

全国高校出版社优秀畅销书特等奖

中国高等院校计算机基础教育课程体系规划教材
丛书主编 谭浩强

C程序设计(第四版)

谭浩强 著

发行逾1100万册



清华大学出版社

全国高校出版社优秀畅销书特等奖

中国高等院校计算机基础教育课程体系规划教材

丛书主编 谭浩强

C 程序设计 (第四版)

谭浩强 著

发行逾1100万册



清华大学出版社
北京

内 容 简 介

由谭浩强教授著、清华大学出版社出版的《C 程序设计》是一本公认的学习 C 语言程序设计的经典教材。根据 C 语言的发展和计算机教学的需要,作者在《C 程序设计(第三版)》的基础上进行了修订。本书按照 C 语言的新标准 C 99 进行介绍,所有程序都符合 C 99 的规定,使编写程序更加规范;对 C 语言和程序设计的基本概念和要点讲解透彻,全面而深入,按照作者提出的“提出问题—解决问题—归纳分析”三部曲进行教学、组织教材;本书的每个例题都按以下几个步骤展开:提出任务—解题思路—编写程序—运行程序—程序分析—有关说明。符合读者认知规律,容易入门与提高。

本书内容先进,体系合理,概念清晰,讲解详尽,降低台阶,分散难点,例题丰富,深入浅出,文字流畅,通俗易懂,是初学者学习 C 程序设计的理想教材,可作为高等学校各专业的正式教材,也是一本自学的好教材。本书还配有辅助教材《C 程序设计(第四版)学习辅导》。

本书扉页为防伪页,封面贴有清华大学出版社防伪标签,无上述标识者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计 / 谭浩强著. —4 版. —北京:清华大学出版社, 2010.6

(中国高等院校计算机基础教育课程体系规划教材)

ISBN 978-7-302-22446-4

I. ①C… II. ①谭… III. ①C 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(2010)第 066130 号

责任编辑:张 民

责任校对:焦丽丽

责任印制:李红英

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, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185×260 印 张:26 插 页:1 字 数:626 千字

