21世纪高等学校规划教材 | 计算机科学与技术

C语言程序设计
（第3版）

黄保和　江弋　等编著

清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

C语言程序设计
（第3版）

黄保和　江弋　等编著

清华大学出版社

内容简介

本书的内容包括程序设计语言概念、程序设计概念、C语言的词法语法、三种基本控制结构、函数的定义和调用以及数组、结构体、指针等复杂数据类型的定义和应用、文件操作等。每章都配有例题和练习题。

本书的特点：①重点介绍C语言基本的、常用的语法，忽略不常用或重复的语句。②注重程序设计语言的共性，使读者学习C语言之后具有自学其他程序设计语言的能力。③以介绍C语言的语法为线索，通过精心组织的示例，将程序设计的一般方法和技术贯穿在示例分析中，适合于案例教学。

本书适合作为高等学校非计算机专业“C语言程序设计”课程的教材，特别适合于周授课3学时或3学时以下的学生使用。

图书在版编目(CIP)数据

C语言程序设计/黄保和，江弋等编著. --3版. --北京：清华大学出版社，2014
21世纪高等学校规划教材·计算机科学与技术
ISBN 978-7-302-36866-3

Ⅰ. ①C… Ⅱ. ①黄… ②江… 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(2014)第127068号

责任编辑：盛东亮
封面设计：傅瑞学
责任校对：李建庄
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：14.75　　**字　　数**：365千字
版　　次：2006年9月第1版　2014年9月第3版　　**印　　次**：2014年9月第1次印刷
印　　数：1～3000
定　　价：29.00元

产品编号：057338-01

前言

目前，我国高校非计算机专业的计算机教育普遍实施“三个层次”的教学方式，即计算机应用基础、计算机技术基础和结合专业的计算机应用。“C 语言程序设计”属计算机技术基础课程。

受到课时数不足、配套课程不够等客观因素的制约，任课老师普遍感到要在非计算机专业的学生中上好“C 语言程序设计”课程不容易，学生也普遍感到学好课程有一定难度。为了使课程教学达到预期效果，我们认为应该明确和解决好以下问题。

1. 课程目的

程序设计是计算机技术在各行各业应用的基础。对非计算机专业的学生，在今后的工作中不一定要自己开发应用程序，但一定会使用计算机程序，也有可能与计算机专业人员合作开发本专业领域的应用程序。因此，学习程序设计的一般原理，掌握基本的程序设计方法和技术具有重要的现实意义。“C 语言程序设计”课程的目的就是帮助学生掌握一种高级程序设计语言(C 语言)，并运用语言开展程序设计实践，使学生具备应用高级语言开展程序设计的初步能力。

程序设计必须借助于语言。“C 语言程序设计”课程必然应包含“C 语言”和“程序设计”两方面的内容。必须明确，学习程序设计的一般方法和技术是本课程目的，学习 C 语言只是为了实现程序设计，C 语言是程序设计的工具语言。本课程将通过对 C 语言的词法、语法介绍，通过各种基本控制结构的实际案例分析，向学生介绍程序设计的基本方法。通过上机实验，使学生掌握程序调试和测试方法。通过本课程的学习，应使学生在应用计算机解决问题的能力得到进一步的提高，为后续的计算机应用课程打下坚实的基础。

2. 课程内容

对程序设计语言和程序设计技术的熟练掌握和深入理解是计算机系学生的两大基本要求。“C 语言程序设计”课程是计算机专业学生本科四年中要学习的系列计算机课程之一，属专业基础课，其重要性不言而喻，因此一般安排比较充足的教学课时。计算机系学生学习“C 语言程序设计”，主要的课程内容是深入地研究 C 语言的词法和语法，并通过程序设计练习进一步理解程序设计语言。其教学的重点是程序设计语言，程序设计技术是副产品。

在非计算机专业学生中开设“C 语言程序设计”课程，不能全盘复制计算机专业学生的课程内容。一方面因为课时不够，另一方面还必须对本课程的内容做适当的外延。本课程的基本教学内容应包括：了解软件开发的一般过程，软件开发和软件运行平台的使用，软件工程的基本概念，C 语言的基本词法和语法，结构化程序设计的方法，模块和函数的概念，顺序、分支和循环三种基本程序控制结构的概念和实现，程序调试和测试的一般方法等。

