化设计的概念。本教材正是基于这些要求，并综合各位老师的长期教学经验而编写的。

教材全面介绍了C语言的语法结构，主要包括基本语言元素；C语言的基本数据类型和构造类型（数组、结构体与联合）、指针类型等数据表示方法；C语言的控制结构；模块设计基础的函数以及输入输出的文件。并介绍了面向对象程序设计语言C++的基本概念。

教材组织精练，例题简单，容易理解，并配备了各种类型的练习，便于学习掌握。教材在介绍C语言的语法结构的同时，也强调了计算机算法以及结构化设计方法的概念和作用。教材适用于工科高等学校非计算机专业的C语言程序设计课程。

本书第1章、第2章由谢文佩编写，第3章、第4章由王金玲编写，第5章、第6章、第10章由吴国凤编写，第7章、第8章、第9章由宣善立编写。全书由吴国凤、宣善立统编定稿，由胡学钢、王浩老师主审。由于编者水平有限，书中难免有疏忽、错误之处，恳请读者批评指正。

本书在编写过程中，得到兄弟高校计算机基础教育教师的关心和帮助，教研室的同仁们提出了许多宝贵意见；专家顾问们给予了悉心指导；在出版过程中，得到了中国科学技术大学出版社和合肥工业大学教材科的极大帮助，在此一并表示衷心的感谢。

编 者

2003年1月

目 录

第 1 章　C 语言概述

1.1　C 语言的发展简史与特色

1.1.1　C 语言的发展简史

C 语言是目前世界上最广泛使用的计算机高级程序设计语言。C 语言既可以编写计算机系统软件，也可以编写各种应用软件，所以在数百种计算机语言中，C 语言仍然是目前最流行、最受欢迎的计算机语言。

最初，C 语言是为编写 UNIX 操作系统而设计的。在 C 语言产生之前，操作系统主要是用汇编语言编写的，但是汇编语言依赖计算机硬件，程序的可读性和移植性都比较差。人们一直在寻找一种很接近计算机硬件的高级语言，C 语言就是在这种情况下应运而生的。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963 年英国剑桥大学在 ALGOL 语言基础上增添了硬件处理能力后，推出了 CPL（Combined Programming Language）语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，难以实现。1967 年英国剑桥大学的 Matin Richards 对 CPL 语言作了简化，推出了 BCPL（Basic Combined Programming Language）语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，作了进一步简化，设计出了很简单的 B 语言（取 BCPL 的第一个字母），并用 B 语言编写了第一个 UNIX 操作系统，在 PDP-7 上实现。但 B 语言过于简单，功能有限。1972 年至 1973 年间，贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练，接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。

最初的 C 语言是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。后来，C 语言多次作了改进，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍注意。1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。1987 年又公布了新标准——87ANSI C。目前流行的各种 C 版本都是以这个标准为基础。

现在使用的各种 C 语言编译系统虽然基本部分是相同的，但也有一些不同。在微型机上使用的有 Microsoft C，Turbo C，Quick C 等，它们的不同版本又略有差异。因此，在使用一

个系统之前，要了解所用的计算机系统配置的C编译系统的特点和规定。

1.1.2 C语言的特色

C语言能得到快速发展，受到用户的广泛欢迎，是因为它独具特色。

1．C语言与其他语言的比较

作为高级语言的C语言常被称为中级计算机语言。这并不意味着C语言功能差难以使用，或不像某些高级语言那样完善。C语言被称为中级计算机语言，只是意味着它把其他高级语言的基本结构与低级语言的实用性结合了起来。

（1）C与汇编语言比较

C语言允许对位、字节和地址进行操作（指针），这三者是计算机基本的工作单元，在编制系统程序时要经常用到，所以它适用于写系统程序。由于汇编语言是非结构化语言，含有大量的跳转、子程序调用以及变址，这种结构的缺陷使得汇编语言程序难以读懂，难以维护，也不能移植。而C语言的结构化、模块化克服了汇编语言难读难维护的缺点。C语言又具有汇编语言的功能，目标代码长度也差不多，效率几乎与汇编语言相近，且具有很好的可移植性。

