高等院校计算机应用技术规划教材

C语言程序设计

免费提供电子教案
下载网址 http://www.cmpedu.com

张宝森　等编著

机械工业出版社
CHINA MACHINE PRESS

高等院校计算机应用技术规划教材

C语言程序设计

张宝森　等编著

机械工业出版社

作为普通高等院校的程序设计基础教材，本书针对学生的特点和认知规律，按照“从问题到方法，再到程序”的编写思路，打破按语法知识体系结构组织教材内容的传统方法，由浅入深，循序渐进，全面、系统地介绍了C语言程序设计及其应用知识。

全书主要涉及了程序设计基础，顺序、分支和循环结构的程序设计，函数，数组，指针，结构体、文件与编译预处理命令等，并在相应的章节论述了程序运行模式，逻辑意图的表达，迭代与穷举的基本算法，复杂问题、任务的分解，现实问题描述与处理，数据的保存和重建等基本和必要的程序设计技术，尤其对指针的概念进行了深入本质的论述。

本书通过100多个例题，介绍了设计程序所经历的过程，注重对学生思维的训练和编程经验的积累，培养其应用能力。本书各章都配有习题，附录中提供了常用的资料。

本书可作为大学本科或专科计算机和与非计算机相关专业的“C语言程序设计”教材，也可作为计算机软件开发者的入门书籍。

图书在版编目（CIP）数据

C语言程序设计/张宝森等编著. —北京：机械工业出版社，2009.5
（高等院校计算机应用技术规划教材）
ISBN 978-7-111-26968-7

Ⅰ. C… Ⅱ. 张… 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（2009）第065588号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
策划编辑：赵 轩
责任编辑：赵 轩
责任印制：洪汉军

北京外文印刷厂印刷

2009年7月第1版·第1次印刷
184mm×260mm·18.5印张·456千字
0001—3000册
标准书号：ISBN 978-7-111-26968-7
定价：32.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294 68993821
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

出 版 说 明

随着国民经济的需求和教育事业的发展，计算机基础教育得到了很大程度的普及。在大学基础课程中开设面向应用的计算机课程对优化大学生的知识结构，提高综合素质起到了非常重要的作用。

为了满足大学各专业对计算机基础教学的需求，我社出版了“高等院校计算机应用技术规划教材”。本系列教材以计算机应用为主线，在突出实用性的同时也兼顾知识结构的完整性。教材具有以下特色：

一、服务于计算机基础教育课程体系建设

当前高校中，如何能够使学生打下坚实的计算机应用技术的基础，培养学生具有把计算机技术与本专业技术相结合，开发新技术的能力已成为教学的基本目标。根据这个目标，大多数院校在计算机基础教育方面已经形成或正在形成计算机基础教育的课程体系，使得学生在整个大学学习期间能够得到必要的、较全面的计算机应用教育。

为了支持和服务于大学计算机应用性基础教育课程体系建设，本系列教材及其内容充分吸收了教育部2006年颁布的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求（试行）》和全国计算机基础教育研究会发布的“中国高等院校计算机基础教育课程体系2008”等意见和研究成果。我社在聘请高校相关课程的主讲教师进行了深入、广泛地调研和论证工作之后，出版了本套系列教材。

二、尽量满足不同类型学校在不同教学阶段的需求

本系列教材涵盖计算机应用方面的各主要知识。每个方面的教材又有不同的难度和知识重点，供各高校根据课程体系的需要，在整个大学的学习期间选用。

1．计算机基础知识方面，出版《大学计算机应用基础》、《大学计算机基础实践教程》等教材，分别以基础知识、实践能力和技术应用为重点组织教学。

2．数据库应用方面，主要以Visual FoxPro、Access和SQL Server数据库的应用为主，在讲解数据库基本知识的基础上，以数据库应用案例为依托，通过案例教学的方式组织教学。