考虑到能提供给本课程的学时有限、学生在学习本课程之前数学基础和专业知识还较

欠缺,加上在一、二年级其他功课压力较重等客观因素,具体安排课程内容时应注意以下几点:

(1) 介绍程序设计语言时应有重点而不是面面俱到。C 语言语法比较庞杂,有些语句可以相互替代,有些语句不常使用。课程中要重点介绍基本的、常用的语句,忽略重复的、不常用的语句。

(2) 注重程序设计语言的共性。学生以后不一定用 C 语言写程序,而本课程又不可能介绍所有的计算机语言。所以应该介绍计算机程序设计语言共性的东西,让学生掌握语言比较学的方式方法,使学生具有自学其他程序设计语言的能力。

(3) 不要从理论上讲解程序设计。不能安排太多的时间专门讲授程序设计理论,而应以介绍语言为线索,通过合适的实例分析,将程序设计的一般方法和技术传授给学生。

长期的计算机基础教学实践,长期的计算机基础教学研究,是本书形成与完善的基础。我们希望奉献给老师和学生一本易教易学的教材。

为配合新一轮的本科教学改革,本书第 3 版在原第 2 版的基础上,对各章内容作了大幅度调整和精简。重修设计例题,更适合于案例教学。补充和调整习题,使习题更有针对性,更有梯度,能满足不同程度的学生课后练习。放弃了原第 12 章面向对象的程序设计。

本书由黄保和策划和统稿。江弋编写第 1 章,黄洪艺编写第 5、8、10 章,李慧琪编写第 2、3、4、9 章,张丽丽编写第 6、7 章,黄保和编写第 11 章。本教材编写和改版过程中,得到厦门大学公共计算机教学部各位老师的大力支持,借此表示感谢。

在使用本书过程中,如有宝贵意见和建议,恳请与我们联系(huangbh@xmu.edu.cn)。

编著者

2014 年 6 月

目 录

第1章 绪论

C语言是当今世界上使用最广泛的程序设计语言之一。本章主要介绍程序设计语言、程序设计、算法的概念及程序设计的主要步骤，同时介绍C语言的产生和发展以及C程序的构成和C程序调试测试等。

1.1 程序设计语言

用计算机解决问题，就要编写程序。所谓程序，是指以某种程序设计语言为工具对解决问题的操作序列的描述。程序表达了解决问题的过程，用于指挥计算机进行一系列操作，从而实现问题的解决。程序设计语言就是用户用于编写程序的语言，是一种人造语言。程序设计语言是计算机软件系统的重要组成部分，可划分为机器语言、汇编语言和高级语言等。用高级语言编写的程序（称其为源程序）要通过编译程序翻译成机器语言程序（称其为目标程序）后计算机才能执行，或者通过解释程序对源程序边解释边执行。

1. 机器语言

最初的程序是用机器语言编写的，机器语言由一系列基本指令组成，这些指令可以由机器直接执行。机器语言编写的程序是由二进制代码组成的代码序列。使用机器指令进行程序设计要求程序设计者具有扎实的计算机专业知识，对机器的硬件有充分的了解。这种程序的可读性差，而且由于不同机器的指令系统不同，程序的可移植性差，大大地限制了程序的通用性。用机器语言描述问题的处理方式也与人们习惯的思维方式有较大差距。

2. 汇编语言

为了便于记忆，人们很自然地用助记符号来代表机器语言中的01代码，这种用助记符号描述的指令系统称为汇编语言。为了把汇编语言编写的程序转换为机器语言程序，人们开发出了称为“汇编程序”的“翻译程序”。汇编语言指令与机器语言指令是一一对应的，因此，汇编语言也是与具体计算机硬件相关。汇编语言除了可读性比机器语言好外，同样也存在机器语言的缺点，尤其是描述问题的方式与人们习惯相差太远。

