21世纪高职、高专计算机类教材系列

C语言程序设计教程

（第二版）

李志球　刘　昊　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

21 世纪高职、高专计算机类教材系列

C 语言程序设计教程

（第二版）

李志球　刘　昊　编著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书是《21 世纪高职、高专计算机类教材系列》之一，共 11 章。本书从先进性和实用性出发，较全面地介绍了 C 语言程序设计所涉及的基本理论、程序设计方法和实际应用技能。内容包括：程序设计和 C 语言基本概念、顺序结构、选择结构、循环结构程序设计方法，数组和字符串、函数和预处理、指针、结构体、共用体与位运算、文件、应用程序设计实例等。

本书叙述简明扼要，通俗易懂，实用性强，各章有小结，习题部分题型丰富。第 10 章可作为课程设计参考内容，第 11 章实验实训内容供学生实验时参考使用。华信教育网（http://hxedu.com.cn）提供了本书的电子教案和习题参考答案，供教师和学生下载。

本书可作为应用型本科院校、高职高专、成人高校及民办高校的计算机类和电子信息类各专业和其他专业的教材，也可作为有关技术人员自学参考用书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

C 语言程序设计教程/李志球，刘昊编著. —2 版. —北京：电子工业出版社，2007.7
（21 世纪高职、高专计算机类教材系列）
ISBN 978-7-121-04501-1

Ⅰ. C… Ⅱ. ①李… ②刘… Ⅲ. C 语言—程序设计—高等学校：技术学校—教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 098495 号

责任编辑：张 榕（zr@phei.com.cn）
印 刷：北京市顺义兴华印刷厂
装 订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1 092 1/16 印张：19.75 字数：505 千字
印 次：2007 年 7 月第 1 次印刷
印 数：4 000 册 定价：29.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

序　言

1. 缘起与背景

20 多年来，我国应用型高等教育、高等职业教育得到了长足的发展。在这一领域从事计算机教育的师生在教学改革和教学建设方面取得了很多成绩，有的还列为国家重点教学改革项目进行试点。1998 年 12 月 24 日教育部发布了《面向 21 世纪教育振兴行动计划》，提出“积极发展高等职业教育”。我国的高等职业教育进入了高速发展阶段，这一新形势向我们提出了新的更高要求。认真总结应用型高职、高专的教学教改经验，制定一套适合当前改革、发展要求的应用型高等教育（含高等职业教育）的计划、大纲和编写教材就成了当务之急，基于这样一个认识，我们组织了十余所学校的教师进行研讨，并组织编写这套 21 世纪高职、高专计算机类教材。

2. 编写原则

高职、高专有自身特色，正如《振兴计划》中指出的：“高等职业教育必须面向地区经济建设和社会发展，适应就业市场的实际需要，培养生产、服务、管理第一线需要的实用人才，真正办出特色。”培养出符合国家建设需要的高素质的应用型人才是高职、高专发展的根本目的。因此，在这套教材的编写中，我们遵循“适用、实用、会用、通用”的原则，避免低水平重复。

“适用”就是要符合目前行业要求的新知识、新技术、新方法。由于计算机技术始终处于高速发展中，因此，如果只讲那些已经“十分成熟”的技术，那么，学生毕业后，这些技术可能已经过时。这样培养出来的学生，不能适应职业岗位的需要。因此，本套教材在选材上，既注意讲透基本理论，也注意讲解新技能，具有一定的前瞻性。

“实用”就是计算机行业最广泛应用的知识、方法和技能，使学生能胜任岗位工作，切实符合社会需要。

“会用”就是培养学生在具备一定理论基础的前提下，能够用自己所学的知识，解决在工作中遇到的具体问题。注重动手能力和操作技能。

“通用”是指本套教材不仅限于高等职业教育，对于应用型高等院校，如技术学院、技术师范学院、职业大学等也是对口的教材。

3. 编写情况

本套教材的作者都是多年从事应用型高等教育和高等职业教育的教师，他们对应用型高等教育的实际、学生的学习情况、学生就业后面临的岗位要求等有深入的了解。在本套教材编写中，我们反复研讨，得到了许多学校领导和教师的大力支持，许多章节都是在优秀教案、讲义的基础上推敲而成的，吸收了计算机试点专业的教改经验，并由主编全文统稿。在此基础上，我们组织专家审阅、把关，以确保质量。今后还将根据我们这十余所学校的使用情况，认真听取读者的意见，不断修订、补充、完善，以跟上计算机行业发展的步伐。