3．程序设计方面，主要以Visual Basic、C和C ++语言程序设计为主，为了配合每种语言程序设计的教学，同时出版相应的实验指导、习题集等配套教材，以适合不同类型学校、不同专业对程序设计方法学习和训练的需求。

4．网络和多媒体技术方面的教材以实用为主，学习如何有效和安全地获取和处理数字（数值）或模拟信息。引导学生从多方面获取知识，交流信息。

5．针对一些理工科专业和计算机高级应用教学的需求，本系列教材还包括《微型计算机原理与应用》、《微机接口及应用》和《嵌入式系统原理及应用》等。教材内容对于高校高年级学生，实际又实用。学生通过学习和实习后，完全可以结合自己专业，设计出具有一定应用价值的软硬件。

三、按照教学规律组织教材内容

本系列教材按照分析问题、找出问题的解决方法，总结提高到理论的认知过程，进行了精心地编写。聘请的所有作者都是活跃在教学第一线的、有多年教学经验的教师。作者根据教育部的要求，结合自己的教学经验，在教材中按照教学规律安排教学内容和层次，做到叙述精炼、图文并茂、案例适当、习题丰富，非常适合各类普通高等院校、高等职业院校使用，也可以作为培训教材或自学参考书。

我社将根据教学过程中师生的反映和计算机应用技术的发展情况，不断调整内容，改进写作方法，使本系列教材成为受广大师生欢迎的精品教材。

机械工业出版社

前　言

“有 3 个苹果放在桌子上，找出最大的一个，怎么找呢？”一些学生会不以为然。“如果有十几个苹果，并蒙上眼睛，找出其中最大的一个，又怎么找呢？”学生们会感到茫然。“如果编写程序让计算机完成这项任务，程序怎样编写呢？”学生们会不知所措。

上述第 1 个问题，反映了人对人与计算机在处理问题方法的认识上存在的模糊性，人“一眼”就能看出最大的，而计算机要两两比较。似乎人与计算机处理问题的方法存在差异。实际上，人“一眼看出”是飞快地进行了两两比较，这种“飞快”的程度已经让人感觉不出或意识不到所用的方法了。

第 2 个问题，反映了人处理问题的方法是否有条理性。尤其是“当把眼睛蒙上”时，更需要有一个条理化的方法。有了条理化的方法，人和计算机处理问题就没有什么不同了。所以程序设计在于找到解决问题的办法（算法），而编程能力在于把这个算法条理化，描述成适合计算机的操作方式。

第 3 个问题，反映了解决问题的编程方法——用计算机语言来表述解决问题的步骤。

因此笔者认为，计算机程序设计课程的最终任务是在借鉴大量编程经验的基础上，引导学生构建对问题的解决方案，并用程序设计语言表述出来。虽然人们会认为该课程的另一个任务是涉及程序语言本身的知识，但是笔者认为，程序语言本身的知识也是为设计程序服务的，完全可以通过实际的应用而体现出来。

本着上述这样的指导思想，本书从实际问题的应用出发，先构思解题的方案，再给出程序。书中例题都是经过精选的，一方面介绍典型的编程经验，另一方面引导学生构思解决方案。

全书共分为 9 章，第 1～3 章介绍程序设计基本概念、过程和 C 语言的基本语句与实际应用。第 4～6 章围绕算法的设计与实现，展开函数、数组与指针的内容讲解。第 7～9 章针对客观事物的描述（表示）、存储与处理，述及了链表、文件和开发程序所用的程序结构。

笔者认为，程序设计语言对开发程序来说，它是一个整体，不应当将各部分割裂开来；学习的过程是一个渐进积累的过程，要反复实践才能熟练。因此本书打破按知识体系组织教材的做法，将结构体、指针概念等以问题的实际要求方式提到各章中，意图是从应用中学，先用起来，有感性认识，再给出系统的知识介绍，达到反复学习与实践的目的。教师在备课时，应当注意到这点，并建议补充相关的习题或实验。

