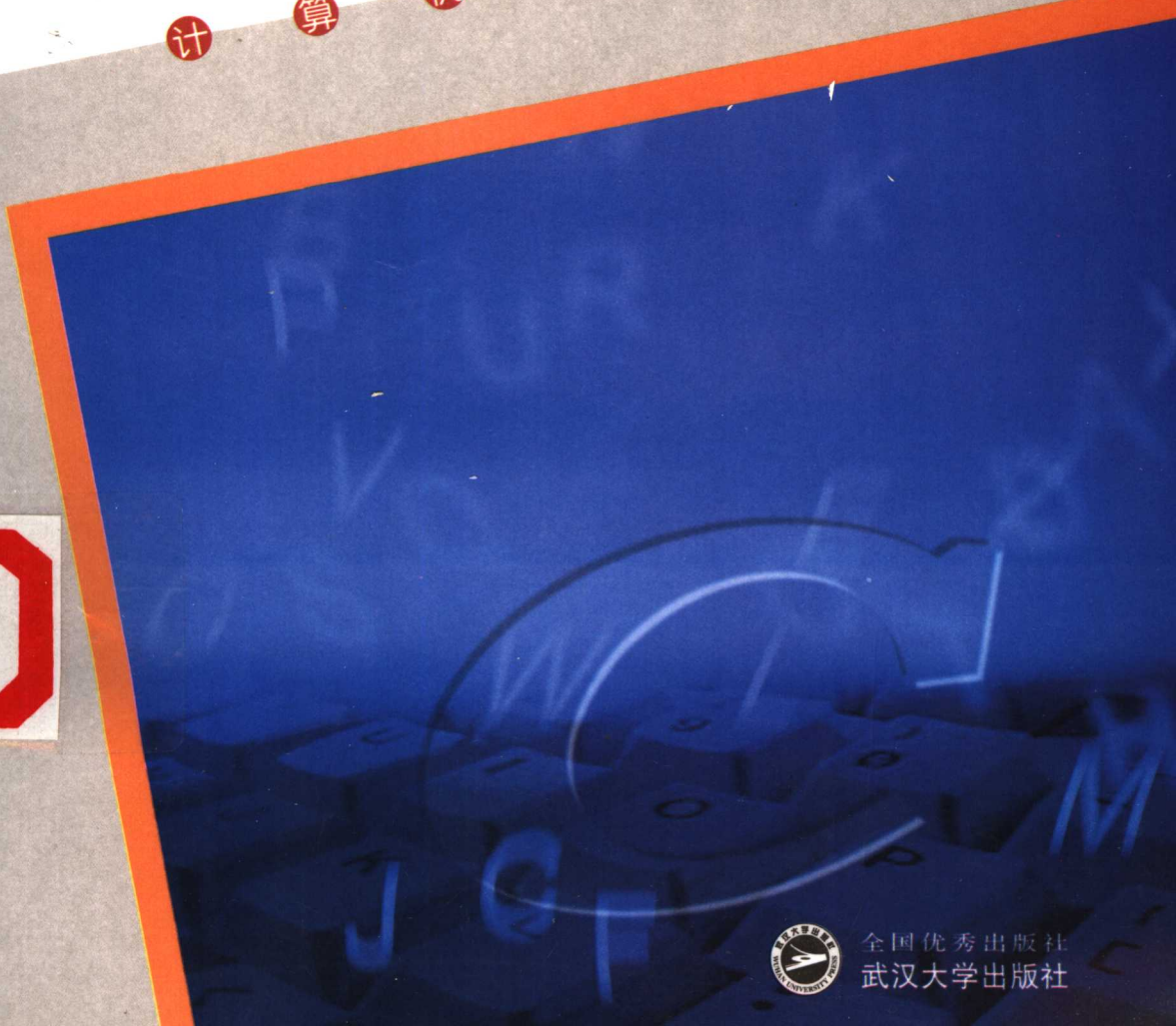
C语言程序设计教程

郑军红 主编

计算机系列教材

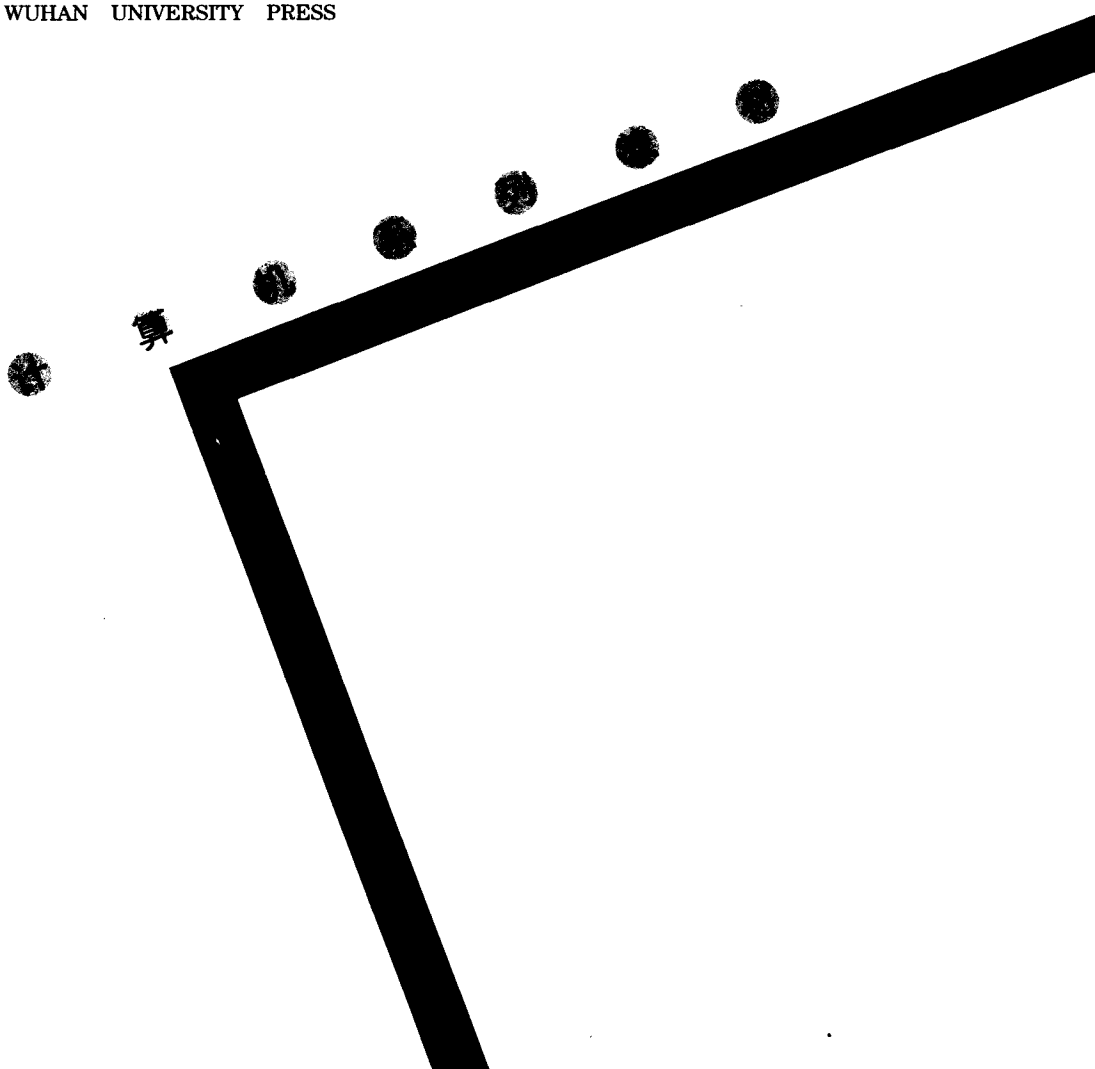


全国优秀出版社
武汉大学出版社

C语言程序设计教程

主　编　郑军红
副主编　胡　岚　胡　雯

武汉大学出版社
WUHAN UNIVERSITY PRESS



图书在版编目(CIP)数据

C语言程序设计教程/郑军红主编;胡岚,胡雯副主编.—武汉:武汉大学出版社,2005.2
计算机系列教材
ISBN 7-307-04430-7

Ⅰ.C… Ⅱ.①郑… ②胡… ③胡… Ⅲ.C语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2005)第001614号