4. 适用学校和专业

本套教材除了特别适合高等职业学校计算机类专业（包括“计算机应用”、“计算机网络”、“信息管理”、“计算机科学教育”、“会计电算化”等）使用外，也可供其他应用型高等专科学校使用。对那些迫切需要提高自己应用技能的读者，本套教材作为自学读物，亦颇为得当。

21 世纪高职、高专计算机类教材编委会

前　言

C 语言功能丰富，表达能力强，使用灵活方便，程序执行效率高，可移植性好；它既具有低级语言可直接操作计算机硬件的能力，又有高级语言的特点；它既可编写系统软件，又适合编写应用软件。因此，C 语言得到了广泛应用。

《C 语言程序设计教程》一书自出版以来，受到了广大读者的厚爱。但由于作者水平有限，加上计算机技术发展日新月异，本书在内容充实和章节选取上还需要进一步改进。根据《21 世纪高职、高专计算机类教材系列》的编写目的，遵循“适用、实用、会用和通用”的原则，并根据有关高校的使用情况，认真听取了读者的意见，对第一版教材进行修正、补充、调整和完善。

教材第二版对第一版各章中的错漏和不妥之处做了修正，调整了部分章节的顺序，增加了应用程序设计实例等章节，各章习题部分增加了一些练习题。教学实施时，各高校可根据自己的教学大纲和教学时数的要求，灵活组织教学内容。授课时教学时数不够时，可对本书内容进行选择，其中第 1～8 章是基础，为必选内容，各章的选学与提高和其余章节作为选学内容。

全书共分 11 章，第 1 章介绍了程序、程序设计、算法和流程图，C 语言实现方法、C 语言构成和 C 语言数据类型、基本运算符、表达式等基本概念。第 2 章介绍了 C 语言基本语句、常用的输入/输出函数、顺序结构程序设计方法、goto 语句和语句标号的使用，以及程序调试方法等。第 3 章介绍了关系、逻辑和条件运算符及相应的表达式、if 语句和 switch 语句及选择结构程序设计方法。第 4 章介绍了循环概念、三种循环语句及循环程序设计方法。第 5 章介绍了数组的概念及数组的使用，同时还介绍了一些常用的字符串处理函数。第 6 章介绍了 C 语言函数的基本格式、参数传递、调用方法，以及变量的存储类型、作用域、函数存储类别和预处理。第 7 章主要介绍了指针的概念、指针的定义和使用、指针作为函数参数、指针与数组、函数指针、指针数组。第 8 章介绍了结构体和共同体类型及变量定义、引用、枚举类型定义及变量引用、链表的操作、位运算。第 9 章主要介绍文件概念、文件类型指针、文件的打开、读/写、关闭等。第 10 章通过实例介绍了 C 语言应用程序的框架结构和设计方法。第 11 章介绍实验实训内容，供学生在上机操作时参考使用。

本书是《21 世纪高职、高专计算机类教材系列》教材之一，是编者根据多年教学经验及从事 C 语言开发的基础上编写而成的。编写过程中，力求理论与实践相结合，突出实用和实际操作，在语言叙述上注重概念清晰、逻辑性强、通俗易懂、便于自学；在体系结构上安排合理、重点突出、难点分散、便于掌握。

本书由李志球、刘昊编著，李志球编著第 1～5 章、第 11 章及全部附录，刘昊编著第 6～10 章，李志球对全书进行了统稿和定稿。赵丽松、张榕、邵晓根等同志在本书编写出版过程中给予了大力支持，并提出了宝贵意见，在此深表感谢！

本书可作为高职高专、本科院校、成人高校及民办高校的计算机类、电子信息类各专业和其他非计算机类专业的教材，也可作为有关技术人员的自学参考用书。

由于作者水平有限，书中的错漏之处在所难免，殷切希望广大读者提出宝贵意见，以使教材不断完善。为了帮助读者学习本书，华信教育网（http://hxedu.com.cn）提供了本书的电子教案和习题参考答案，供教师和学生下载。需要其他教学包的教师也可以和作者联系。作者 E-mail：lizq98@xzcat.edu.cn。

李志球、刘昊

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

试读结束，需要全本PDF请购买 www.ertongbook.com

第1章

程序设计和C语言基础

本章要点

- 了解程序和程序设计的基本概念
- 掌握算法和流程图的概念和作用
- 了解C语言的构成和特点
- 掌握C程序开发环境和编译、连接和运行步骤
- 掌握C语言整型、实型和字符型等基本数据类型
- 掌握标识符、常量、变量概念
- 初步掌握常用的运算符及表达式
- 初步掌握编译、连接错误程序调试方法