版 次:2010 年 6 月第 4 版 印 次:2010 年 7 月第 2 次印刷

印 数:10001~60000

定 价:29.00 元

产品编号:031938-01

教授计算技术的大师

普及现代科技之巨擘

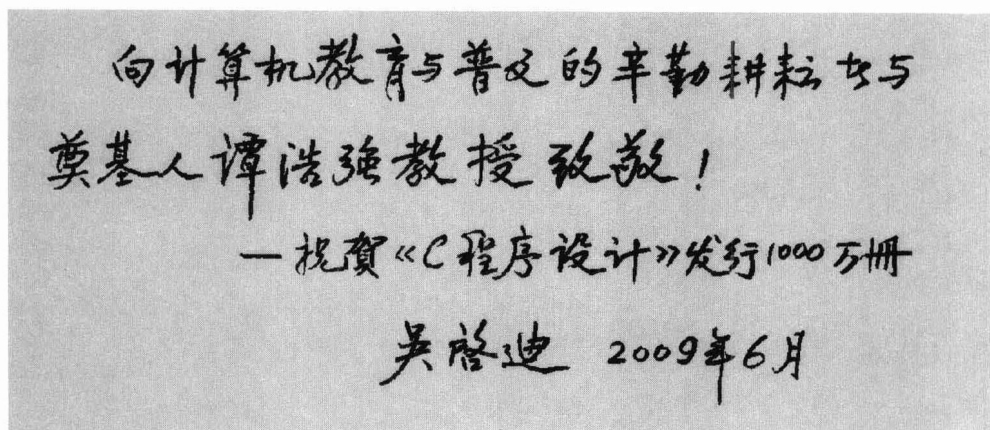
敬颂谭浩强教授创杰垂成就

宋健

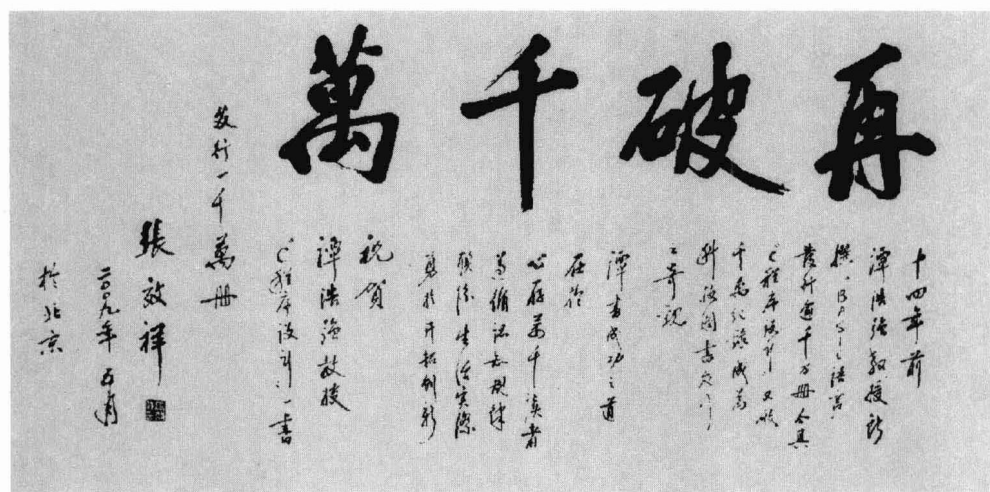
一九九五年一月

▲ 原全国政协副主席、国务委员、国家科委主任、
中国工程院院长宋健同志给谭浩强教授的题词

祝贺 《C程序设计》 发行1000万册



▲ 全国人大常委会、教育部原副部长吴启迪题词



▲ 我国计算机事业的开创者、中国科学院资深院士、中国计算机学会名誉理事长张效祥题词：“再破千万——十四年前谭浩强教授所撰《BASIC语言》发行逾千万册，今其《C程序设计》又破千万纪录，成为科技图书发行之奇观。谭书成功之道在于：心存万千读者，遵循认知规律，联系生活实际，勇于开拓创新。祝贺谭浩强教授《C程序设计》一书发行一千万册”。

中国高等院校计算机基础教育课程体系规划教材

编审委员会

主 任：谭浩强

委 员：(按姓氏笔画为序)

王路江	冯博琴	曲建民	刘瑞挺	杨小平
吴文虎	吴功宜	李凤霞	张 森	汪 蕙
高 林	黄心渊	龚沛曾	焦 虹	詹国华

策划编辑：张 民

从 20 世纪 70 年代末、80 年代初开始,我国的高等院校开始面向各个专业的全体大学生开展计算机教育。面向非计算机专业学生的计算机基础教育,牵涉的专业面广、人数众多,影响深远,它将直接影响我国各行各业、各个领域计算机应用的发展水平。这是一项意义重大而且大有可为的工作,应该引起各方面的充分重视。

20 多年来,全国高等院校计算机基础教育研究会和全国高校从事计算机基础教育的老师始终不渝地在这片未被开垦的土地上辛勤工作,深入探索,努力开拓,积累了丰富的经验,初步形成了一套行之有效的课程体系和教学理念。20 年来高等院校计算机基础教育的发展经历了 3 个阶段:20 世纪 80 年代是初创阶段,带有扫盲的性质,多数学校只开设一门入门课程;20 世纪 90 年代是规范阶段,在全国范围内形成了按 3 个层次进行教学的课程体系,教学的广度和深度都有所发展;进入 21 世纪,开始了深化提高的第 3 阶段,需要在原有基础上再上一个新台阶。

在计算机基础教育的新阶段,要充分认识到计算机基础教育面临的挑战。

(1) 在世界范围内信息技术以空前的速度迅猛发展,新的技术和新的方法层出不穷,要求高等院校计算机基础教育必须跟上信息技术发展的潮流,大力更新教学内容,用信息技术的新成就武装当今的大学生。

(2) 我国国民经济现在处于持续快速稳定发展阶段,需要大力发展信息产业,加快经济与社会信息化的进程,这就迫切需要大批既熟悉本领域业务,又能熟练使用计算机,并能将信息技术应用于本领域的新型专门人才。因此需要大力提高高校计算机基础教育的水平,培养出数以百万计的计算机应用人才。

