·高等学校计算机基础教育教材精选·

C语言程序设计

李文杰 主编　徐英慧 副主编
周长胜 主审

清华大学出版社

·高等学校计算机基础教育教材精选·

C语言程序设计

李文杰 主编　徐英慧 副主编
周长胜 主审

清华大学出版社
北京

内 容 简 介

C语言作为一种简洁高效的语言，目前是绝大多数高校本科生学习程序设计的入门语言。本书所有作者均来自教学第一线，具有多年的教学经验，尤其是针对非计算机专业的学生特点，经过深思熟虑的分析研究，努力做到使本书概念清晰，不拘泥于细节，注重实用，运用大量的例题和精选的习题，帮助本科新生能够快速掌握C语言程序设计的基本方法。

本书可以作为高等院校非计算机专业学生的正式教材，也可以作为本科新生或C语言爱好者的自学读物。另外，本书还配套辅助教材《C语言习题、实验指导和课程设计》，可以帮助读者熟练掌握C语言。

图书在版编目（CIP）数据

C语言程序设计 / 李文杰主编. —北京：清华大学出版社，2010.9
（高等学校计算机基础教育教材精选）
ISBN 978-7-302-23154-7

Ⅰ. ①C… Ⅱ. ①李… Ⅲ. ①C语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2010)第120309号

责任编辑：白立军
责任校对：梁　毅
责任印制：孟凡玉

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京市清华园胶印厂
经　　销：全国新华书店
开　　本：185×260　　**印　　张**：18.25　　**字　　数**：419千字
版　　次：2010年9月第1版　　**印　　次**：2010年9月第1次印刷
印　　数：1～4000
定　　价：29.00元

产品编号：035864-01

前言

当今社会，大家对计算机已不陌生，但是会用计算机的人是否都懂得计算机是如何完成交给它的任务的，相信会有很多人不明白。这也是目前各高校无论计算机专业还是非计算机专业，甚至文科专业，都要开设程序设计这类课程的原因之一。C 语言作为一种简洁高效的编程语言，并且支持结构化编程，在讲究效率的时代，对于帮助人们掌握程序设计的基本思想和方法，进而更好地理解计算机是如何工作的，会有极大的帮助。

本书虽然像传统的教材那样，按照学习的规律，顺序介绍 C 语言的语法和用 C 语言解决实际问题的方法，但却并不是一本 C 语言的语法书。本书完全是从应用的角度出发，来对 C 语言的语法进行介绍，所以不要把本书作为语法大全。

本书的读者对象是大学本科新入学的学生，尤其是非计算机专业的学生，他们希望通过学习 C 语言程序设计，来理解计算机的工作。尽管他们可能今后并不是一名程序员，但他们希望知道计算机能做什么以及如何做，以便当他们在今后的工作中，需要向程序员提出他们的软件需求时，能够很好地与软件开发人员沟通，这是非计算机专业本科生所必备的能力。

程序设计是一项工作，程序就是这项工作的产品。如果要制造产品，必须有好的原材料，这些原材料对于程序来说，就是语言的语法知识、各种问题的解决方法和步骤。所以本书在介绍 C 语言语法的同时，介绍了大量的各种常见问题的解决方法和思路（算法），读者通过对这些内容的掌握，等到动手创造自己的作品（程序）时，就会手到拈来。所以读者一定要尽可能多地积累算法，真正做到熟能生巧。

本书第 1 章和第 8 章由李文杰编写，第 2 章和第 9 章由徐英慧编写，第 3 章由贾艳萍编写，第 4 章由张良编写，第 5 章由崇美英编写，第 6 章由黄宏博编写，第 7 章由李颖编写，第 10 章由方炜炜编写。全书由李文杰负责统稿，北京信息科技大学计算中心主任周长胜副教授主审。在书稿规划和撰写的各个阶段，刘梅彦和刘亚辉多次提出了建设性的意见，清华大学出版社的白立军也为教材的出版做了工作，使本书得以及时出版。

本书的所有实例都是在 Visual C++ 6.0 集成开发环境调试的，所给出的程序运行情况也是在 Windows XP＋Visual C++ 6.0 环境下进行的。本书配套的教学资源有多媒体教学课件、所有实例程序的源代码，这些资源可以从清华大学出版社网站获取。

因编者水平所限，书中错误在所难免。欢迎读者就书中内容与作者进行交流。作者 E-mail 地址：jszx_jsjjc@yahoo.com.cn、computer_base@126.com。