引导与自修

计算机高级语言种类很多，但是功能最强大、能被大多数程序员所认可的，还是 C 语言。C 语言虽是一种高级语言，但它也可以完成许多只有机器语言和汇编语言才能完成的面向硬件底层的工作。很多常用的软件系统，也是由 C 语言编写而成的。例如，Windows 系列、Linux 等操作系统中相当多的内容是由 C 语言编写的。

1.1　程序和程序设计

1.1.1　程序

计算机系统分为硬件系统和软件系统两大类。计算机裸机（没安装软件系统）只是一台机器，各种软件让计算机具有了处理具体问题的能力。例如，在计算机上安装了办公软件，就可以完成文字录入、排版、绘制表格等工作；安装了网络系统软件，人们就可以在网络上查询资料、聊天等。因此，如果把计算机比做一个人，那么计算机硬件系统是躯体，而软件系统是大脑，由大脑指挥躯体来完成各种工作。软件系统由程序和程序的相关文档（如说明书、源程序代码等）组成。

所谓程序是指为解决某一特定问题而预先编排好的处理方法和步骤的指令序列，程序由计算机语言编写而成。计算机所做的工作都是在程序的控制下完成的，当程序被执行时，就自动按指定的顺序执行这一条条的指令并完成相应的工作。

计算机语言分为机器语言、汇编语言和高级语言等种类。机器语言直接用二进制代码编写，它运行效率很高，但机器语言的编写、阅读、修改等工作相当困难；汇编语言是一种使用有助于记忆的符号来代表二进制代码的语言，它采用可以识别的符号表示数据和数据所在内存中地址的符号语言。机器语言和汇编语言都与具体的计算机硬件有关，不同的计算机所使用的语言也不同，互相之间不能通用，是不可移植的。

高级语言是一种和机器无关、表达形式接近人类自然语言和数学语言的计算机程序设计语言。几乎所有的高级语言所使用的词汇都是英语中常用的词汇或缩写，人们学习和操作起来十分方便。高级语言有上百种，如 BASIC、Pascal、Fortran、Java 和 C 语言等。

计算机只能识别处理由“0”和“1”构成的二进制代码指令或数据，而由汇编语言或高级语言编写的程序对计算机来说都是不可读的，它们都需要被翻译成二进制代码后才能被执行。翻译方式有“解释方式”和“编译方式”两种，解释方式类似于口语翻译（说一句翻译一句），它逐句取出源程序中的语句，翻译完一条命令立即被执行，然后再翻译下一条命令并执行。BASIC、Java 等语言属于这种方式。解释方式的优点是在调试程序时，能边执行边改错，人-机交互性好，缺点是因为逐句解释执行，速度较慢，而且源程序不能脱离解释程序单独执行。编译方式类似于整篇文章的笔译（整篇文章全部翻译好再交给读者），它先将源程序一次性翻译成机器语言（目标程序），然后再通过连接程序将多个目标程序和库程序连接装配成可执行程序。C、C++、Pascal 等大多数高级语言采取编译方式。编译方式的优点是编译连接好的可执行程序计算机可以直接运行，不再需要编译环境和连接程序就能独立运

行，缺点是每次修改源程序后都需要重新编译连接以产生新的可执行文件。

注意：除了解释方式和编译方式外，还有将这两种方式结合起来的方式，即先编译源程序，首先产生还不能直接执行的中间代码，然后通过解释程序解释并执行中间代码。FoxPro 就属于这种方式。这种方式比直接解释执行快，中间代码可以独立于计算机，执行程序时系统中只要有相应的解释程序，就可在任何计算机上运行。

通常把由高级语言编写的程序称为“源程序”，由源程序转换成相应的二进制代码程序称为“目标程序”。为了能使计算机处理高级语言源程序，需要把源程序转换成机器能够接收的目标程序，具有翻译功能的软件称为“编译程序”或“解释程序”。每种高级语言都有与它对应的翻译程序。

1.1.2　程序设计

1. 一般程序设计

程序设计（Programming）就是首先对需要解决的问题进行分析，用合理的数据结构描述问题中所涉及的对象，找出解决问题的算法，并将算法用计算机语言编写出来。因此，程序设计主要有三个方面的任务：一是数据结构的设计，二是算法设计，三是根据算法进行编码。数据结构和算法的设计是最难掌握又是最重要的两个方面，算法和数据结构紧密联系，具体的算法依赖于特定的数据结构，合理的数据结构又可有效地简化算法。因此，数据结构设计比算法设计更为重要。