3. 高级语言

为了提高程序开发效率，针对机器语言和汇编语言的缺点，20世纪50年代中期，IBM

公司的一个编程小组提出了一个设想,能否用数学公式来编写人们想要进行的计算,然后再让计算机将这些公式翻译(解释)为机器语言呢?最终该团队于1955年完成了ForTran(Formulation Translation)的最初版本,这是第一个高级编程语言。从那时以后,各种高级语言相继涌现,而能引起广泛关注和使用的主要有BASIC、Algol、COBOL、Pascal、C等。在众多的高级语言当中,最为流行、也是最成功的当数C语言,它既可用来写应用软件,也可用来写系统软件(如UNIX操作系统等)。如今流行的C++和Java也都是基于C语言的升级版。

与低级语言(机器语言、汇编语言)相比,高级语言的表达方式更接近人类自然语言的表述习惯,具有较好的可读性,并且不依赖于计算机的具体型号,具有良好的可移植性。

1.2 程序设计

程序设计是一门用计算机解决问题的科学,在掌握程序设计的错综复杂的内容之前,有必要对什么是程序设计概念有一个感性认识。

1.2.1 程序设计概念

程序设计的目的就是用计算机解决问题。用计算机解决问题大体上要经过两个步骤,第一步是通过分析问题构造出一个解决问题的算法(即解决问题的方法和步骤),这个过程称为算法设计;第二步是用一种程序设计语言(如C语言)将该算法表达为程序,这个过程称为编码。程序设计的两个步骤是相辅相成的,作为一个程序设计新手,只能从简单的问题入手,而简单问题的解决方法也较简单,所以算法设计阶段不会有什么困难,倒是C语言对于初学者来说是全新的,因此初学程序设计时,编码常常会是两个步骤中较为困难的阶段。但可以确定的是,随着你对C语句语法的了解,随着你编程实践的不断增加,编码会变得越来越简单。相反地,随着遇到的问题愈来愈复杂,算法设计会变得更加困难和更具有挑战性。

1.2.2 程序设计的步骤

对于初学编程的人来说,往往简单地把程序设计理解为编写一个程序,认为能根据实际问题直接用计算机语言编出一个程序就行了,这样理解是不全面的。事实上,程序是程序设计的最终产品,需要经过中间每一步的细致加工才能得到。如果企图一开始就编写出程序,往往会适得其反,达不到预想的结果。对于大型的程序,由算法设计和编码两个步骤组成只是大略说法,严格地说,程序设计一般应遵循以下步骤。

1. 分析问题

用计算机解决问题,首先要对问题进行分析,以便确定这个问题需要计算机做些什么?如果没有把要解决的问题分析清楚之前就贸然开始编写程序,只能起到事倍功半的效果,而且很难得到满意结果的。因此,分析问题,弄清楚要解决的问题并给出问题的明确定义是解决问题的关键。

2. 系统设计

在弄清要解决的问题之后，就要考虑如何解决它，即如何做？从本质上看，用计算机解决问题的方式就是对数据进行处理，因此，首先要对问题进行抽象，抽取出能够反映本质特征的数据并对其进行描述，即给出问题的数据结构设计；然后考虑对数据如何进行操作以获得问题的结果，即进行算法设计。不同的程序设计方式在处置数据结构和算法两者之间关系是有所区别的。在面向过程程序设计中，把数据结构设计和算法设计分开考虑，而面向对象程序设计是把两者结合成对象来考虑。

3. 编码

在进行数据结构设计和算法设计时，往往采用某种与具体程序设计语言无关的语言（如伪代码或自然语言等）来描述算法。这样做的目的是为了避免一开始就陷入程序设计语言的具体细节中。因为过多地涉及实现细节，不利于从较高抽象层次对问题本质的东西进行考虑，并造成对设计过程难以把握和理解。当然，用伪代码（或自然语言或框图）描述的算法是不能被计算机执行的，必须用具体程序设计语言把它们表示出来，即编程实现。因此，用某种程序设计语言编写程序，也是对问题处理方案的描述，并且是最终的描述。程序文本保存在一个文件或多个文件中。包含程序文本的文件称为源文件，即源程序文件。