作　者

2010 年 5 月于北京信息科技大学计算中心

目录

第1章 程序设计概述

1.1 程序和程序设计

1.1.1 程序的概念

计算机作为当今社会应用领域最广泛的设备，其本质就是一台“执行程序的机器”。

程序这个词来源于生活，通常指完成某件事情的一种既定方式和过程。从表述方面看，可以将程序看成对一系列动作的执行过程的描述。日常生活中可以找到许多“程序”实例。例如，一位班车司机早上起床后的行为可以描述如下：

(1) 起床。

(2) 刷牙。

(3) 洗脸。

(4) 发动汽车。

(5) 开车接职工上班。

这是一个直线型程序，是最简单形式的程序。描述这种程序就是给出一个包含其中各个基本步骤的序列。如果按顺序实施这些步骤，其整体效果就是完成早上起床这件事情。

现在再来看一个复杂的程序例子，学生到餐馆去吃饭这件事情可以描述如下：

(1) 进入餐馆。

(2) 查菜谱。

(3) 向服务员点菜。

(4) 可能由于某种原因，菜谱上的菜没有。学生可以有两种选择：

① 回到第(2)步(进一步查找其他想吃的菜)。

② 放弃在此吃饭，到其他餐馆吃。

(5) 若点的菜有，在此吃饭。

(6) 吃完饭离开餐馆。

这个程序比前一个复杂得多。可以看到，这个程序不是一个平铺直叙的动作序列，其中的步骤更多，还出现了分情况处理和可能出现的重复性动作。

顾名思义，计算机是计算的机器。计算机在设计时，就是让它通过完成一系列的简单操作来实现计算。每个简单操作的学名就是“指令”，一台计算机所能理解的简单操作(指令)

的集合称为计算机的指令系统。早期的计算机程序就是完成一项计算任务的指令序列。

1.1.2 程序设计

计算机本身是一个通用的计算机器，它执行一个完成某项任务的程序，就可以把它变成一个特定用途的计算机。通常把描述(或者叫编制)计算机程序的工作称为程序设计或者编程，这项工作的产品就是程序。由于计算机的本质特征，从计算机诞生之初，就有了程序设计这项工作。

正是由于程序设计这项工作，使得计算机这个通用的机器，可以在工厂大量统一制造，然后再通过给它运行不同的程序，使得计算机可以应用到不同领域，所以程序设计的工作在计算机应用的实现中非常重要。

1.1.3 程序设计语言

语言一词通常是指人们生活和工作中进行交流所使用的沟通方式，如汉语、英语等，它是随着人类社会的发展而逐步形成的。

为了能和计算机沟通和交流，指挥它工作，同样需要一种与它交流的方式。这种交流方式必须是人和计算机都能理解的方式。

目前的计算机都是采用冯·诺依曼原理制造的，因此计算机从一出生就懂得的语言只有一种，那就是二进制语言，也就是 0 和 1 这两个符号组合表达的含义。前面提到的指令就是用 0 和 1 来表达的，如下列 0 和 1 组合：

```
0001000100000010010000
0001000100010000101000
0011010100000110000100
0000000100010000110000
0010010000001100000100
0010001000000010111000
```

想描述的是计算算术表达式 x×y+z(这里的符号 x、y、z 分别代表地址为 2000、2010 和 2100 的存储单元)，然后将结果保存到单元 2110 的计算过程(程序)。

这种只有 0 和 1 的语言称为**机器语言**。显然一般人如果使用这样的语言来编写程序，将是非常困难的，只有非常专业的程序员才能用它写程序，因为它是计算机可以直接理解的唯一一种语言。

既然人类交往是采用像汉语、英语这样的自然语言，而计算机内部又是采用二进制的机器语言，程序设计是要人与计算机进行沟通，正如两个说不同语言的人，如果你说汉语，他说英语，而双方又只会自己的语言，这两个人就很难沟通，必须找出解决办法。或者说汉语的人学会英语，或者说英语的人学会汉语，也可以找一个既懂汉语又懂英语的人做翻译，程序设计语言的发展也是类似的。计算机要想懂得人类语言比较难，人类要学习计算机的机器语言也很难，所以计算机程序设计语言的发展过程就是一个双方逐渐妥协的过

