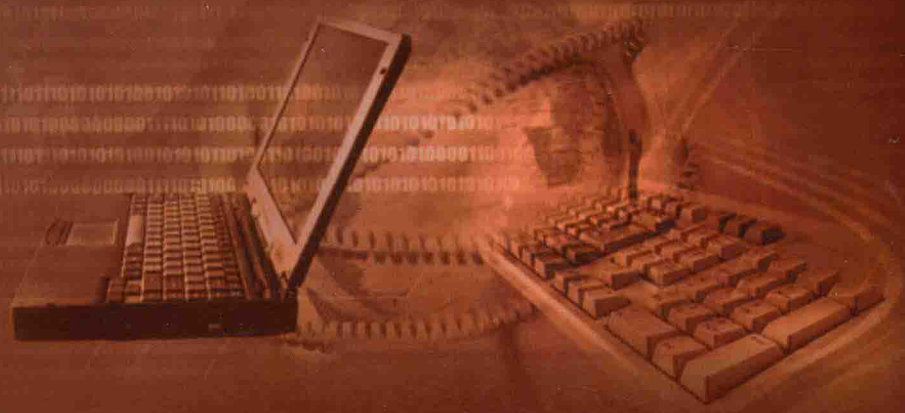
普通高等教育“十三五”规划教材

C语言程序设计

C YUYAN
CHENGXU SHEJI

牛 荣 主编



北京邮电大学出版社
www.buptpress.com

普通高等教育“十三五”规划教材

C语言程序设计

主　编　牛荣
副主编　化希耀　高贤强　张　著

北京邮电大学出版社
·北京·

内 容 简 介

本书以程序设计为主线，编程应用为驱动，通过案例和问题引入内容，重点讲解程序设计的思路和方法，并贯穿介绍与程序设计相关的基本概念和语言知识。本书共12章，内容包括C语言程序设计概述，基本数据类型与表达式，输入、输出与顺序程序设计，选择结构程序设计，循环结构程序设计，函数，数组，指针，结构体与共用体，编译预处理，文件以及C++简介等内容。在结构设计上，本书强调实践，使学生练习编程贯穿整个教材的始终。注重内容的可读性和实用性，以及实例的引入和程序设计思想的建立是本书的重点。

本书可作为高等学校相关课程和计算机等级考试的教学用书，也可作为对C语言感兴趣的读者的自学用书。

图书在版编目(CIP)数据

C语言程序设计/牛荣主编. --北京：北京邮电大学出版社，2016.1(2016.12重印)
ISBN 978-7-5635-4660-2

Ⅰ.①C… Ⅱ.①牛… Ⅲ.①C语言—程序设计 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2016)第008364号

书　　名	C语言程序设计
主　　编	牛　荣
责任编辑	韩　霞
出版发行	北京邮电大学出版社
社　　址	北京市海淀区西土城路10号(100876)
电话传真	010-82333010　62282185(发行部)　010-82333009　62283578(传真)
网　　址	www3.buptpress.com
电子信箱	ctrd@buptpress.com
经　　销	各地新华书店
印　　刷	北京泽宇印刷有限公司
开　　本	787 mm×1 092 mm　1/16
印　　张	18.5
字　　数	460千字
版　　次	2016年1月第1版　2016年12月第2次印刷

ISBN 978-7-5635-4660-2　　定　价：39.00元

前　言

C语言是目前国际上广泛流行的一种结构化程序设计语言，兼具高级语言和低级语言的功能，提供类型丰富、使用灵活的基本运算和数据类型，具有较高的可移植性。C语言不仅适用于开发系统软件，而且是开发应用软件和进行大规模科学计算的常用程序设计语言。

在多年的教学过程中我们发现，许多学生在学完了“C语言程序设计”课程后，虽然能够了解和掌握一些语句的语法知识和语义，但不会应用C语言来编写程序，把编程视为十分艰难而又高不可攀的工作。为帮助学生学会程序设计，我们总结多年的教学经验编写了本书。

本书共包含12章内容，其中第1章，C语言程序设计概述；第2章，基本数据类型与表达式；第3章，输入、输出与顺序程序设计；第4章，选择结构程序设计；第5章，循环结构程序设计；第6章，函数；第7章，数组；第8章，指针；第9章，结构体与共用体；第10章，编译预处理；第11章，文件；第12章，C++简介。

本书由牛荣担任主编，化希耀、高贤强和张著担任副主编，参加编写的还有刘付勇、邬欢欢和王彦群。其中化希耀编写了第1、第2、第3章及附录，牛荣编写了第4、第5章，王彦群编写了第6、第10章，张著编写了第7章，高贤强编写了第8章，邬欢欢编写了9章，刘付勇编写了第11、第12章。