本书第 1、4 章由李智编写，第 2 章由尉林明编写，第 3、5、6、7 章由张宝森编写，第 8、9 章由周海燕编写，全书由张宝森统稿。此外，上述每位老师对全书各章均参与了讨论、修改和校对，尤其是周海燕副教授通审了全稿，提出了大量的宝贵建议。

虽然作者竭力将十几年的教学经验融汇于本书，但由于水平有限，书中难免出现不足之处，希望读者能够不吝时间与精力，给予批评和指正，在此先表示衷心感谢。

编　者

目　录

第1章　简单的C语言程序

程序设计是计算机软件开发的基础。掌握程序设计可以深入了解计算机的工作过程，可以更高效、可靠、安全地直接指挥计算机工作。C 语言是高级程序设计语言之一，它概念丰富，功能强大，应用广泛。C 语言程序设计涵盖了大量计算机工作原理性的知识，是任何有志从事软件开发人员的必修课。

知识点

1）程序设计和程序语言的基本知识。

2）算法的基本概念。

3）简单的 C 语言程序结构。

4）C 语言的基本语法规则。

5）顺序结构的特点。

6）字符串的输出。

1.1　程序设计和程序设计语言

1.1.1　程序设计的基本概念

在生活中，“程序”即所需完成的工作，按时间先后安排的工作步骤。

在人们的生活中处处事事都离不开“程序”。例如，一般情况下，每个人的日常活动总是遵循着这样一个步骤：

1）早上起床，洗漱，吃早饭，然后进行上午时间段的各项活动。

2）中午吃午饭，休息片刻，进行下午时间段的各项活动。

3）晚上吃晚饭，进行一些晚间的活动，晚间活动后睡觉。

这些步骤就是人们日常生活的“程序”。实际上，做什么事情都有一定的程序，大家可以举出很多类似的例子。人们对熟悉的事情可能在不经意的过程中就一步一步地完成了，但是有些不熟悉的事情，像航天器的发射、奥运会举办过程的细节，却需要人们事先制订计划，研究方案，进行详细的设计，才能很好地完成。

在计算机里，“程序”两字的含义就是为了让计算机能够自动完成各项任务，而事先准备的指令序列。

计算机的一个程序要完成某个计算任务，必须对这个任务进行描述。描述应包括两方面的内容，一个是对客观“对象”的描述，称为“数据结构”；另一个是对“对象”施加的“操作”的描述，称为“算法”。

譬如要计算圆柱体的体积，必须先让计算机得到圆柱体的底面半径和高度，这两个数据是用程序中的“变量”存储的，计算机访问这两个变量，就可以得到圆柱体的底面半径和高度。这两个变量就是对客观事物“圆柱体”的描述，它们是两个独立的变量，这就是它们的数据

结构。

有了底面半径和高度两个变量，可以安排计算圆柱体体积的公式，先计算底面积，再计算体积。这些指令的次序反映出计算的步骤，也就是“算法”。可见，算法就是对数据施加的操作。

定义数据变量，写出对变量的操作步骤，就是程序设计的工作过程。其间所使用的工具就是“程序设计语言”。

1.1.2 可执行程序的形成与程序运行过程

程序设计语言的发展经历了机器语言、汇编语言和高级语言 3 个阶段，高级语言又分为面向过程的语言和面向对象的语言。虽然现在面向对象的语言渐渐成为程序设计的主流，但它仍然以面向过程的语言作为基础语言。

计算机只能识别和执行二进制的指令代码，所谓“机器语言”就是计算机指令代码的集合，计算机可以直接运行用机器语言编写的程序。而用汇编语言和高级语言书写的程序，从外观上看，属于人类文字符号的排列，计算机不能直接运行它，人们将这种程序称为“源程序”。

“高级语言”是一种接近人类自然语言（英文）的符号体系，这种语言有它的语法规则，语义含义。这样，既可让编程人员能够非常容易地书写程序，便于阅读和相互交流，又可让计算机能够有效、准确地处理它。