4. 测试与调试

源程序文件要经过编译程序翻译成等价的目标程序文件，并将一系统列目标文件及库文件链接在一起生成可执行文件，程序才能被计算机执行。

程序编写好后，还需要进行调试和测试。只有经过调试的程序才能正式运行。所谓调试，是指找出程序中的错误并改正错误。因此，调试又称查错。而所谓测试，是指通过精心设计的一批测试用例（包括输入数据和与之相应的预期输出结果），然后分别用这些测试用例运行程序，看程序的实际运行结果与预期输出结果是否一致，以尽可能多地发现程序中的错误。因此，测试的目的也是为了发现程序中的错误，而不是证明程序正确。调试和测试往往是交替进行的，通过测试发现程序中的错误，通过调试进一步找出错误的位置并改正错误。这个过程往往需要重复多次，这一步骤是程序设计中最困难的一步。

程序中的错误通常有三种：语法错误、逻辑错误和运行异常错误。**语法错误**是指源程序中存在违反 C 语法规则的地方，如语句后遗漏了分号“;”或忘记了定义变量等，程序编译时可以发现这类错误，并会给出“出错信息提示”和出错位置，用户可以根据编译器提供的提示信息修改源程序；**逻辑错误**是指程序没有完成预期的功能，如源程序中错将 c=a%b 写成 c=a/b 或将该用“==”的地方却误用了“=”等，编译程序时，编译器发现不了这类错误，但程序运行时，会产生错误的结果，甚至根本就无法显示结果。遇到此类错误，用户往往需要通过设置断点，跟踪程序的运行过程等测试手段，才能发现它们。因此程序设计中最致命的错误是逻辑错误，因为它不太容易被发现，并会导致程序不能正确地解决问题。在程序中找出并修正逻辑错误的过程称为**程序调试**，它是软件开发中不可缺少的一个环节。俗话说，“三分编程七分调试”，说明程序调试的工作量要比编程的工作量大得多。**运行异常错误**是指程序对程序运行环境的非正常情况考虑不足而导致的程序运行异常终止。逻辑错误和运

行异常错误可能是编程阶段导致的,也有可能是系统设计阶段或问题分析阶段的缺陷。这两类错误一般都要通过测试才能发现。

无论你采用何种测试手段,都只能发现程序有错,而不能证明程序正确。即使程序通过了良好的测试,总还是会有一些隐蔽的错误。作为一个编程者,你的任务就是尽可能地对程序进行彻底测试,不断找出并改正错误。

5. 整理文档资料

对于程序设计人员来说,平时的归纳和总结很重要。程序员应将平时的源程序和各种文字资料进行归类保存,以便今后的查询。最后还要编写使用和维护该程序的说明书,供程序用户参考。

6. 运行与维护

程序通过测试后就可交付使用了,但程序在使用中还需要不断得到维护。程序维护可分为正确性维护、完善性维护和适应性维护。程序即使经过大量测试,源代码中依然可能存在逻辑错误,这些错误会在程序的使用过程中不断暴露。当出现一些不常见的未预料到的情况时,之前隐藏的异常错误就会使程序运行失败。一个程序使用了一段时间后,客户会期望程序能做些别的事情,对于用户提出的完善和扩充程序功能的要求,程序开发者应考虑响应这些要求。无论是排除逻辑错误还是要完善程序功能,在任何情况下都需要有人查看程序,做出必要的修改,并确保这些修改能使程序更好地工作。

如果对程序的功能进行了较大的更改,应发布程序的新版本。

1.3 C语言发展和C++简介

1.3.1 C语言发展简述

1972年,为了编写UNIX操作系统,贝尔实验室的D. M. Ritchie设计并实现了C语言。

后来,C语言多次做了改进,使其逐渐成熟,先后移植到各种计算机上,现在C语言已成为世界上使用最广泛的程序设计语言之一。在软件产业蓬勃发展的今天,越来越多的程序都是由C语言编写。C语言在其30多年的发展中,涌现了众多不同的版本,它们都普遍遵守两个重要的标准:一是Brian W. Kernighan和Dennis M. Ritchie于1978年合著的名著*The C Programming Language*,被称为标准C。二是美国国家标准化协会(ANSI)于1983年开始制定并于1988年最终完成的ANSI标准,即"ANSI C"。ANSI C比原来的标准C有了很大的发展。20世纪90年代后,尽管C++、Visual C++开发如火如荼,C语言也并没有停滞不前,更没有被替代,版本不断更新、升级,但C的特征未变,C仍然是C。