程。由于机器语言太难掌握，又很容易出错，所以随着计算机的发展，计算机语言也有了使用助记符表达的机器语言，即**汇编语言**或者符号机器语言。例如，上面机器语言程序使用某机器的汇编语言程序书写就成了如下形式：

```
load  r0  x
load  r1  y
mult  r0  r1
load  r1  z
add   r0  r1
save  r0  t
```

虽然它比机器语言程序变得简单和易懂了，但由于它是与机器语言一一对应的，没有程序结构，在编写和理解上仍然很困难。

妥协的过程在继续，随着计算机技术的发展，程序设计语言也进入到逐渐接近人类自然语言——称为高级程序设计语言（**高级语言**）的阶段。1954 年，一种称为 Formula Translation(公式翻译)的程序设计语言诞生，命名为 FORTRAN。它是为科学、工程问题或企事业管理中的那些能够用数学公式表达的问题而设计的，其数值计算的功能较强。简单地说，这种高级语言由于使用非常接近人与人交往的英语符号，同时又包含多种程序流程的控制机制，这些控制机制使编程者可以摆脱许多具体细节，方便了复杂程序的书写，写出的程序也更容易阅读，有错误也更容易辨认和改正。

从 FORTRAN 语言诞生至今，人们已设计的程序设计语言超过千种，其中大部分只是实验性的，只有少数语言得到了广泛使用。随着时代的发展，今天绝大部分程序都是用高级语言编写的。人们也已习惯于用程序设计语言特指各种高级程序设计语言了。在高级语言(例如本书介绍的 C 语言)的层面上，描述前面同样的程序片段只需一行代码：

```
t=x*y+z;
```

它表示要求计算机求出等于符号右边的表达式，而后将计算结果存入由 t 代表的存储单元中。这种表示方式与人们所熟悉的数学形式直接对应，因此更容易阅读和理解。

当然，计算机是不能直接执行除机器语言程序以外的汇编语言程序和高级语言程序的。人们在设计好一个语言之后，还需要开发出一套实现这一语言的软件，这种软件被称为语言系统，也常被说成是这一语言的实现，它类似于在计算机和人之间的一个翻译，是一个计算机能理解的程序。

高级语言的基本实现技术是**编译**和**解释**。

采用编译方式实现高级语言，其过程是人们首先针对具体语言(如 C 语言)开发出一个翻译软件，其功能就是将采用该种高级语言编写的程序翻译为所用计算机的机器语言的等价程序。用这种高级语言写出程序后，只要将它送给翻译程序，就能得到与之对应的机器语言程序。此后，只要命令计算机执行这个机器语言程序，计算机就能完成人们所需要的工作。

采用解释方式实现高级语言，其过程是人们首先针对具体高级语言开发一个解释软件，这个软件的功能就是读入这种高级语言的程序，并能一步步地按照程序要求工作，完

成程序所描述的计算。有了这种解释软件，只要直接将写好的程序送给运行着这个软件的计算机，就可以完成该程序所描述的工作了。

简而言之，编译方式需先把源程序全部翻译成可执行的目标代码，然后再运行此目标代码；解释方式则是按照源程序中语句的顺序逐句进行分析、解释并立即执行。目前在实际程序设计工作中，最常用的是第一种实现方式，也就是编译方式。本书介绍的C语言就是采用这一方式，后面还会进一步介绍这种实现方式的一些具体情况。

1.2 C语言概述

1.2.1 C语言简介

正如前面所述，自从第一个高级语言FORTRAN诞生以来，高级语言的种类不下千种。C语言作为众多优秀高级语言中的一种，由于它既可以编写计算机系统软件，又可以编写各类应用软件，目前绝大多数高校仍然把它作为给学生教授程序设计思想和方法的第一门程序设计语言。

1971年，美国电话电报公司(AT&T)贝尔实验室的Dennis Ritchie根据早期的一种编程语言BCPL(Basic Combined Programming Language，B语言)编写了C语言，并于1972年首次在运行UNIX操作系统的DEC PDP-11计算机上使用。1973年，他与K. Thompson又用C语言把UNIX系统重写了一遍，使得C语言与UNIX操作系统共同发展起来，并逐渐发展成为一个通用的程序设计语言。1983年，美国国家标准化协会(ANSI)成立了一个委员会，制定了C语言的标准，也就是ANSI C，该标准的制定为C语言的发展奠定了基础，并成为以后的C语言标准。现在各种C语言的版本都是在此基础上发展起来的。