程序设计过程主要包括以下几个步骤：

（1）确定数据结构。根据用户提出的要求、原始数据和输出结果形式，分析问题，设计合理的数据结构。

（2）确定算法。针对数据结构，确定解决问题、完成任务的步骤。

（3）编写程序。根据确定的数据结构和算法，使用计算机语言编写程序代码，输入到计算机中，简称编程。

（4）调试程序和测试程序。调试程序主要是检查程序的语法错误或逻辑错误，并输入各种可能的少量数据对程序进行测试，使它能得到正确结果，同时对非法数据进行容错处理。

如果程序有缺陷或严重错误，需要回到步骤（1），重新分析问题并设计算法。

（5）整理并写出文档资料。

2. 模块化程序设计

大多数应用程序一般都由许多人员共同协作开发完成，可采用“自顶向下，逐步求精”的模块化程序设计方法。即开始时并不涉及问题的实质和具体解决的步骤，而只是从问题的全局出发，给出一个概要的抽象描述。处理的方法是：将一个复杂的大任务分解为若干个较小的、容易处理的子任务，如果分解的子任务本身还很复杂，可以再细分为许多更小的子任务。一个子任务只完成一项简单的功能，这些子任务称为“模块”。程序员分别完成一个或多个小模块，这种程序设计方法称为模块化程序设计方法，由一个个功能模块构成的程序结构为模块化结构。模块化程序设计可将一个任务由一组人员同时进行分工编写，并分别调

试，最后将所有模块组装，这样可大大提高了程序设计的效率。

3. 面向过程和面向对象的程序设计

高级语言程序设计目前分为两大类：一类是以描述计算机解题过程为主的语言，称为面向过程的语言 POP（Process-Oriented Programming language），如 BASIC、C、Pascal 等。另一类则是以描述操作对象的特性和行为特征为主的语言，称为面向对象的语言 OOP（Object-Oriented Programming language），如 VB、C++等。

还有人提出了一种称为“面向问题的语言”，它实际上属于 OOP 的一种。面向问题的语言是为了便于描述并求解某个特定领域问题的一种非过程语言。面向问题的语言不仅不用过问计算机内部逻辑，也不关心问题的求解算法和求解的过程，只需要指出需要做什么、数据的输入和输出形式，就能得到所需结果。如报表语言、SQL（Structured Query Language）语言（SQL 语言是数据库查询和操作语言，能直接使用数据库管理系统）等。面向问题的语言能方便用户的使用，提高程序的开发速度。

4. C、C++和 VC++语言

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 语言做进一步发展和扩充，制定了新的标准，称为 ANSI C。目前，使用较多的 C 语言版本有 Microsoft C（或称 MS C）、Borland Turbo C（或称 Turbo C）和 AT&TC 等几种版本，它们不仅实现了 ANSI C 标准的全部功能，而且各自还做了一些扩充，使之更加方便、完善。

在 C 语言的基础上，1983 年贝尔实验室的 Bjarne Strou-strup 推出了面向对象的 C++语言，它进一步扩充和完善了 C 语言。C++提出了一些更为深入的概念，如支持面向对象的概念，容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统的结构化程序设计所不同的思维方式和编程方法，由此也增加了 C 语言的复杂度，学习起来有一定的难度。

C++目前有 Borland C++，Symantec C++和 Microsoft Visual C++（VC++）等版本。VC++是实现 C++语言的一种具体编译、连接和执行 C++程序的环境。

为什么不直接学习面向对象的 C++语言，而要学习 C 语言呢？

（1）C 语言是 C++语言的子集，C++语言包含了 C 语言的全部内容，C 语言本身运算符就很多，优先级也较繁杂。但 C++语言在 C 语言的基础上扩充了很多内容，运算符等需记忆的内容更多，这也使得学习 C++要比学习 C 语言困难得多，所以不太适合程序设计的初学者。

（2）C 语言是面向过程的，C++语言是面向对象的语言，而面向对象的基础是面向过程。C++用于开发大型应用软件，但在实际应用中，并非所有问题都需要编成大型软件。

（3）C 语言是 C++的基础，因此，掌握了 C 语言，再进一步学习 C++这种面向对象的语言，会达到事半功倍的效果。

C 语言的细节较难掌握，程序调试也较为麻烦。有时一个很小的程序，仔细核查后也没发现错误，但程序运行时却出错，主要是由于 C 语言的语法检查不够严格。因此，初学者学习时感觉较为困难。但它能培养初学者程序设计能力，也是“数据结构”、“操作系统”等课程的先导课，因此，学好 C 语言是必需的。