(3) 21 世纪,信息技术教育在我国中小学中全面开展,计算机教育的起点从大学下移到中小学。水涨船高,这样也为提高大学的计算机教育水平创造了十分有利的条件。

迎接 21 世纪的挑战,大力提高我国高等学校计算机基础教育的水平,培养出符合信息时代要求的人才,已成为广大计算机教育工作者的神圣使命和光荣职责。全国高等院校计算机基础教育研究会和清华大学出版社于 2002 年联合成立了“中国高等院校计算机基础教育改革课题调研组”,集中了一批长期在高校计算机基础教育领域从事教学和研究的专家、教授,经过深入调查研究,广泛征求意见,反复讨论修改,提出了高校计算机基础教育改革思路和课程方案,并于 2004 年 7 月发布了《中国高等院校计算机基础教育课程体系 2004》(简称 CFC 2004)。国内知名专家和从事计算机基础教育工作的广大教师一致认为 CFC 2004 提出了一个既体现先进性又切合实际的思路和解决方案,该研究成果具有开创性、针对性、前瞻性和可操作性,对发展我国高等院校的计算机基础教育具有重要的指导作用。根据近年来计算机基础教育的发展,课题调研组于 2006 年和 2008 年又发布了《中国高等院校计算机基础教育课程体系 2006》(简称 CFC 2006)和《中国高等院校

计算机基础教育课程体系 2008》(简称 CFC 2008), 由清华大学出版社出版。

为了实现 CFC 提出的要求, 必须有一批与之配套的教材。教材是实现教育思想和教学要求的重要保证, 是教学改革中的一项重要基本建设。如果没有好的教材, 提高教学质量只是一句空话。要写好一本教材是不容易的, 不仅需要掌握有关的科学技术知识, 而且要熟悉自己工作的对象、研究读者的认识规律、善于组织教材内容、具有较好的文字功底, 还需要学习一点教育学和心理学的知识等。一本好的计算机基础教材应当具备以下 5 个要素:

(1) 定位准确。要明确读者对象, 要有的放矢, 不要不问对象, 提笔就写。

(2) 内容先进。要能反映计算机科学技术的新成果、新趋势。

(3) 取舍合理。要做到“该有的有, 不该有的没有”, 不要包罗万象、贪多求全, 不应把教材写成手册。

(4) 体系得当。要针对非计算机专业学生的特点, 精心设计教材体系, 不仅使教材体现科学性和先进性, 还要注意循序渐进、降低台阶、分散难点, 使学生易于理解。

(5) 风格鲜明。要用通俗易懂的方法和语言叙述复杂的概念。善于运用形象思维, 深入浅出, 引人入胜。

为了推动各高校的教学, 我们愿意与全国各地、各学校的专家和老师共同奋斗, 编写和出版一批具有中国特色的、符合非计算机专业学生特点的、受广大读者欢迎的优秀教材。为此, 我们成立了“中国高等院校计算机基础教育课程体系规划教材”编审委员会, 全面指导本套教材的编写工作。

这套教材具有以下几个特点:

(1) 全面体现 CFC 的思路和课程要求。可以说, 本套教材是 CFC 的具体化。

(2) 教材内容体现了信息技术发展的趋势。由于信息技术发展迅速, 教材需要不断更新内容, 推陈出新。本套教材力求反映信息技术领域中新的发展、新的应用。

(3) 按照非计算机专业学生的特点构建课程内容和教材体系, 强调面向应用, 注重培养应用能力, 针对多数学生的认知规律, 尽量采用通俗易懂的方法说明复杂的概念, 使学生易于学习。

(4) 考虑到教学对象不同, 本套教材包括了各方面所需要的教材(重点课程和一般课程; 必修课和选修课; 理论课和实践课), 供不同学校、不同专业的学生选用。

(5) 本套教材的作者都有较高的学术造诣, 有丰富的计算机基础教育的经验, 在教材中体现了研究会所倡导的思路和风格, 因而符合教学实践, 便于采用。

本套教材统一规划、分批组织、陆续出版。希望能得到各位专家、老师和读者的指正, 我们将根据计算机技术的发展和广大师生的宝贵意见随时修订, 使之不断完善。

全国高等院校计算机基础教育研究会荣誉会长
“中国高等院校计算机基础教育课程体系规划教材”编审委员会主任

谭浩强

20 世纪 90 年代以来, C 语言迅速在全世界普及推广。无论在中国还是在世界各国, “C 语言程序设计”始终是高等学校的一门基本的计算机课程。C 语言程序设计在计算机教育和计算机应用中发挥着重要的作用。