责任编辑:杨 华 黄金文 责任校对:程小宜 版式设计:支 笛

出版发行:**武汉大学出版社** (430072 武昌 珞珈山)
(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.whu.edu.cn)
印刷:湖北省黄冈日报社印刷厂
开本:787×980 1/16 印张:14 字数:252千字
版次:2005年2月第1版 2005年2月第1次印刷
ISBN 7-307-04430-7/TP·160 定价:23.00元

前言

C语言是当前广泛使用的计算机语言之一。由于它简单、易学，使用时方便灵活，所以学习和使用C语言的人越来越多，国内高等院校理工科专业大都开设了这门课程；同时，全国计算机二级考试科目中也有C语言。学好C语言对进一步学习其他计算机语言具有积极的意义，特别是对学习C++语言意义甚大。

C语言程序设计是一门实践性很强的课程，它包含理论学习、编程方法和程序调试三方面的内容。要学好C语言，必须从这三个方面着手。根据当前的形势和教学需要，从C语言教学实际出发，我们编写了这本《C语言程序设计教程》，希望本书能为广大读者提供有益的帮助。

本书全面介绍了C语言程序设计的基础知识和程序设计方法，全书共九章，其中第一章和第二章介绍了C语言的基础知识，第三章介绍了结构化程序设计方法，第四章介绍了C语言函数的应用，第五章、第六章、第七章详细介绍了C语言中特殊的数据类型及其应用（主要有：数组、指针、结构体、共用体和链表），第八章和第九章分别介绍了位运算和文件操作。本书从教学实际出发，语言简洁、通俗易懂、由浅入深，采用了许多与生活、工作实践相结合的例题和应用小程序，比较适合初学者使用。为了帮助读者学习本课程，我们另外编写了《C语言程序设计实验教程与练习题解》，作为本书的配套教材同时出版。

本书可作为高等院校、高职高专计算机专业及其相关专业以及各类培训班的C语言程序设计课程的教材，也可作为工程技术人员的参考用书。

本书的绪论、第一章、第三章、第六章和第七章由郑军红编写，第四章和第五章由胡岚编写，第八章、第九章及附录部分由胡雯编写，第二章由郑军红、胡雯共同编写，全书由郑军红修改定稿。本书在编写过程中，得到了武汉大学王化文教授的大力支持和帮助以及武汉科技大学中南分校各位同仁的协助，我们在此表示衷心感谢！

本书肯定有不足之处，竭诚希望得到广大读者的批评指正。

编　者

2004年11月

目　录

C语言程序设计教程

C 绪 论

C语言程序设计教程

0.1 C 语言的重要地位与学习 C 语言的必要性

0.1.1 为什么要学习 C 语言

自 20 世纪 80 年代以来，计算机应用在我们国家得到了极大的发展，涌现了一大批计算机科技人员，特别是近几年，随着计算机的普及和面向对象程序设计语言的出现，计算机应用已经渗入到社会生活中的方方面面，我们的工作、学习、生活越来越离不开计算机，对计算机的依赖也越来越强。

在计算机的发展过程中，出现了多种程序设计语言，C 语言就是其中的一种，并且以其鲜明的特色受到了人们的普遍欢迎。现在，C 语言不仅被计算机专业人员所使用，而且还广泛地被计算机应用人员所使用，已经成为世界上应用最广泛的几种计算机语言之一。

C 语言是一种极具有生命力的语言，它简单易懂，功能全面，使用灵活方便，移植性好，特别适合编写系统软件，许多原来用汇编语言编写的软件完全可以用 C 语言来编写。可以说，在一定程度上，C 语言比汇编语言更实用。C 语言具有结构化语句，实现了结构化编程，使程序编写变得更容易，更快捷，可以编写出任何类型的程序。

目前，我国大部分高校理工类专业都开设了 C 语言课程。全国计算机等级考试、全国计算机应用技术证书考试(NIT)以及全国各地区组织的大学生计算机统一考试都将 C 语言列入了考试的范围，所以，学好 C 语言是完全有必要的。