由于计算机科学与技术发展迅速，加之作者水平有限，书中错误在所难免，恳请广大读者和同行批评指正，并多多提出宝贵意见。

编　者

目　　录

第 1 章

C 语言程序设计概述

电子计算机自从 1946 年诞生以来，无论在硬件还是在软件方面，都有了极大的发展，在计算机应用的各个领域也都取得了丰硕的成果。计算机是人们处理信息的一种重要工具，它在人的控制下，按照人的意志正确地工作。人给机器一个指令，机器就完成一个操作。如果把一系列指令输入计算机存储起来，计算机就能按照指令序列实现操作的自动化。

人和计算机之间的通信必须使用人工设计的语言，即计算机语言。要使计算机能够运行起来，为人类完成各种各样的工作，就必须让它执行相应的程序。这些程序都是依靠程序设计语言编写出来的。在众多的程序设计语言中，C 语言有其独特之处。它作为一种高级程序设计语言，具备方便性、灵活性和通用性等特点。同时，它还向程序员提供了直接操作计算机硬件的功能，具备低级语言的特点，适合各种类型的软件开发。因此，C 语言是深受软件工程人员欢迎的程序设计语言。

本章主要从程序设计的角度，介绍有关程序设计的基本概念，结合 C 语言的发展，描述 C 语言程序的基本结构、算法等内容。

1.1 程序与程序设计语言

1.1.1 程序的基本概念

为了让计算机能够按人的意图处理事务，人们必须事先设计好完成各种任务的程序，并预先将它们存放在存储器中。

程序并不是计算机程序设计中独有的概念，在日常生活中我们也常见到这个词。例如，一个活动的日程、一场演出的节目单等，这些程序都是由一项一项活动组成的，有序地完成每一项活动也就实现了程序的目标。

计算机的工作方式有两种：一种是交互式的，即人给机器一个指令，机器就完成一个操作；另一种是程序控制式的，即把计算机要完成的操作用指令按序排好，计算机逐步执行这个指令序列，完成人们希望它做的事情，同时在整个指令序列执行过程中不需要人来干预。为了解决某一特定问题，用某一种计算机语言编写的指令序列称为程序。

所谓程序，实际上是用计算机语言描述的某一问题的解决步骤，是符合一定语法规则的符

号序列。人们借助计算机能够处理的语言，告诉计算机要处理什么(处理哪些数据)以及如何处理(按什么步骤来处理)，这便是程序设计。

为解决某一个问题所编写的程序并不是唯一的，不同的用户所开发的程序也都不完全相同。不同的程序有不同的效率，这涉及程序的优化，涉及程序所采用的数据结构以及算法等多方面的因素。

1.1.2 程序设计语言

要完成程序设计，自然离不开程序设计语言。不同的问题可以用不同的程序设计语言来解决，但解决问题的难易程度会各不相同。了解程序设计语言的发展过程，有助于加深对程序设计语言的认识，能更好地利用程序设计语言来解决有关问题。

当今程序设计语言的发展非常迅速，新的程序设计语言层出不穷，其功能越来越强大。但不管现代程序设计语言的功能如何增强，程序设计语言的种类怎样增多，从其发展历史以及功能看，大致可分成如下几个阶段。

1. 机器语言

本质上计算机只能识别0和1这样的二进制信息，机器语言的程序全部由0和1表示出来。例如，一个16位的计算机，由16个二进制数组成一条指令，这些指令叫机器指令。16个0和1可以组成256个不同的指令或信息，这些指令的集合叫机器语言。人们用这种语言编写出的程序非常烦琐，而且阅读程序、调试程序也都非常困难。

例如，以下是某计算机的两条机器指令。

加法指令：　1 0 0 0 1 0 0 0

减法指令：　1 0 0 0 0 0 0 0

另外，机器语言依赖于具体机器而不是通用的，因此只有在早期使用这种语言，现在已经没有人直接使用这种语言编程了。但机器语言是计算机能直接识别和执行的唯一语言，因此其他各种语言编写的源程序最终都要通过语言处理程序翻译成等价的机器语言程序——目标程序后，计算机才能执行。

2. 汇编语言

20世纪50年代中期，为了减轻人们使用机器语言编程的负担，开始采用一些助记符号来表示机器语言中的机器指令，这样便形成了汇编语言。助记符一般都是采用代表某种操作的英文字母的缩写，与机器语言相比，便于识别和记忆。例如，上例中的两条指令用汇编语言描述如下。

```
ADD A,B
SUB A,B
```