源程序必须经“翻译”将它转化为机器的指令序列，计算机才能够执行，“翻译”的工作是用软件完成的。这类软件按其工作方式分为两种：一种是解释程序；另一种是编译程序。

1．解释程序的工作方式

解释程序的工作方式如图 1-1 所示。

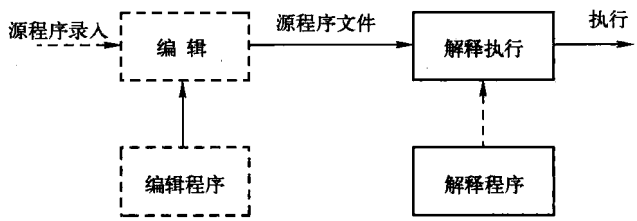

图 1-1　解释程序的工作方式

解释程序先将编辑的源程序调入自己的工作区，然后逐行翻译，每翻译一行源程序，就执行翻译后的二进制指令代码，即翻译一行，执行一行，直至程序结束或出现错误。翻译得到的二进制程序代码并不保存。这种方式，非常类似于外语翻译中的“口译”方式，说一句，翻译一句，并没有形成译文。

2．编译程序的工作方式

编译程序的工作方式如图 1-2 所示。系统对 C 语言程序采用的处理方式就是编译方式。

用文本编辑程序编辑成的源程序文件（文件的扩展名为.c），在编译时由“编译”程序调入内存，进行语法检查，如果检查出错误，停止编译并给出出错提示，由编程人员修改后再进行编译。如果没有错误，则将源程序翻译成机器指令码，形成目标文件（文件的扩展名为.obj）；然后经“连接”程序，将目标文件与系统提供的函数库及其他目标程序连接、装配成一个完整的可执行程序文件（文件的扩展名为.exe）。这是编译程序工作的最终结果。

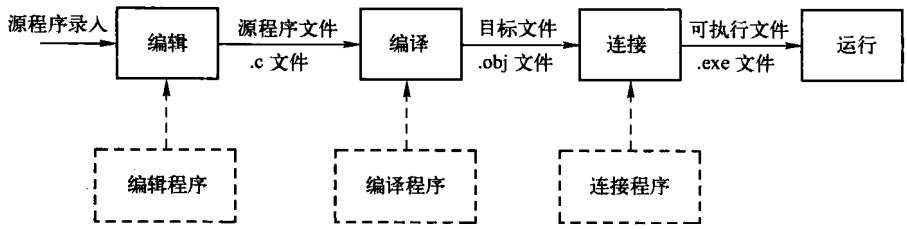

图 1-2　编译执行方式

运行可执行程序文件，就等于运行了高级语言程序。

编译程序的这种方式极其类似于外语翻译中的“笔译”，只有在修改定稿后才形成最终的译文。

为了方便程序的设计与开发，很多公司和机构都将编辑程序、编译程序和连接程序集成在一起，形成一个集成的软件编译环境，如早期的 Turbo C 和当今流行的 VC++。

3．程序的运行过程

计算机运行程序的过程都是按照“存储程序控制”原理进行的。这一原理是 1946 年由美籍匈牙利数学家约翰·冯·诺依曼（John von Neuman）提出的，所以又称为“冯·诺依曼原理”。这一原理在计算机的发展过程中，始终具有重要影响，确立了现代计算机的基本组成和工作方式。如今计算机已经发展到了第四代、第五代，但依然遵循着这个原理。

“存储程序控制”原理的要点：程序的数据和指令均采用二进制形式表示，程序事先被输入到计算机的内存储器中存储（存储原理），并使指令指针 IP 指向指令代码开始的地址。在运行时，控制器按照指令指针 IP 的地址顺序取出存放在内存储器中的指令（即按地址顺序访问指令），然后分析指令，执行指令的功能，遇到转移指令时，则转移到相应地址，再按转移后的地址接着顺序执行指令（程序控制）。经过这样一步步地操作，计算结果便逐步被计算出来。