0.1.2 学习 C 语言的意义

C 语言接近自然语言，易于理解，编写程序比较自由，经常用来编写计算机系统软件和大型应用软件，也可以用来编写各种各样的计算机应用程序来解决日常生产、生活中碰到的各类问题，以节约时间，提高工作效率。

C 语言是一种面向过程的程序设计语言。编写程序时，必须对计算机完成某项工作进行详尽分析，仔细设计程序的每一个执行步骤，这不仅可以训练我们

的思维能力，还可以提高我们分析问题、处理问题、解决问题的能力。

学习C语言，不仅要学习理论知识，还要学习程序的编写、编辑、编译、连接、运行，掌握程序的动态调试方法，跟踪程序的运行过程。这可以提高我们的实践动手能力和计算机应用操作能力。通过对C语言的学习，掌握算法的分析和运用、数据的存取和计算，了解计算机程序设计的一般方法，可为以后的学习、工作打下基础。

计算机程序设计语言的基本知识是相通的，对数据的存取和处理方法基本上是一样的，学好了一门计算机程序设计语言后，再学习其他计算机程序设计语言，就显得十分简单、易学。学好C语言能为日后学好其他计算机语言奠定良好的基础，特别对以后学习C++语言是大有裨益的。

0.2 C语言的发展历程及其特点

0.2.1 C语言的发展历程

C语言是在B语言的基础上发展而来的。

在C语言出现以前，主要使用汇编语言来编写计算机操作系统等系统软件（包括UNIX操作系统在内）。由于汇编语言对计算机硬件依赖性强，使用局限性大，程序的可读性和移植性都比较差，故许多功能难以实现，不适合实际需要。为了打破这种局面，人们不断寻找一种新的语言来替代它。1960年，产生了一种面向问题的高级语言ALGOL 60，它离计算机硬件比较远，不适合用来编写计算机系统程序。1963年，英国剑桥大学推出了CPL语言。CPL语言离硬件较近，但规模大，难以编写系统程序。1967年，剑桥大学的Matin Richards对CPL语言进行了优化，推出了BCPL语言。BCPL语言是CPL语言的改良版，尽管有许多地方作了改进，但还是有很大的局限性，使用不方便。1970年，美国贝尔实验室的Ken Thompson以BCPL语言为基础，对其进一步简化，设计出了很接近硬件的B语言，并用B语言编写了第一个UNIX操作系统。然而B语言过于简单，功能有限。1972年左右，贝尔实验室的D. M. Ritchie在B语言的基础上设计出了C语言。C语言继承了B语言的优点，又克服了B语言的缺点，使用时比较方便。后来，C语言又做了多次改进，功能日趋完善，但主要还是在贝尔实验室内部使用，直到1975年以后，C语言的突出优点才引起了人们的普遍关注。1977年，出现了不依赖具体机器的C语言编译文本《可移植C语言编译程序》，C语言得到了迅速推广，出现了各种不同版本的C语言。1983年，美国的标准化协会（ANSI）对已经出现了的各种C语言版本进行了扩充，制定了一套完善的

新标准,称为标准 C(ANSI C)。1987 年,美国标准化协会又公布了新标准——87 ANSI C。1990 年,国际标准化组织 ISO(International Standard Organization)开始接受 87 ANSI C 为 ISO C 的标准(ISO 9899 1990)。到 20 世纪 90 年代,出现了编译系统基础部分相同的不同版本的 C 语言编译系统,如 Microsoft C,Turbo C,Quick C,Borland C 等。

0.2.2　C 语言的特点

C 语言从产生到现在,一直长盛不衰,为人们普遍重视,使用广泛,是因为它具有鲜明的特色,其主要特点如下:

(1) 语言简洁、紧凑,使用方便、灵活。C 语言一共有 32 个关键字,9 种控制语句。程序书写相当自由,主要用小写字母表示,语法控制不严格,没有严格的格式要求,源程序简练,编辑快捷。