作者于 1991 年编著了《C 程序设计》一书, 由清华大学出版社出版。该书针对初学者的特点和认知规律, 精选内容, 分散难点, 降低台阶, 例题丰富, 深入浅出。出版后受到广大读者的热烈欢迎。许多读者说“C 语言原来是比较难学的, 但自从《C 程序设计》出版后, C 语言变得不难学了”。根据 C 语言的发展和教学的实践, 作者先后对该书进行了两次大的修订, 先后重印 160 多次, 累计发行超过 1100 万册, 平均每年印刷 60 万册, 成为我国广大初学者学习 C 语言程序设计的主流用书。国内许多介绍 C 语言的书籍多以本书为蓝本。本书曾荣获原电子工业部优秀教材一等奖、全国高等院校计算机基础教育研究会优秀教材一等奖、全国高校出版社优秀畅销书特等奖。这是对我的莫大鼓励和鞭策。

在此书再版之际, 作者想对学习程序设计问题提出以下几点看法。

1. 为什么要学习程序设计

大学生不能满足于只会用办公软件, 应当有更高的要求, 对于理工科的学生尤其如此。

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。程序设计是软件开发人员的基本功。只有懂得程序设计, 才能进一步懂得计算机, 真正了解计算机是怎样工作的。通过学习程序设计, 学会进一步了解计算机的工作原理, 更好地理解和应用计算机; 掌握用计算机处理问题的方法; 培养分析问题和解决问题的能力; 具有编制程序的初步能力。即使将来不是计算机专业人员, 由于学过程序设计, 理解软件生产的特点和生产过程, 就能与程序开发人员更好地沟通与合作, 开展本领域中的计算机应用, 开发与本领域有关的应用程序。

因此, 无论计算机专业学生还是非计算机专业学生, 都应当学习程序设计知识, 并且把它作为进一步学习与应用计算机的基础。

2. 为什么选择 C 语言

进行程序设计, 必须用一种计算机语言作为工具, 否则只是纸上谈兵。可供选择的语言很多, 各有特点和应用领域。C 语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好, 既具有高级语言的优点, 又具有低级语言的许多特点, 既适于编写系统软件, 又能方便地用来编写应用软件。

有人以为 C++ 语言出现后, C 语言过时了, 淘汰了, 这是一种误解。C++ 是为处理较大规模的程序开发而研制的大型语言, 它比 C 语言复杂得多, 难学得多。事实上, 将来并不是每个人都需要用 C++ 编制大型程序。C 语言是更为基本的。美国一位资深软件专家写了一篇文章, 题目是“对计算机系学生的建议”, 是经验之谈, 可供参考。他说“大学生毕业前要学好 C 语言, C 语言是当前程序员共同的语言。它使程序员互相沟通, 比你在大学学到的‘现代语言’(比如 ML 语言、Java 语言、Python 语言或者正在教授的流行语言)都更接近机器”。他指出: “不管你懂得多少延续、闭包、异常处理, 只要你不能解释为什么 `while(*s++ = *t++)` 的作用是复制字符串, 那你就是在盲目无知的情况下编程, 就像一个医生不懂最基本的解剖学就在开处方”。

C 语言更适合于解决某些小型程序的编程。C 语言作为传统的面向过程的程序设计语言, 在编写底层的设备驱动程序和内嵌应用程序时, 往往是更好的选择。

现在大多数高校把 C 语言作为第一门计算机语言进行程序设计教学, 这是合适的, 有了 C 的基础, 在需要时进一步学习 C++, 也是很容易过渡的。

3. 怎样组织程序设计的教学? 怎样处理算法和语言的关系

要进行程序设计, 要解决两个问题:

(1) 要学习和掌握解决问题的思路和方法, 即算法;

(2) 学习怎样实现算法, 即用计算机语言编写程序, 达到用计算机解题的目的。

因此, 课程的内容应当主要包括两个方面: 算法和语言。算法是灵魂, 不掌握算法就是无米之炊。语言是工具, 不掌握语言, 编程就成了空中楼阁。二者都是必要的, 缺一不可。作者的做法是: 以程序设计为中心, 把二者紧密结合起来, 既不能孤立地抽象地研究算法, 更不能孤立地枯燥地学习语法。