1.3.2 C++简介

当C语言程序达到一定的规模后,维护和修改显得相当困难。为了满足管理程序复杂性的需要,贝尔实验室的Bjarne Stroustrup博士于1979年开始对C语言进行了改进和扩

充，并引入了面向对象程序设计的内容，最初取名为“带类的C”，1983年改名为C++。在经历了3次重大修订后，于1994年制定了标准C++草案，后又经不断完善，成为目前的C++，它具有以下特点：

(1) C++是C语言的超集。C++由两部分组成：一是过程性语言部分，这部分与C语言无本质区别，一般遵守ANSI C标准；二是类和对象部分，这是C语言所没有的，它是面向对象程序设计的主体。

(2) C++充分保持了与C语言的兼容性，绝大多数C语言程序可以不经修改直接在C++环境中运行。

(3) C++仍然支持面向过程的程序设计，是一种理想的结构化程序设计语言，又几乎全部包含了面向对象程序设计的特征。

(4) C++继承了C语言的高效率、灵活性等优点。用Bjarne Stroustrup博士的话来说，C++使程序“结构清晰、易于扩展、易于维护而不失效率”。

(5) C++是一种标准化的、与硬件基本无关的、广泛使用的程序设计语言，具有很好的通用性和可移植性。C++程序通常无需修改，或稍作修改，即可在其他计算机系统上运行。

目前主要的C++开发环境有Borland公司的Borland C++、Microsoft公司的Visual C++等。由于Visual C++既能支持C语言程序，又能运行C++程序，同时考虑到C++对C语言的兼容性，因此本书所有例子的编程环境均为Visual C++ 6.0。

1.4 C语言程序的基本构成

C语言属于人造语言，因此C语言与自然语言之间有很多相似之处。自然语言(如英文)由句子、单词和字母组成，C语言也由基本符号(字符)构成一系列单词(语法元素)，由多个单词构成句子(语句)，再由多个语句构成程序。不同的是C语言具有较严格的语法规则，在语义上也不像自然语言那样具有多义性，程序文本所代表的语义是单一的、确定的。正像写文章时分章节、段落和层次一样，C程序也具有层次结构，但无论程序规模大小，C程序都是由函数组成的，而函数又是由语句组成的。

为了使大家从整体上对C语言程序的基本结构有一个感性认识，下面先看一个简单程序。

[**例 1.4.1**] 已知圆柱体的半径r、高h，求其体积v。

解 程序如下：

```
/*                                          }
 * file: compute volume                     }  程序注释
 * 这程序是用于计算圆柱体体积                }
 */                                         }
#include<stdio.h>                           }  包含库文件
#define PI 3.14159                          }  常量定义
float volume(float r,float h);              }  函数原型声明
```

```
void main()
{float radius, height,vol;          /*定义 radius, height, vol 为 float 型变量*/
printf("input radius, height: ");   /*提示用户输入半径 radius 和高 height 的值*/
scanf("%f%f",&radius,&height);      /*从键盘输入两个实数分别存入变量 radius
                                      和 height*/
vol = volume(radius, height);       /*调用函数 volume()计算圆柱体的体积*/
printf("vol = %f",vol);             /*输出圆柱体的体积*/
}
/*
*函数: volume
*用法: v = volume(r,h);
*该函数的功能是计算半径为 r、高为 h 的圆柱体的体积
*/
float volume(float r, float h)
{float v;                           /*局部变量的定义*/
v = PI*r*r*h;
return v;
}
```