不过计算机不能直接执行用汇编语言编写的程序，它必须经过一个叫汇编程序的系统软件翻译成机器语言程序后才能执行。通常称前者为源程序，后者为目标程序。

汇编语言指令和机器语言指令之间具有一一对应的关系，因此，不同的计算机其汇编语言也不尽相同，并且程序编写时仍需要对计算机的内部结构比较熟悉，依然比较烦琐。但相对于机器语言来说，汇编语言要简单多了。因此，在实际中，如果程序运行时间要求比较严格，程序与硬件操作联系紧密，人们还是常用汇编语言编写有关程序来解决相关问题。

3. 高级语言

机器语言和汇编语言都属于低级语言，后来人们创造出高级语言，它非常接近于人类的自然语言和数学语言，它的一个语句往往对应几条机器指令。一般来说，高级语言不再是面向机器的了，而是面向过程的语言，即把解题过程的每一步用计算机语言描述出来，再配上适当的语言处理程序，计算机就能执行。因此，这种语言也称“算法语言”。常见的高级语言有 FORTRAN 语言、BASIC 语言、COBOL 语言、PASCAL 语言、C 语言等。

C 语言作为 UNIX 操作系统的主要使用语言，由于 UNIX 操作系统的成功，C 语言得到了广泛使用。C 语言具有很强的功能和高度的灵活性。和其他高级语言一样，C 语言能提供丰富的数据类型及丰富的供运算和数据处理用的运算符。

高级语言有很多种，不管哪种高级语言，其源程序都必须经过相应的语言处理程序翻译成机器指令才能执行。

4. 面向任务的程序设计语言

利用算法语言求解一个复杂的问题，必须先要分析解决问题的过程，描述问题如何求解，然后才能用算法语言进行程序设计来实现。面向任务的程序设计语言是非过程化的语言，也就是说，不需要知道问题是如何求解的，只需要描述需要求解的问题是什么，然后便可用程序设计语言来实现。

数据库操作语言便是一种面向任务的程序设计语言。例如，设在某数据库应用系统中，有一个学生情况表 XS，若要从表中查找学生的信息，可以使用数据库查询语言(SQL)。采用 SELECT 语句便可完成这个任务，SELECT 语句描述如下：

```
SELECT XSNO,XSNAME,XSAGE,XSSEX FROM XS
```

该语句从 XS 表中查找到学生的 XSNO、XSNAME、XSAGE、XSSEX 等方面的信息。至于 SELECT 语句是如何进行查询的，用户就不必了解了。

面向任务的程序设计语言可以提高应用程序的开发速度和质量，也可使应用程序迅速地修改。同时，非计算机专业人员也能很方便地使用面向任务的程序设计语言开发自己的程序。这类语言在应用系统的开发中使用较为广泛，尤其是一些管理信息系统应用程序的开发。

5. 面向对象的程序设计语言

面向对象的程序设计语言在 20 世纪 90 年代开始流行。现在，面向对象的程序设计语言已成为程序设计的主流语言之一。由 C 语言发展出来的 C++就是一种非常优秀的面向对象的程序设计语言。

面向对象的方法是一种分析方法、设计方法和思维方法的综合。面向对象方法学的出发点和所追求的基本目标是使人们分析、设计和实现一个系统的方法尽可能接近人们认识一个系统的方法。

在面向对象编程中，程序被看成是相互协作的对象集合，每个对象都是某个类的实例，所有的类构成一个通过继承关系相联系的层次结构。面向对象的程序设计语言具有对象生成功能、消息传递机制以及类和继承机制。

对象是对客观事物的抽象，面向对象的编程，就是针对客观的事物设计程序。因此，面向对象的编程是非常直观的。面向对象的程序设计方法比面向过程的程序设计方法更清晰，更

适合于开发大型复杂的软件。

综上所述，可以知道每一种语言都有它的优势和劣势。对于不同的问题，要根据实际情况来选择程序设计语言，以便更高效、更优质地解决相关的问题。

1.2 C语言的发展及特点

1.2.1 C语言的发展概况

C语言是当今社会应用广泛，并受到众多用户欢迎的一种计算机语言。它既可作为系统软件的描述语言，也可用来开发应用程序。

C语言的出现是与UNIX操作系统紧密联系在一起的，C语言本身也有一个发展过程，目前仍然处于发展和完善之中。

从历史发展来看，C语言起源于1968年发表的CPL(Combined Programming Language)，它的许多重要思想来自于Martin Richards在1969年研制的BCPL以及以BCPL为基础的由Ken Thompson在1970年研制成的B语言。Ken Thompson用B语言写出了第一个UNIX操作系统，用在PDP-7计算机上。D. M. Ritchie于1972年在B语言的基础上研制了C语言，并用C语言写成了第一个在PDP-11计算机上实现的UNIX操作系统。1977年出现了独立于计算机的C语言编译程序——可移植C语言编译程序，从而大大简化了把C语言编译程序移植到新环境所需要做的工作，这就使得UNIX操作系统迅速在众多的计算机上得以实现。例如，VAX、AT&T等计算机系统都相继开发了UNIX。随着UNIX的广泛使用，C语言也就迅速得到推广。