精讲与必读

1.2　算法和流程图

1.2.1　算法

1．算法（Algorithm）概述

在人们的日常生活和生产活动中也存在算法，都在自觉或不自觉地使用算法。例如，对一天生活和工作的安排：几点起床、几点出门，走哪条路、坐几路车、如果堵车怎么办、一天具体做哪些工作等，这些都需要提前做好计划，这就是一个最简单的算法。

算法描述了一个待处理问题需要“做什么处理”和“如何处理”两个问题，也就是“做什么”和“怎么做”。“做什么”是指对问题的认识、分析和判断，“怎么做”则是要找出具体的、正确的和有效的解决问题的方法和步骤。算法是一组有穷的规则，它们规定了解决某一特定类型问题的一系列运算，是对解题方案的准确和完整的描述。

程序设计中的算法是指使用计算机完成一个任务所采取的方法和执行的步骤，一般包含给定初始状态或输入数据，经过计算机的有限次运算，最后得出所要求或期望的终止状态或输出数据等内容。设计好了算法，就可以用具体的语言进行编写，最终转化为解决问题的程序。当然，方法和步骤的正确性与有效性是相对的。

对于一个给定的可计算（可解）的问题，可以设计多种算法，不同的人可以编写出不同的程序。算法的优劣可以用空间复杂度与时间复杂度来衡量。

2．算法的特性

算法主要有以下几个特性：

1）有限性

一个算法必须在有限的步骤内完成，完成这些步骤也应该在一个合理的时间内。

2）可行性

可行性又称有效性，算法中的操作都可以通过已实现的基本运算执行有限的次数来实现。

3）确定性

算法中的语句必须有确切的含义，不能有二义性，相同的输入必须有相同的结果。

4）输入

一个算法必须有零个或多个输入。

5）输出

一个算法应有一个或多个输出。算法的最终目的是得到结果，如果没有输出结果，那么算法就失去其目的性，不能称为算法。

3．算法的复杂性

算法的复杂性可以用空间复杂度与时间复杂度来衡量，复杂度是算法效率和性能的度量。时间复杂度是指完成算法所需的时间，空间复杂度是指完成算法所需占用的计算机内存

空间大小。

对于一个实际问题，设计出尽可能简单的算法是在设计算法时考虑的一个重要目标。当给定的问题存在多种算法供选择时，选择复杂性最低者，是选用算法应遵循的一个重要准则。

4. 算法的描述

算法的描述有很多种方法，主要有以下几种：

（1）自然语言；

（2）流程图，如N-S图、流程图，图的描述与算法语言的描述对应；

（3）算法语言，如计算机语言、程序设计语言、伪代码；

（4）形式语言，用数学的方法，可以避免自然语言的二义性。

算法的描述方法虽然不同，但作用是一样的。

【例1.1】 使用自然语言描述求解5!（5的阶乘）的算法。

求解5！的算法如下：

第一步，计算1×2，结果为2；

第二步，计算2×3，结果为6；

第三步，计算6×4，结果为24；

第四步，计算24×5，结果为120；

算法结束，即5! =120。

使用自然语言描述算法通俗易懂，但比较烦琐，也不能直观地描述条件转向等复杂问题。对于初学者，可以使用流程图、N-S流程图等方法来描述算法。

1.2.2 结构化程序设计和流程图

1. 结构化程序设计

结构化程序设计（Structured Programming）由荷兰学者E. W.Dijkctra提出，它指出了编程时必须遵守的一些原则，目的在于保证程序的可读性，便于程序推广应用和交流。它的显著特征是代码和数据的分离，分离的一个方法是调用包含局部变量的子程序。这种包含局部变量的子程序，对程序的其他部分没有副作用，这使得编写共享代码段的程序变得十分简单。调用函数时只需知道函数功能是什么，而不必知道它是如何实现的。结构化程序设计语言比非结构化语言更清晰，结构更合理，程序设计和程序维护更为容易。

1）结构化程序设计原则

（1）自顶向下，逐步细化，实现模块化

“自顶向下，逐步细化”就是前面介绍的模块化程序设计方法。这一原则是指将任务或问题划分为若干个相对独立的模块，而每个模块又可细分成若干个子模块，如此从上而下，逐步细化。然后再考虑每个模块中使用的具体函数和语句。

（2）清晰第一、效率第二

清晰第一、效率第二是从提高程序的可读性、方便交流、调试、修改和维护的角度出发而提出的。程序只有在结构清晰的基础上，再去考虑它的效率。

试读结束，需要全本PDF请购买 www.ertongbook.com