主程序　函数注释　函数体　函数定义

以上C程序由注释、预处理命令、主函数、子函数定义等组成。尽管其结构简单,但它是本书所有程序的样板,应将其作为C语言程序组织的范例。

1. 注释

在一个具有良好书写风格的程序开头,往往是该程序的注释部分,注释有助于人们阅读和理解程序,有利于对程序的维护。在C语言中,注释是程序中位于符号/*和符号*/之间的所有文字,可以占连续的几行。注释是写给人看的,而不是写给计算机的。当C编译器将源程序转换成机器能直接执行的目标代码时,注释被完全忽略。C++除了兼容C注释方法外,还提供一种新的注释,即某语句行从符号//开始至本行结束的所有字符均为注释内容。

2. 预处理命令

一般来说,每个C程序的开始部分都会有几行是编译预处理命令。例如,

```
#include<stdio.h>
```

是常见的编译预处理命令,它说明该程序使用了由ANSI C提供的标准库文件stdio.h。

预处理命令并非C语句,是在C源程序编译之前由预处理程序处理的命令,预处理命令以字符#开头,有关编译预处理命令的用法请参见第7章。

3. 程序级定义

在#include命令之后,大多数的程序都会包含对整个程序有效的一些定义,如符号常量定义、数据类型定义、全局变量定义等。在本程序只包含了下面一行:

```
#define PI 3.14159
```

它虽然也属编译预处理命令,但C可用它来定义符号常量,经过定义,PI就代表3.14159。

4. 函数原型声明

在C程序中，所有计算都是由相应函数完成的。函数是能够完成一定操作、具有具体名称的一组语句。在本程序中包含了两个函数：main和volume。下面一行代码：

```
float volume(float r,float h);
```

是函数原型声明，它使得C编译器能够检查函数调用是否和相应的函数定义一致，从而能检查代码中的错误。

5. main函数

每个完整的C程序，无论其功能大小，都是由一个或多个函数组成的，而这些函数可以是系统提供的也可以是用户自定义的，但其中总有一个且仅有一个被称为主函数的特殊函数，即main()。该函数作为程序执行的起点，而不论main函数在整个程序中的位置如何。在主函数main()中，通常会调用其他函数来完成某些具体工作，被调用的函数可以是由编程员自己编写的，如本例中的函数volume；也可以是来自于函数库，如本例中的scanf和printf。一般较大的程序都由多个函数组成，使程序结构清晰和管理方便，通常将一些关系密切的函数组织在一起放在同一个文件中，因此，C程序通常由多个源文件组成，并且每个源文件可以单独编译。

6. 用户自定义函数

由于较大的程序理解和修改都不方便，因此大部分的程序都被分成若干个函数。每个函数完成相对独立的功能，作为程序的一个模块，函数由一组相关的语句组成。在本例中，函数volume()是一个用户自定义函数，其功能是计算一个半径为r，高为h的圆柱体的体积。主函数通过函数调用语句“vol=volume(radius , height);”调用它。

1.5 算法

算法泛指解决某一个问题的方法和步骤。事实上，做任何事情都有一定的方法和步骤，例如洗手，你要首先打开水龙头，把手淋湿，然后用肥皂抹手，最后再伸手冲水把肥皂洗净，并关上水龙头。这些步骤都是按一定的顺序进行的，缺一不可，次序错了也不行。因此，从事各种工作和活动，都必须事先设计好步骤，然后按部就班地进行，才能避免产生错乱。算法是一种解决问题的策略，是人们对问题进行分析和抽象的结果。无论是数学公式，还是计算机程序，都属于算法的具体表现。但算法不等于程序，其含义和范围更广。算法是程序设计的灵魂。不了解算法就谈不上程序设计，因此对于程序设计人员来说，必须会设计算法，并且能根据算法编写程序。需要说明的是，不要认为只有“计算”的问题才有算法。

由于现实问题的多样性，因此算法在形式上也是多种多样的，那么不同的算法在结构上有没有一些规律？美国学者I. Nassit和B. Schneiderman对这个问题进行了研究，他们发现，所有问题的算法都可以用三种基本结构表示：