(2) 运算符十分丰富。C 语言一共有 34 种运算符,运算类型极为丰富,表达式类型多样化,能实现各种复杂的运算。

(3) 数据结构丰富。C 语言具有多种数据结构,数据类型有整型、实型、字符型、数组、指针、结构体、共用体等,能实现各种复杂的数据运算(如链表、树、栈等)。

(4) 具有结构化控制语句。C 语言实现了程序结构化、模块化,将函数作为程序的模块单位,是一种很好的结构化语言。

(5) C 语言是一种中级语言,能直接访问计算机物理地址,进行位运算,可以直接对硬件进行操作,实现了汇编语言的大部分功能,可以用来直接编写系统软件。

(6) C 语言程序生成目标代码质量高,在编译时可以采用多种模式,程序执行效率高,移植性好,能适应各种型号的计算机和操作系统。

0.3　学好 C 语言的正确方法

任何一门课程都有其固有特点和发展规律,要学好一门课程必须有正确的学习方法,才能起到事半功倍的效果。要真正学好 C 语言,应从以下几个方面着手:

0.3.1　端正学习态度,持之以恒

C 语言相对其他计算机语言而言,尽管程序编写比较自由,使用方便灵活,但对程序编写人员的综合能力要求较高,要完全掌握 C 语言的编程技巧,学好 C

语言，必须坚持长期不懈地学习和训练。在学习过程中要有恒心，切忌“三天打鱼，两天晒网”。

0.3.2 全面掌握基本概念，注重理解，灵活运用

C语言的基本概念很多，不仅要学习各种控制语句、各种运算符、各种数据类型和相关算法，还要学习并掌握常用标准库函数的使用。应全面掌握这些基本概念的含义及其使用要求，了解其相关作用，加强理解，在理解的基础上灵活使用。C语言中，要完成某一操作的应用程序可以用许多不同的算法和函数来实现。同一个操作可以用许多不同的具体程序段来完成，哪种方法最好？哪种算法最优？究竟选择何种方式？这需要在编写程序时灵活运用。

0.3.3 独立思考，转换观念，学会正确的思考方法

C语言是一种面向过程的程序设计语言，描述的是完成某个具体操作的步骤。这就要求我们仔细思考，详尽设计，转变传统观念，学会从计算的角度看待问题，进行批判性的学习，敢于创新，以发展的观点来对待学习。

0.3.4 理论联系实际

我们学习任何一种知识都是为了解决生产、生活中碰到的问题，提高效率，所以，理论不能脱离实际，必须与实践相结合。C语言的实践性很强。在学习的过程中，可以利用所学知识，编写一些与生活、学习相关的应用程序来提高自己的程序编写能力，增强学习兴趣。同时，还要加强程序调试实践操作，上机实践进行程序调试特别重要。只有不断地反复操作、调试，才能够熟练地完成程序设计。

第一章 C语言程序的一般介绍

C语言作为一种计算机程序设计语言，与其他计算机语言有相同之处，也有不同之处，下面将对C语言作一些基本的介绍。

1.1 程序与程序设计方法

1.1.1 什么是程序

要利用计算机来处理问题，必须事先编写出使计算机按照人的意愿工作的应用程序。所谓程序，就是让计算机完成某项工作的具体详细规定和先后步骤，是一组计算机指令，每一个指令都使计算机执行一个特定的操作。任何一个程序都必须包含三方面的内容。

1. 算法

算法也称计算方法，是为了解决一个问题而采取的方法和具体步骤。例如去北京旅游，可以事先制定一条详细的旅游路线，先参观什么，后参观什么，列出要参观景点的先后顺序，然后按照这个顺序参观，这就是算法。对同一个问题，可以有不同的算法。就像去北京旅游一样，先参观某个景点或后参观某个景点没有什么区别，只要将旅游路线上的景点参观完就可以了。当然，在旅游中，如果选定了一条合理的路线，则可以节省时间，达到最好的旅游效果。也就是说，尽管解决一个问题的算法有多种，但要考虑到算法的质量，选择合理的算法。

2. 数据结构

数据结构是程序设计时的具体数据对象。任何一个程序都离不开具体的数据操作。这好比参观某个旅游景点需要花多少钱购买门票，参观完毕需要多少时间等。

3. 语言

算法必须通过具体的语言来表述，还要采用合适的方法来表述，才能够形成

程序。可以这样说,算法是程序的灵魂,是解决“做什么”和“怎样做”的问题。一个好的程序必须有一个合理、高效的算法,数据结构是程序要处理的具体对象,语言是描述算法过程的工具。