对高级语言编写的程序来说，源程序在被编译、连接成可执行程序后，它的运行依然遵循着“存储程序控制”原理，只是由于高级语言程序结构复杂，往往一条代码翻译成机器语言后，会形成许多条机器语言代码，“存储程序控制”显得不那么直接而已。

图 1-3 是一个 C 语言程序运行过程的示意。其中，图 1-3a 是一个 C 语言程序，图 1-3b 是程序中的 4 个变量。其第一排的 4 个方框分别表示变量在内存中的存储单元，初始时它们的数值是不确定的（因此未写出）。下面各排的方框是在程序执行某条语句后，这 4 个变量的变化情况。

为了理解高级语言的运行过程，读者应当先建立“**当前的执行点**”的概念。因为数字计算机是以时钟脉冲为节拍一步一步工作的，反映到高级语言程序中语句是一条一条地执行的，每条语句的执行是有先后次序的，一条语句执行完，自动接着执行下一条。所谓“当前的执行点”就是正在执行的语句。

对于图 1-3a 的程序来说，程序一开始的“当前的执行点”是主函数名 main，这一条的执行任务是由内部代码完成的，编程人员可以不予理睬，然后“当前的执行点”改变到“a=5;”（“=”为赋值，这条语句上，所跳过的变量定义也是由系统内部完成。a=5;语句执行完后，“当前的执行点”又改变到“b=6”；语句上……直至执行完最后一条语句，其后的收尾工作也是由系统内部完成的。

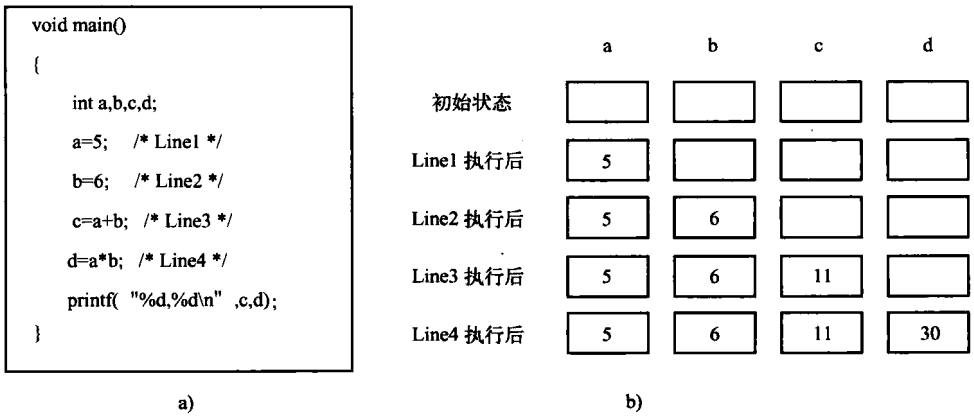

图 1-3　高级语言程序运行过程示意

a) 一个 C 程序　b) 内存中变量随程序的运行而变化的情况

从上面的过程可以看出，程序是一步一步地顺序向下执行的，一次完成一个操作。这是程序的基本运行模式。在以后的各章中还可以看到其他的运行模式，但是顺序执行是最基本的。

对于用高级语言设计的程序来说，变量在内存的分布和指令代码放在内存的位置，都是由编译系统程序完成的，编程人员不必关心这些细节。这也就是人们为什么发明了高级语言，乐于用高级语言编写程序的缘故。但是，学会为变量画图，养成研究变量的变化的习惯，可以加快学习程序设计语言的步伐，迅速提高程序设计的能力。

4．如何实现程序设计

前面已经叙述了程序的概念和计算机运行程序的过程，那么如何设计一个程序呢？

由于现实世界中问题的复杂程度不同，因此程序的规模可大可小，小到只包含一条语句，大到需有多人共同完成。一般来说，程序的设计要经过以下几个步骤：

1）确定解决问题的方案。针对一项编程任务，要进行对问题的分析和归纳，包括：任务都有哪些要求？需要得到怎样的结果？需要提供怎样的数据作为输入？系统能够提供哪些工具或已有程序？解决任务采用怎样的数据结构？程序的模块划分又是怎样的？每个模块的算法是什么？