（2）C与其他高级语言比较

C有丰富的运算符，达34种，其中有很多运算符对应于常用的机器指令，比如++、——等运算符可直接编译成机器代码，使用起来简单精炼。

C有多样化的表达式类型，因此可以将C语言说成是表达式语言。C语言中的表达式几乎无处不在，而在其他高级语言中，表达式则要受到很大的限制。

C的数据类型丰富，具有现代语言的各种数据结构。C的数据类型有：整型、实型、字符型、数组、指针、结构体、共用体等。它们能用来实现各种复杂的数据结构：链表、树、队列、栈等。其中指针类型使参数传递简单、迅速，节省内存。

C的输入输出使用的是数据流，而其他高级语言使用的是记录。

C程序生成的机器代码质量高，内存占用少，运行速度快，程序执行效率高。这也是衡量一种语言优劣的重要指标之一。如FORTRAN等语言编程生成的代码长，占用内存多，不能用于实时操作。

2．C是结构化语言

C语言是以函数为模块来编写源程序的，所以C程序是模块化的。

C语言具有结构化的控制语句，如if～else语句，while语句，do～while语句，for语句等。因此是结构化的理想语言，符合现代编程风格的要求。

3．C是编程者的语言

C语言简洁、紧凑，使用方便灵活；一共只有32个关键字，9种控制语句，它们构成了C语言的全部指令；程序书写形式自由，压缩了一切不必要的成分。使得C语言能达到汇编语言的高效率和广泛的应用范围，所以在许多情况下它是编程者首选的计算机语言。

4．C语言的“缺点”

（1）语法限制不严格

C 程序在运行时不做诸如数组边界检查和变量类型兼容性检查，而是由编程者自己保证程序的正确性。故初学者在编程过程中应给予相当的重视，否则很容易发生错误。

（2）程序设计自由度大

C 语言虽有五种固有的数据类型，但不像 Pascal、Ada 语言有很强的类型区分。实际上 C 语言允许几乎所有数据类型的转换。例如在大部分表达式中，字符型和整型数据都可以自由的混合使用；所有类型均可作逻辑型（非零为'真'，零为'假'）；可以自己定义新的类型（typedef）等，还可以把某些类型强制转换为指定类型，编程者可以自由发挥。

以上这些“缺点”实际上也是 C 语言的特点，它可以使程序员有更大的自主性，能编出灵活、优质的程序。但对于初学者来说这些特点却不易掌握，似乎是“缺点”，只有在熟练掌握 C 语言程序设计之后，才能体会出其灵活的特性。正是 C 语言的这种灵活性使它的适应性更强，应用范围更大，成为目前最流行的语言。

1.2　简单的 C 程序介绍

C 语言是一种结构化的程序语言，通过掌握 C 语言程序设计来理解结构化程序设计的思想。下面先介绍几个简单的 C 程序，然后从中分析 C 程序的特点。

【例 1.1】 输出一条信息。

程序如下：

```
main()
{
printf("This is the first program.\n");
}
```

运行结果如下：

```
This is the first program.
```

其中 main()被称作“主函数”，在任何一个 C 程序中都必须有一个且只有一个 main()函数。在 main()函数中有很多 C 语句，用一对大括号括起来。在本例的程序中，只有一条 C 语句。这条 C 语句是一个输出函数，作用是向屏幕输出信息。它将双引号之间的内容原样输出到屏幕。“\n”是换行符，它的作用是将光标移到下一行的开始处。

【例 1.2】 计算两个数的乘积。

程序如下：

```
main()
{
  int x，y，result;          /*定义变量*/
  x=14；y=2;                 /*给变量赋值*/
  result=x*y;                                    /*求积*/
  printf(" \nThe result is %d"，result);         /*输出两个数的乘积*/
```

```
}
```

运行结果如下：

```
The result is 28
```

程序中，“/*… */”表示注释，注释部分不参与也不影响程序的运行，这些注释只是用来帮助人们阅读和理解程序的，注释部分可以加在程序的任何部分。第 3 行是变量定义，说明了 3 个整型变量；第 4 行是赋值语句，给变量 x 和 y 赋值；第 5 行将 x * y 的结果送入变量 result 中；第 6 行是将结果输出到屏幕，“%d”表示在这个位置上将用一个整型数值代替，本程序中是用整数 28 来代替 %d 这个位置。