1983年美国国家标准化协会(ANSI)根据C语言问世以来的各种版本，对C语言的发展和扩充制定了新的标准，称为ANSI C。1987年ANSI又公布了新标准——87ANSI C。

目前在微型计算机上使用的有Microsoft C、Quick C、Turbo C等多种版本。这些不同的C语言版本，基本部分是相同的，但在有关规定上又略有差异。本书以Visual C++ 6.0的环境对C语言进行介绍。Visual C++ 6.0是一种快速、高效的编译程序。它不仅提供了一个集成开发的环境，同时也按传统方式提供了一个命令行编译程序版本，以满足不同用户的需要。

1.2.2 C语言的特点

事实证明，C语言是一种极具生命力的语言，它的特点是多方面的。一般可归纳如下。

(1)C语言具有结构语言的特点，程序之间很容易实现段的共享。它具有结构化的流程控制语句(如if-else语句、switch语句、while语句、do-while语句、for语句)，支持若干种循环结构，允许编程者采用缩进的书写形式编程。因此，用C语言设计出的程序层次结构清晰。

(2)C语言的主要结构成分为函数，函数可以在程序中被定义完成独立的任务，独立地编译成代码，以实现程序的模块化。

(3)C 语言运算符丰富,运算符包含的范围很广。C 语言把赋值、括号、强制类型转换都作为运算符处理。灵活地使用各种运算符可以实现在其他高级语言中难以实现的运算。

(4)C 语言数据类型丰富。数据类型有整型、实型、字符型、数组型、指针型、结构体型、共用体型等,能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是 C 语言的指针型数据的运算,更是灵活、多样。

(5)C 语言允许直接访问物理地址,即可直接对硬件进行操作,实现汇编语言的大部分功能。C 语言的这一特点,使得它成为编制系统软件的基本语言(UNIX 的绝大部分程序就是由 C 语言写成的)。

(6)C 语言语法限制不太严格,程序设计自由度大,这样使 C 语言能够减少对程序员的束缚。"限制"与"灵活"是一对矛盾。限制严格,就会失去灵活性;而强调灵活,就必然放松限制。从这个角度来看,使用 C 语言编程,要求编程者对程序设计技巧更加熟练一些。

(7)用 C 语言编程,生成的目标代码质量高,程序执行效率高。同时用 C 语言编写的程序可移植性好。

C 语言有很多优点,但是它也存在一些缺点,如运算优先级太多,数值运算能力方面不像其他高级语言那样强,语法定义不严格等。尽管 C 语言目前还存在一些不足之处,但由于它目标代码质量高、使用灵活、数据类型丰富、可移植性好而得到广泛的普及和迅速的发展,成为一种受到广大用户欢迎的实用型程序设计语言,同时也是一种在系统软件开发、科学计算、自动控制等各个领域被广泛应用的程序设计语言。

1.2.3　结构化的 C 语言

1. 结构化程序设计方法

结构化程序设计是一种程序设计的技术。它最早由 E. W. Dijkstra 提出,并由 Bohm 和 Jacopini 从理论上证明了只用三种基本的控制结构,通过组合和嵌套就能实现任何单入口单出口的程序——这就是结构化程序设计基本原理。这三种控制结构是顺序结构、选择结构和循环结构。

结构化程序设计支持自顶向下、逐步求精的结构化分析方法。

对于一个较复杂的问题一般不能立即写出详细的算法或程序,但可以很容易写出一级算法,即求解问题的轮廓,然后对一级算法逐步求精,把它的某些步骤扩展成更详细的步骤。在细化过程中,一方面加入详细算法,一方面明确数据,直到根据这个算法写出程序为止。

自顶向下、逐步求精的方法符合人类解决复杂问题的思维方式,用先全局后局部、先整体后细节、先抽象后具体的逐步求精过程开发出的程序不仅能显著提高软件的生产率,而且可以保证获得结构清晰、易于测试、修改和验证的高质量的程序。