其中最为重要的是数据结构和算法。数据结构是用数据的形式描述客观事物，解决客观事物在计算机内存储的问题。没有描述、存储，编写程序代码就无从谈起。算法是对表示客观事物的数据的操作，好的算法可以高效、高速处理问题，得到正确的结果。

2）确定解决问题的算法。根据程序模块的要求和结果，以及确定的数据结构，设计具体的解决问题的方法和步骤。

3）编程和调试。根据确定的算法，选用一种较为适宜的计算机语言，编写出源程序文件，然后上机运行调试，在运行过程中发现编译或运行错误，应及时改正，确保程序能够运行。

4）程序测试。程序能够运行后，还要测试一下程序运行中对全部合法的数据都应有正确的结果，对非法的操作和数据应给出相应的提示，以确保程序在运行中不会出现错误的结果，

或造成程序运行中断，或导致系统运行瘫痪的问题。

5）编写整理技术文档。为保障应用程序的正确使用，便于维护和修改，应由程序设计人员提供程序使用说明书、技术说明书等文档。

1.1.3 算法

1. 什么是算法

“算法”是指为解决某个具体问题而采取的方法和步骤，程序代码所体现的操作步骤就是算法的具体实现。因此，确定“算法”是编写程序代码前的一个非常重要而又关键的步骤。

计算机解决问题的算法归纳起来分为两大类：数值计算算法和非数值处理算法。

数值计算算法所要解决的问题是数值求解的问题，如求方程的根，求定积分的结果等。这些算法都已比较成熟，学习的重点是理解和掌握它。而非数值运算的算法涉及的内容非常广泛，常见的是信息管理类的，如人事、学籍和图书管理等。由于涉及的面比较多，难于规范化。其中某些典型的应用有比较成熟的算法，如排序、检索等。但有相当一部分非数值运算的问题，需要设计者参考已有的类似算法，针对具体问题重新设计专门的算法。

2. 算法应具备的特性

即使是同一个问题，不同的设计者可能设计出不同的方法和步骤，那么如何评价和衡量一个算法是否正确呢？通常可以从以下几个方面来考虑。

1）有穷性：一个算必须包含有限个操作步骤，其中每一步都应在合理的时间范围内完成。这个合理的限度没有严格的标准，要具体问题具体分析。

2）确定性：算法中的每个步骤都必须是确定的。不能出现模棱两可的二义性，“怎么做都行”是绝对不允许的。

3）有效性：算法中的每个步骤都应该是能够执行的。比如，在分数运算中，要避免出现分母为 0 的情况；一旦出现，程序中应当留有采取措施的代码。

4）有输入或无输入：既然算法是处理数据的操作步骤，因此必须要有数据输入。通常数据是通过输入环节提供的，但某些情况下数据是固定的常量，此时可以没有输入环节。

5）有输出：数据处理完毕后，必然有结果输出，没有结果的算法是毫无意义的。

3. 怎样表示算法

描述算法的主要工具有 3 种：自然语言、伪代码和流程图。其中流程图又有两种，即传统流程图和 N-S 流程图。本节给出用自然语言和流程图描述算法的例子。至于伪代码实际就是自然语言和某种高级语言的混合使用，且不严格遵守高级语言的语法限制。

（1）自然语言描述法

【例 1-1】　需要解决的问题：求 n!　（1×2×3×…×n）（n≥0）。

【解题思路】　设置一个初值为 1、表示阶乘的变量 fac 和一个 1～n 逐一变化的变量 i，该变量表示乘数。如果随着每次 i 的变化，反复计算 fac*i，并且每次计算后再存入 fac，也就是反复进行两步计算：fac=fac*i，i 增 1，那么，变量 fac 得到的就是阶乘。