1.2.2 C语言程序示例

下面通过一个具体的应用来认识C语言程序。

例1.1 计算输入的两个整数的和并显示结果。

```
#include "stdio.h"
int main()                          /*求两个整数的和*/
{
    int a,b,sum;                    /*变量说明*/
    printf("请输入两个整数:");      /*提示用户输入数据*/
    scanf("%d%d",&a,&b);            /*从键盘读入两个整数,保存到变量a和b中*/
    sum=a+b;                        /*完成求和,并把和值放到变量sum中*/
    printf("sum=%d\n",sum);         /*输出和值*/
    return 0;
}
```

程序运行情况如下(注意其中的3、5是程序运行时从键盘输入的):

```
请输入两个整数:3 5
sum=8
```

用C语言编写的程序称为C源程序,上述内容中的从"#include "stdio.h""开始,到"}"结束的内容在计算机上将被保存到一个文件扩展名为.c的文件,例如example.c中,example.c称为C源文件。

例1.1这个简单程序可分为两个基本部分:第一行是个特殊行,以"#"开始,称为**预处理**行,此处说明程序中要用到C语言系统提供的标准功能(这里是输入scanf和输出printf),为此要将标准库文件stdio.h包含在程序中,用双引号或者尖括号把文件名括起来,以后大家会看到,本书所有的程序都会包含这一行(因为所有程序都要使用printf),并且还会用到其他的标准库文件;其后的几行是程序基本部分,描述程序所完成的工作。

在程序的基本部分中,main称为函数名,通常称为主函数。在C语言中,程序中一个名称后跟一对圆括号代表函数这种语法形式,在C程序中会看到很多这种形式。main前面的int表示主函数的值是整数类型的,在主函数内的最后有一行"return 0;"与之对应。这部分内容的详细理论在后面第6章会介绍,这里大家先记住这么表达就可以了。

从"{"开始到"}"结束的部分,称为函数main的函数体,其中又分为两大部分:说明部分(本例中为"int a,b,sum;")和执行部分(除说明部分以外的其他内容)。说明部分对执行部分要用到的变量、类型等进行说明,这是C语言的一个规定,叫**先说明后使用**,在编写C程序时一定要记住这个规定。执行部分在函数中完成函数的功能,对主函数来说,也就是完成C程序的功能。

在C程序中,用分号";"表示一个意思表达的结束,相当于人与人之间交流时的一句话的结束。如说明部分的整数型变量定义"int a,b,sum"结束处有分号,执行部分每个分号结束的部分都代表C语言的一个语句的结束,**每条C语言的语句都要以分号结束**,即使它是最后一条语句(如return 0后的分号也必须有)。

C语言程序中以"/*"开始,"*/"结束的部分称为C语言的注释,注释符号中的"/"和"*"之间不能有任何其他符号,必须成对出现,注释的内容可以使用中文或者英文。注释是对程序中的语句或者程序段进行说明,主要目的是帮助阅读程序的人理解程序,它对程序的逻辑功能没有任何作用。尽管如此,现代程序设计中非常重视注释的使用,主要原因是,现代程序开发往往不是一劳永逸的,程序的修改和完善是经常性的工作,即使是自己编写的源程序,如果程序有成千上万行,没有很好的注释,过一段时间再读,也有可能不知道当时是怎么考虑的了,再加上现在公司里开发人员流动特别快,如果让新来的开发人员修改前面开发人员的没有注释的程序,难度有多大是不可想象的事情。所以软件公司对开发人员有一项严格的要求,那就是所有源程序一定要有比较详尽的注释。可以不夸张地说,软件源代码文件的80%以上文本是注释很常见,所以也希望大家建立良好的注释习惯。

C语言程序中对中文的使用比较受限制,除了注释中可以使用外,另外一个可以用中文的地方就是printf中的双引号里面。其他地方包括任何标点符号都不要使用中文格

式，中文符号(汉字或标点符号)在程序中起逻辑作用是初学者常犯的一个错误。

C语言的程序书写格式非常自由，上面那个程序除第一行的预处理行外，如果能写下，整个程序的其他行都写在一行里和现在这样书写，从程序功能的角度来说是没有任何区别的。尽管如此，还是提倡大家养成良好的书写习惯，程序不仅要实现预定的功能，还要考虑程序的可读性。按照现在这样的逐层缩进格式来书写C语言程序，可以使读程序的工作变得容易，前面说过，程序的修改和完善工作通常都要先读懂程序，可读性好也是对程序的一项要求。在多年程序设计实践中，人们在这方面取得了统一认识：由于程序可能很长，结构可能很复杂，因此程序必须采用良好格式写出，所用格式应很好体现程序的层次结构，反映各个部分之间的关系。人们普遍采用的方式如下：