1.1.2 程序设计的具体方法

程序设计方法分为两大类:**面向过程**的程序设计方法和**面向对象**的程序设计方法。

面向过程的程序设计方法是将完成某项工作的每一个步骤和具体要求都考虑在内来设计程序,程序主要用于描述完成这项工作所涉及的数据对象和具体操作规则,如先做什么,后做什么,怎样做,如何做。C 语言是一种面向过程的程序设计语言。

面向对象的程序设计方法是将任何事物都看成一个对象,它们之间通过一定渠道相互联系,对象是活动的、相对独立的,是可以激发的,每个对象都是由数据和操作规则构成的。程序设计时,主要面对一个个对象,所有数据分别属于不同的对象,封装在对象内,只要激发每个对象完成相对独立的操作功能,整个程序就会自然完成全部操作。可以这样认为,面向对象的程序设计注重对象的结果,忽略对象内部的具体过程。

1.2 C 语言程序的基本结构与书写规则

1.2.1 C 语言程序的基本结构

下面先介绍两个简单的 C 语言程序。

【例 1.1】

```
main()
{
printf("how are you?\n");          /*直接输出字符串*/
}
```

程序运行结果如下:

```
how are you?
```

【例 1.2】

```
#include "stdio.h"
int max(int x, int y)
{int z;
 if( x > y)z = x;
 else z = y;
 return z;
}
main()
{int num1,num2;
 printf("Input the first integer number: ");
 scanf("%d", &num1);
 printf("Input the second integer number: ");
 scanf("%d", &num2);
 printf("max = %d\n", max(num1, num2));
}
```

程序运行结果如下：

```
Input the first integer number: 3 ↙
Input the second integer number: 7
max = 7
```

从上面的例子可以看出：

(1)一个完整的 C 语言程序是由一个 main()函数(又称主函数)和若干个其他函数结合而成的,或仅由一个 main()函数构成。C 语言程序离不开函数,函数是构成 C 语言程序的基础。

(2)一个 C 语言程序总是从 main()函数开始执行,与 main()函数在程序中所处的位置无关。当 main()函数执行完毕时,整个程序也就执行完毕。

(3)任何函数都是由说明部分和函数执行部分(函数体)组成的。其一般结构如下：

```
函数类型说明符  函数名(函数参数表)                    /*函数说明部分*/
{ 说明语句部分;
  执行语句部分;        /*函数体*/
}
```

函数的说明部分主要用于说明该函数的名称、类型、属性、参数名和参数类

型等。

例 1.2 中的函数 max( ),其函数说明各部分如下所示:

```
函数类型说明符  函数名  (参数类型  参数名  参数类型  参数名)
      ↓          ↓       ↓        ↓        ↓        ↓
     int        max     ( int      x      , int      y)
```

函数的执行部分(函数体)写在函数说明部分后面的{ }里面,是函数的具体执行部分,主要用于描述该函数的功能和具体操作过程。函数体一般由说明语句和执行语句两部分构成:说明语句部分放在执行语句部分的前面,主要对函数体里面出现的数据结构进行定义说明。执行语句部分由一条或若干条执行具体操作的语句构成。

下面是例 1.2 中 main( )函数体的示意说明。

```
main( )
{int num1,num2;                                    说明语句
printf("Input the first integer number: ");
scanf("%d", &num1);
printf("Input the second integer number: ");       执行语名   函数体
scanf("%d", &num2);
printf("max = %d\n", max(num1, num2));
}
```

C 语言系统提供了大量的标准库函数供用户使用。如果在程序中使用了 C 语言标准库函数,则必须在程序的开头用#include 将其相应头文件包含进来。附录四列出了 C 语言常用的标准库函数及其相应的头文件,读者可进行查阅。

(4)C 语言没有输入、输出语句,输入和输出操作由 C 语言系统提供标准库函数来完成。

1.2.2 C 语言程序的书写规则

前面曾提到,C 语言程序的书写比较自由,它没有十分严格的格式要求,但在具体编写 C 语言程序时,还是有一些简单的格式要求的,主要如下:

(1)程序一般用小写字母书写,个别的标识符可以用大写字母书写。

(2)所有语句都必须以分号";"结束,即使是函数的最后一条语句也不能