算法是重要的, 但本课程不是专门研究算法与逻辑的理论课程, 不可能系统全面地介绍算法; 也不是脱离语言环境研究算法, 而是在学习编程的过程中, 介绍有关的典型算法, 引导学生思考怎样构造一个算法。编写程序的过程就是设计算法的过程。

语言工具也是重要的, 掌握基本的语法规则是编程的基础, 如果不掌握必要的语法规则, 连最简单的程序也编不出来, 或者编出来的程序错误百出, 无法运行。但是掌握 C 语言绝不能靠死学死记, 就像熟读英语的语法不一定会写英文文章, 靠字典是学不好外语的。如果你去看 C 语言标准文本, 可能感觉如看“天书”一样, 恐怕只有计算机专家才能看懂。绝不能把程序设计课程变成枯燥地介绍语法的课程, 学习语法要服务于编程。

在 30 年前我们编写《BASIC 语言》时就已经遇到了这个问题, 我们坚决摒弃了孤立地介绍语法的做法, 而是以程序设计为中心, 把算法与语言紧密结合起来。不是根据语言规则的分类和顺序作为教学和教材的章节和顺序, 而是从应用的角度切入, 以编程为目的, 以编程为主线, 从初学者的认知规律出发, 由浅入深, 由易到难, 构造了教材和教学的体系。一开始就让学生看懂简单的程序, 编写简单的程序, 然后逐步深入。语法规则不是通过孤立的学习而是在学习编程的过程中学到的。随着编程难度的逐步提高, 算法和语法的学习同步趋于深入。学生在富有创意、引人入胜的编程中, 学会了算法, 掌握了语法, 把枯燥无味的语法规则变成生动活泼的编程应用。事实证明这种做法是成功的。多年来, 我们坚持和发展了这种行之有效的方法, 取得了很好的效果。

近年来许多学校的经验表明,按照这种思路进行教学,教师容易教,学生容易学,效果很好。

4. 怎样学习 C 程序设计

(1) 要着眼于培养能力。C 语言程序设计并不是一门纯理论的课程,而是一门应用的课程。应当注意培养分析问题的能力、构造算法的能力、编程的能力和调试程序的能力。

(2) 要把重点放在解题的思路,通过大量的例题学习怎样设计一个算法,构造一个程序。初学时更不要在语法细节上死背死抠。一开始就要学会看懂程序,编写简单的程序,然后逐步深入。语法细节是需要通过较长期的实践才能熟练掌握的。初学时,切忌过早地滥用 C 语言的某些容易引起错误的细节(如不适当地使用++和--)。

(3) 掌握基本要求,注意打好基础。在学校学习阶段,主要是学习程序设计的方法,进行程序设计的基本训练,为将来进一步学习和应用打下基础。不可能通过几十小时的学习,由一个门外汉变成编程高手,编写出大型而实用的程序,要求应当实事求是。如果学时有限,有些较深入的内容可以选学或自学,把精力放在最基本、最常用的内容上,学好基本功。

(4) 要十分重视实践环节。光靠听课和看书是学不会程序设计的,学习本课程既要掌握概念,又必须动手编程,还要亲自上机调试运行。读者一定要重视实践环节,包括编程和上机。既会编写程序,又会调试程序。学得好与坏,不是看你“知不知道”,而是“会不会干”。考核方法应当是编写程序和调试程序,而不应该只采用是非题和选择题。

(5) 要举一反三。学习程序设计,主要是掌握程序设计的思路和方法。学会使用一种计算机语言编程,在需要时改用另一种语言应当不会太困难。不能设想今后一辈子只使用在学校里学过的某一种语言。但是无论用哪一种语言进行程序设计,其基本规律是一样的。在学习时一定要学活用活,举一反三,掌握规律,在以后需要时能很快地掌握其他新的语言进行编程。

(6) 要提倡和培养创新精神。教师和学生都不应当局限于教材中的内容,应该启发学生的学习兴趣和创新意识。能够在教材程序的基础上,思考更多的问题,编写难度更大的程序。在本书每章的习题中,包括了一些难度较大的题目,建议学生尽量选做,学会自己发展知识,提高能力。