C 程序中的语句最重要的一个特点就是每条基本语句的后面都要跟一个分号。根据语句的功能，C 语句大致可分为以下六类：

1．说明语句

说明语句用来定义变量的数据类型，如：

```
int x;          /*说明 x 是整型变量*/
float y;        /*说明 y 是实型变量*/
char c[5];      /*说明 c 是字符数组，长度为 5*/
```

2．复合语句

复合语句是用大括号括起来的若干语句，这些语句被看成一个整体。如：

```
{    t=x;  x=y;  y=t;   }
```

在此复合语句中，共有三个语句，每个语句都以分号结尾。

注意：复合语句的大括号后面没有分号。如果复合语句中只有一个语句，那么大括号可以省略。

3．控制语句

控制语句用来规定语句的执行顺序。共有 9 种控制语句：

（1）	if(条件){…} else {…}	条件语句
（2）	for(条件) {…}	循环语句
（3）	while(条件) {…}	循环语句
（4）	do {…} while(条件)	循环语句
（5）	continue;	继续语句
（6）	break;	中断语句
（7）	switch(表达式) {…}	多分支语句
（8）	goto 标号;	转向语句
（9）	return(表达式);	返回语句

上面 9 种语句中，{…}表示复合语句。

4．函数调用语句

由一个函数调用加一个分号构成函数调用语句。如：

```
printf(" How do you do ? ");
```

上面这条语句是由一个 printf 格式输出函数加一个分号，构成函数调用语句。

5．表达式语句

在任何一个表达式的后面加一个分号就构成了一条表达式语句。在 C 语言中，赋值和函数调用都是表达式，所以赋值语句和函数调用语句也是一种特殊的表达式语句。

例如：赋值表达式 x=3 ，在此表达式后加一个分号（即 x=3;）就构成一条赋值语句。

6．空语句

仅由一个分号构成的语句就是空语句。如：

```
;          /*表示什么也不做*/
```

具体的用法在以后的章节会详细介绍。下面再举一个例子来总结 C 语言程序的特点。

【例 1.3】 输入两个整数，输出两个数中较小的数。

```
main()
{ int x，y，z;
  scanf("%d，%d"，&x，&y);
  z=min(x，y);
  printf("\nThe Min number is %d"，z);
}
int min(int x，int y)
{ int z;
  if (x<y) z=x;
  else  z=y;
  return(z);
}
```

若程序输入：3，5↙

运行结果为：The Min number is 3

这个程序包括两个函数，一个是主函数，另一个是 min()函数，min()函数的作用是求出两个整数 x、y 中较小的一个。在执行时，先由 scanf()函数从键盘读取两个数据，这时由用户从键盘上输入 3，5↙（↙表示回车）键。此时 x 被赋值 3，y 被赋值 5。然后执行第 4 行，将 x 和 y 的值传入 min 函数中。在 min 函数中经过判断后，z 中的值就是两个数的较小值。用 return 语句将 z 的值返回函数调用处。此时程序又回到第 4 行，将 min 函数的返回值赋给变量 z。第 5 行，将变量 z 的值输出到屏幕上。

从以上三个例子可以总结出 C 语言程序的几个特点：

（1）　同其他语言一样，C 语言的变量在使用之前必须先定义其数据类型，未经定义的变量不能使用。定义变量类型的语句应在可执行语句的前面，如上例 main() 函数中的第一个语句就是变量定义语句，它必须放在第一个执行语句 scanf()的前面。

（2）　C 程序是由函数构成的。一个 C 程序至少要有一个 main()函数，用户也可以根据需要设计自己的函数，并且在主函数中调用它，如[例 1.3]中的 min()函数。因此，函数是 C 程序的基本组成单位。C 语言中提供了丰富的函数，被称作库函数。标准 C 中提供了 100 多条库函数，Turbo C 和 MS C 中提供了 300 多条库函数。利用这些系统提供的函数可以非常轻松地编写一些功能强大的程序。C 程序的函数式结构，使得 C 程序非常容易实现模块化，便于阅读和维护。