2. 结构化程序设计的特点

(1)有一个入口、一个出口。

(2)没有死语句(永远执行不到的语句),每一个语句都至少应当有一条从入口到出口的路径通过它。

(3)没有死循环(无限制的循环)。

在介绍 C 语言的程序结构之前我们先看下面的两个小程序。

【例 1.1】 打印"hello,world"。

```
#include <stdio.h>
main( )
{
printf("hello,world\n");
}
```

运行结果为

```
hello,world
```

程序中 main 表示主函数,每个 C 语言程序都必须有一个 main 函数。函数体由{}组成。本程序内只调用函数 printf,参数是"hello,world\n"。printf 是一个打印输出的库函数,双引号内的字符原样输出。\n 是 C 语言的换行符,即输出

```
hello,world
```

后换行,下一个输出将在下一行的最左边开始。每一个语句结尾有一分号。

注意:C 语言程序中没有行号,本书中的行号仅仅是为了描述解释方便而设。

【例 1.2】 从键盘输入两个数,然后输出两数中的较大值。

```
#include <stdio.h>
main()
{int a,b,c;                          /*定义三个整型变量*/
scanf("%d,%d",&a,&b);                /*输入两个整数*/
c=max(a,b);                          /*调用 max 函数,把函数值赋给 c*/
printf("max=%d\n",c);                /*输出 c 的值*/
}
int max(int x,int y)                 /*定义函数 max*/
{int z;                              /*函数内说明语句,定义 z*/
 z=(x>y)?x:y;                        /*当 x>y 时把 x 赋给 z,否则把 y 赋给 z*/
return(z);                           /*返回 z 值*/
}
```

本程序由两个函数组成,主函数 main 和被调用函数 max。max 函数的作用是求 x 和 y 中的较大者,并把其赋给 z,通过 return 语句将 z 的值返回主函数 main。返回值通过函数名 max 带回到主函数的调用处。

运行结果为

```
12,23
max=23
```

3. C 语言的程序结构

(1)C 语言程序由一个或若干个函数构成,必须有且只能有一个主函数 main,不管 main

函数放在前或后，程序总是从 main 函数开始执行。被调用的函数既可以是库函数，也可以是自己设计的函数。

(2)函数由函数头和函数体两部分构成，函数说明的一般形式为

```
函数类型 函数名(形参名)
形参说明;
{
内部说明语句;
可执行语句;
}
```

函数头又称函数说明部分，包括函数类型、函数名、形参表和形参说明等，形参可以有也可以没有，但函数名后的()不能省略。函数体由一对{}括起来，可以有若干个语句，通常由说明语句和可执行语句组成。

(3)每行可以写一个或多个语句，一个语句也可以写在多行上，但每个语句必须以分号结束。

(4)可以用/* */对 C 语言程序中的任意部分进行注释，以提高程序的可读性。

(5)组成一个程序的若干个函数可以保存在一个或几个源程序文件中，每个文件分别进行编译，然后再连接起来形成可执行文件。

4. 程序运行

程序运行必须经过四个阶段：编辑—编译—连接—运行。这四个阶段都是由系统提供的系统程序完成的。

(1)操作系统：是管理和协调计算机系统中全部软、硬件资源的一个管理软件，是所有系统程序和应用程序运行的基础，没有操作系统的支持，一切程序都无法工作。程序执行的每一步都由操作系统发出命令，计算机才能执行各种操作。用户上机的第一件事就是用操作系统启动计算机，之后计算机才处于可用状态，操作系统在开机整个过程中常驻内存。微机上常用的操作系统有 DOS、Windows、UNIX、OS/2 等操作系统。

(2)编辑程序：输入源程序是程序实现的第一步。计算机系统为用户提供了编辑程序，利用编辑程序可以输入源程序并对它进行修改。修改完的源程序存放在磁盘上，以后需要时再调入计算机进行编译。

(3)翻译程序：将高级语言源程序翻译成等价的机器代码。翻译程序有以下两种。

● 编译程序：编译程序的主要功能是将高级语言程序翻译成机器语言程序，另外它还包含查错的功能。在翻译过程中如果发现程序有错，则不生成目标程序，并向用户报告出错信息。用户必须重新调用编辑程序修改源程序，然后再次进行编译，直到无错为止，这时产生目标程序。一般高级语言如 C、PASCAL、FORTRAN 都采用编译方式。编译方式速度快、占用内存少，但使用不够方便。

● 解释程序：解释程序也将高级语言源程序翻译成机器代码，与编译程序不同的是，它是边翻译边执行，翻译一句执行一句，不产生整个程序的目标程序，当再次运行该程序时还要重复翻译的过程，所以解释方式效率低、执行时间长、占用内存多，但使用灵活方便。早期的