(7) 如果对学生有较高的程序设计要求,应当在学习本课程后,安排一次集中的课程设计环节,要求学生独立完成一个有一定规模的程序。

5. 从实际出发,区别对待

学习本课程的有计算机专业学生,也有非计算机专业的大学生;有本科生,也有专科(高职)学生;有重点大学的学生,也有一般大学的学生。情况各异,要求不同,必须从实际出发,制订出切实可行的教学要求和教学方案,切忌脱离实际的一刀切。

例如,对计算机专业学生,要求应当比非计算机专业高,尤其是对算法的要求应当高一些,不仅会用现成的算法,还应当会设计一般的算法。最好能在学完本课程后独立完

成一个有一定规模的程序。

对高职学生的要求应不同于本科生，更不应搬用重点大学的做法，不宜在算法上要求太高，因为高职不是培养设计算法的人才，而应切实掌握语言工具，具有较强的动手和实践能力，例如编码能力、调试能力。

对基础较好、学生程度较高的学校，可以少讲多练，强调自学，有的内容课堂上可以不讲或少讲，指定学生自学。引导学生通过自学和实践发展知识，尽可能完成一些难度较高的习题。

6. 为什么要修订《C 程序设计》

任何工作都要与时俱进，不断完善，追求完美。计算机科学技术不断发展，C 语言也在发展，在高校计算机教学过程中不断累积了新的经验，广大读者在学习过程中也对本书提出了一些新的期望和建议。因此，作者对《C 程序设计》一书再次进行修订，第四版保持了前三版的写作风格，保持了概念清晰、通俗易懂的特点，并在以下几个方面作了修改：

(1) 按照 C 99 标准进行介绍，以适应 C 语言的发展，使程序更加规范。

目前大多数介绍 C 语言的教材是按照 C 89 标准介绍的，本书按照 C 99 的规定进行介绍，程序符合 C 99 的要求。例如：

① 数据类型介绍中，增加了 C 99 扩充的双长整型(long long int)、复数浮点型(float_complex, double_complex, long long_complex)、布尔型(bool)等，使读者有所了解。

② C 99 要求，main 函数的类型一律指定为 int 型，并在函数的末尾加一个返回语句“return 0;”。

③ C 99 增加了注释行的新形式——以双斜线//开始的内容作为注释行，这本来是 C++ 的注释行形式，现在 C 99 把它扩充进来了，使编程更加方便。同时保留了原来的/*……*/形式，以使原来按 C 89 标准编写的程序可以不加修改仍可使用。本书采用 C 99 的注释新形式，读者使用更方便，而且符合发展需要。因此，本书的程序基本上采用下面的形式：

```
#include <stdio.h>           //以“//”作为注释行的开始
int main()                   //指定 main 函数为 int 类型
{
    return 0;                 //如函数正常执行,返回整数 0
}
```

④ C 99 增加的其他一些具体内容，在书中有关章节中专门注明，以提醒读者。

由于 C 99 是在 C 89 的基础上增加或扩充一些功能而成的，因此 C 89 和 C 99 基本上是兼容的。用 C 89 编写的程序在 C 99 环境下仍然可以运行。C 99 所增加的有些新的功能和规则，是在编制比较复杂的程序时为方便使用和提高效率而用的，本书对目前暂时用不到的内容不作介绍，以免读者分心，增加学习难度。在将来进行深入编程时再逐步了解和学习。

目前社会上使用的一些 C 编译系统，有的并未实现 C 99 的要求，有的实现了 C 99 的部分功能。

(2) 考虑到国内当前的情况和读者使用的方便，本书中的程序是用 Visual C++ 6.0 编译的，运行结果也是在 Visual C++ 6.0 环境下得到的。在分析时也以 Visual C++ 6.0 环境

为例进行介绍。如果采用其他编译系统，也大同小异，不会产生任何问题。

(3) 加强算法。第2章专门介绍算法的概念、算法的特点、表示算法的工具以及怎样设计算法，并通过一些简单的例子说明怎样构造一个算法。使读者有一个初步的、基本的了解。在以后各章中，由浅入深地结合例题介绍各种典型的算法，并且马上用C语言实现此算法，写出程序。这样就使算法与程序紧密结合，而且通过运行程序，得到结果，便于验证算法的正确性。学习时不会觉得抽象，而会觉得算法具体有趣，看得见，摸得着。