(1) 在程序中适当加入空行，分隔程序中处于同一层次的不同部分。

(2) 同层次不同部分对齐排列，下一层次的内容通过适当退格(在一行开始加空格)，使程序结构更清晰。

(3) 在程序中增加一些注释性信息。

上面程序例子的书写形式就符合这些做法。

开始学习程序设计时就应养成注意程序格式的习惯。虽然对开始的小程序，采用良好格式的优势并不明显，但对稍大一点的程序，情况就不一样了。有人为了方便，根本不关心程序的格式，想的只是少输入几个空格或换行，这样做结果是使自己在随后的程序调试检查中遇到更多麻烦。所以，这里要特别提醒大家：注意程序格式，从一开始写最简单的程序时就养成好习惯。

实际的C程序可能比前面的简单例子长得多。下面再给大家看一个C语言程序的例子，这个例子的内容并不要求大家现在就掌握，只是给大家一个C语言程序的总体印象。

例 1.2 根据输入的圆盘半径，计算圆盘的面积。

```
#include "stdio.h"
double area(double r)                          /*计算半径为 r 的圆盘面积*/
{
    double s;
    s=3.1415926*r*r;
    return s;
}
/*计算圆盘面积的函数结束,下面开始主函数*/
int main()
{
    double r;
    printf("请输入半径:\n");
    scanf("%lf",&r);
    printf("半径为%.2f 的圆盘面积是%.5f\n",r,area(r));
    return 0;
}
```

这个程序的运行情况如下：

程序显示：请输入半径：
用户输入：3.12 并回车
程序显示：半径为 3.12 的圆盘面积是 30.58152

这个程序同第一个程序相比，除了主函数 main 外，又多了一个计算圆盘面积的函数 area，它先于主函数定义。

一般来说，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(包括 0 个)其他函数结合而成的，不管 main 函数在什么位置，C 程序的执行总是从 main 函数的“{”开始，到 main 函数的 return 语句结束(若没有 return，则到 main 的“}”结束)。所以 C 语言程序有且只有一个主函数 main。

这一部分，大家见识了很多关于 C 语言程序的内容，其中很多内容是现在不需要掌握的，现在把需要掌握的内容总结一下。

(1) 函数是 C 语言程序的主要成分，有且必须只有一个 main 函数。

(2) 主函数的描述方法。

(3) C 语言程序一般开始于 #include "stdio.h"。

(4) 每条 C 语言的语句都以分号结束。

(5) C 语言程序中可以使用“/*”和“*/”括起来对程序的注释内容。

(6) C 语言程序中必须遵守先说明后使用的原则(说明的方法及内容以后会逐步学到)。

(7) C 语言程序最好采用逐层缩进的可读性好的格式书写。

(8) C 语言源程序保存在扩展名为.c 的文件中。

目前知道这些就可以了，程序中的其他内容会逐渐学习到。

1.2.3 为何要学 C 语言

计算机及其应用可以说已经渗透到人们工作和生活的方方面面。计算机的本质是“程序的机器”，程序和指令的思想是计算机系统中最基本的概念，只有懂得程序设计才能懂得计算机，真正了解计算机是怎样工作的；通过学习程序设计可以进一步了解计算机的工作原理，更好地理解和应用计算机，学会用计算机处理问题的方法。不从事软件开发的大学生通过学习程序设计培养提出问题、分析问题和解决问题的能力；具有编制程序的初步能力；能更好地理解软件的特点和生产过程，能与程序开发人员更好地沟通与合作。而程序设计能力则是软件开发人员的基本功。

既然程序设计可以帮助人们更好地认识计算机，要学习程序设计，就要选择一个程序设计语言来学习它。C 语言作为一种高级程序设计语言，相比其他高级程序设计语言有着公认的一些特点，这些特点尽管对于只学习一种程序设计语言的人来说可能无法体会，但这里还是把它们列举一下：语言本身简洁、生成的目标代码质量高和使用灵活等。另外学会 C 语言，对于掌握目前应用中比较热门的语言(如 Java、C++、C# 等)都很有帮助。