（3） C 程序总是从 main()函数开始执行，不论 main()函数在程序的什么地方，也就是说，可以将 main()函数放在程序的任何位置。

（4） C 程序的书写格式比较自由，可以在一行上写若干语句，也可以将一条语句写在多行上。

注意：在 C 语言程序中英文字母是区分大、小写的，在定义变量时，同样的字母的大、小写代表不同的变量，而 C 语言的关键字和基本语句都是用小写字母表示的。

（5） C 语言中没有专门的输入、输出语句。标准的输入和输出操作是通过 scanf()和 printf()两个库函数来实现的，这充分体现了 C 语言的函数式结构。

（6） C 程序中可以用/* ...*/对任何部分进行注释，一个好的程序都应有必要的注释以提高程序的可读性。

1.3 算 法

1.3.1 算法（Algorithm）的概念

广义地讲算法是解决问题的逻辑步骤，是对特定问题求解步骤的一种描述。简单地说，任何解决问题的过程都是由一定的步骤组成的，把解决问题确定的方法和有限的步骤称为算法。只有通过算法能够描述出来的问题，才能够通过计算机求解。能够用算法描述的问题称为可以形式化的问题。对同一个问题，可以有不同的解题方法和步骤，也就有不同的算法。

计算机算法：是用程序解决问题的逻辑步骤，是指令的有限序列。

计算机算法可分为两大类：

数值运算算法：求解数值；

非数值运算算法：事务管理领域等。

正确的算法有三个条件：

(1) 每个逻辑步骤有可以实现的语句来完成；

(2) 每个步骤间的关系是唯一的；

(3) 算法要必须在运行有限步骤后能终止（防止死循环）。

1.3.2 算法的基本特征

算法是一个有穷规则的集合，这些规则确定了解决某类问题的一个运算序列。对于该类问题的任何初始输入值，它都能机械地一步一步地执行计算，经过有限步骤后终止计算并产生输出结果。归纳起来，算法具有以下基本特征：

1. 有穷性：一个算法应包含有限的操作步骤而不能是无限的。
2. 确定性：算法中每一个步骤应当是确定的，而不能是含糊、模棱两可的。
3. 可行性：算法中每一个步骤必须可以实现，并得到确定的结果。

4．输入：有零个或多个输入。

5．输出：有一个或多个输出。

1.3.3 算法的表示

原则上说，算法可以用任何形式的语言和符号来描述，通常有自然语言、伪代码、流程图、N－S 图、PAD 图、程序语言等。但通常主要采用自然语言、伪代码、流程图、程序语言四种方法。

1．用自然语言表示

自然语言可以是中文、英文、数学表达式等。用自然语言表示通俗易懂，缺点是可能文字过长，不太严格，表达分支和循环的结构不很方便。

【例 1.4】如求数列 1+2+…+m 的值 N,当 N>10000 时结束。

算法可表示如下：

①N=0

②m=0

③m 加 1

④N 加 m

⑤判 N 是否大于 10000

如果满足关系结束；

不满足关系继续执行③。

【例 1.5】对一个大于或等于 3 的正整数，判断它是不是一个素数。

算法可表示如下：

① 输入 n 的值

② i=2

③ n 被 i 除，得余数 r

④ 如果 r=0，表示 n 能被 i 整除，则打印 n“不是素数”，算法结束；否则执行⑤

⑤ i+1→i

⑥ 如果 i≤n-1，返回③；否则打印 n“是素数”；然后算法结束。

或：⑥ 如果 $i\leq\sqrt{n}$，返回③；否则打印 n“是素数”;然后算法结束。

这两个例子的算法都是用自然语言书写的算法。

2．用流程图表示

流程图是用一些图框来表示各种操作。优点是直观形象，简单易于理解，便于修改和交流。美国国家标准化协会 ANSI（Amenrican National Standard Instiute）规定了一些常用的符号，图 1-1 列出了常用的流程图符号。