在各例题中，在给出问题后，都先进行分析问题，介绍解题思路，也就是构造算法，然后才是根据算法编写程序，而不是先列出程序再解释程序，从中了解算法。这样做，更符合读者认知规律，更容易理解算法，也引导读者在做题时先考虑算法再编程，而不是坐下来就写程序。培养好的习惯。

(4) 更加通俗易懂，容易学习。作者充分考虑到广大初学者的情况，精心设计体系，适当降低门槛，便于读者入门。尽量少用深奥难懂的专业术语，用通俗易懂的方法和语言阐述清楚复杂的概念，使复杂的问题简单化。没有学过计算机原理和高等数学的读者完全可以掌握本书的内容。

本书采用作者提出的“提出问题—解决问题—归纳分析”的新的教学三部曲，先具体后抽象，先实际后理论，先个别后一般。而不是先抽象后具体，先理论后实际，先一般后个别。在介绍每个例题时，都采取以下的步骤：给出问题—解题思路—编写程序—运行结果—程序分析—有关说明，使读者很容易理解。即使没有教师讲解，读者也能看懂本书的内容，就有可能做到，教师少讲，提倡自学，上机实践。

(5) 重新组织内容。根据近年来各校教学中的实际情况，作者对本书内容作了以下调整：

① 根据非计算机专业学生的培养要求和将来工作的情况，在教材中不再包括“位运算”一章。考虑到有部分读者需要学习这部分内容，将“位运算”的内容列入《C程序设计(第四版)学习辅导》一书中，供选学。

② 取消《C程序设计(第三版)》中的第3章“数据类型、运算符与表达式”。这章内容涉及数据在计算机内存中的存储形式，读者感到一开始就接触这些内容比较难学。在第四版中，对这些内容进行精简，不再单独列章，将其中最基本的、必须了解的内容结合在第3章“最简单的C程序设计”中介绍，这样可以降低学习难度。

③ 第三版包括“预处理命令”一章，考虑到这章的内容不是最基本的，在第四版中对它不作详细的介绍，而把它放在《C程序设计(第四版)学习辅导》一书中，供选学。

④ 考虑到绝大多数读者在学习本课程之前未学习过其他计算机语言，缺乏对计算机语言和程序的基本了解，把第三版的第1章“C语言概述”改为“程序设计和C语言”，在这一章中增加了计算机语言和程序的基本知识。

⑤ 许多学校把本课程放在一年级学习，此时还未学习或未学完高等数学，在学本课程时对有关高等数学的程序感到有困难，在第四版中不再包括有关高等数学的内容，具有高中以上文化程度的读者都能看懂本书。

⑥ 在第三版的“结构体与共用体”一章中有设计链表(链表的建立、插入、删除和输出等)的内容，对于非计算机专业学生来说，难度较大。作者认为，这部分内容对非计算

机专业可以不作为基本要求,在第四版中对这部分内容作了精简,只对链表作简单的介绍,读者对之有一定了解即可。考虑到这部分对计算机专业学生是需要的,我们把有关链表的详细内容作为习题供选做,并在《C程序设计(第四版)学习辅导》一书的习题解答中给出完整的程序,供需要者参考。

⑦ 专门编写了“C程序案例”一章,综合应用各章的知识。提供了不同难度、不同类型的程序。阅读这些程序,可以使学生了解怎样去编写应用程序,提高自己的编程能力。这部分内容安排在《C程序设计(第四版)学习辅导》一书中,供需要者参考。

相信经过修改后,本书会更加容易学习,效果更好。

7. 为了满足不同的需要,出版不同层次的 C 程序设计教材

全国各校的情况不同,学生的基础和学习要求也不尽同,不可能都采用同一本教材。教材应当满足多层次多样化的要求。许多学校的老师认为《C程序设计》是一本经过长期教学实践检验的优秀教材,其内容与风格已为广大师生所熟悉,希望在《C程序设计》的基础上组织不同层次的教材,供不同对象选用。作者与清华大学出版社反复研究,决定出版C程序设计的系列教材,包括以下3种:

(1)《C程序设计(第四版)》,即本书。本书系统全面,内容深入,讲授详尽,包含了许多其他教材中没有的内容,尤其是针对编程实践中容易出现的问题作了提醒和分析,是学习C语言程序设计的理想教材。适合程度较高、基础较好的学校和读者使用。

(2)《C程序设计教程》。以《C程序设计》一书的内容为基础,适当减少内容,突出重点,紧扣最基本的要求,适合学时相对较少的本科院校使用。该书已被教育部正式列为“普通高等教育国家级规划教材”。

(3)《C语言程序设计(第2版)》。内容更加精练,要求适当降低,写法上更加通俗易懂,适合应用型大学和程度较好的高职院校使用。该书亦已列为“普通高等教育国家级规划教材”和普通高等教育精品教材。

为了帮助读者学习本书,作者同时编写了《C程序设计(第四版)学习辅导》,内容包括:本书各章习题的参考答案;位运算;C程序案例;Visual C++ 集成环境的使用方法;程序的调试与测试;上机实习指导等。该书由清华大学出版社于2010年与本书同时出版。

在本书出版之际,作者衷心感谢全国高等院校计算机基础教育研究会和全国各高校教师多年来始终不渝的关心与鼎力支持,感谢广大读者给予我的理解与厚爱,感谢清华大学出版社30年来的密切合作与支持。没有这一切,我不可能取得今天的成就。我永远感谢曾经帮助和支持过我的、相识的和不相识的同志和朋友。

薛淑斌和谭亦峰高级工程师参加了本书的研讨并编写部分程序。由于作者水平有限,本书肯定会有不少缺点和不足,热切期望得到专家和读者的批评指正。

谭浩强 谨识

2010年3月于清华园

对使用本教材的建议

(1) 本书是作为高等学校学生学习 C 程序设计的教材,对象是没有学过计算机程序设计的大学生。本书既注重概念清晰,使读者建立起对程序设计和 C 语言的清晰理解,又注意引导学生学以致用,使学生在较短的时间内初步学会用 C 语言编写程序,具有初步的编程知识和能力,而不是仅停留在理论知识层面上。虽然如此,本书与就业上岗前的职业培训教材是有区别的,也不是供软件开发人员使用的手册和技术规范,本书带有基础的性质,主要帮助学生学习程序设计方法,学习怎样去编写程序,为以后的进一步提高与应用打好基础。如果读者准备从事软件开发工作,可以在学习本书的基础上进一步学习有关专业知识和职业要求。

(2) 本书系统全面,内容丰富,供基础较好的学校和学生学习。建议采取课堂讲授与自学相结合的方法。在课堂上教师主要介绍编程思路和怎样用 C 语言去实现算法,不要孤立地一一介绍语法的细节,但是要在介绍程序时重点指出关键之处,以及容易出错的地方。要求学生通过自己上机实践来理解程序设计方法,学会正确使用 C 语言工具,具有初步编程能力。语法不是靠讲和背学会的,而是在实践中掌握的。

(3) 在本书第 11 章“常见错误分析”中列举了初学者在编程序时常出现的错误,这是作者在多年教学实践中收集和总结出来的,是很有价值的,希望教师和同学能充分利用这个资源。教师可以结合教学提醒学生避免出现类似的错误。学生在学习过程中可以随时翻阅,了解在什么情况下容易出错。在经过一段时间的编程和上机实践后,再系统地阅读一下,回顾和总结自己易出错的问题,这样可以减少错误,提高编程效率。

(4) 要善于利用习题。本书各章中的习题包括不同类型、不同程度的 142 道题目。其中有些题目的难度高于书中的例题,这样做的目的是使学生不满足于已学过的内容,而要举一反三,善于发展已有知识,提倡创新精神,培养解决问题的能力。有的专家和读者说,如果能独立地完成全部习题,他的 C 语言学习就过关了。希望教师能指定学生完成各章中有一定难度的习题。希望学生能尽量多做习题,以提高自己的水平。

在《C 程序设计(第四版)学习辅导》一书中,提供了绝大多数习题的参考解答,列出了程序。对于比较难的习题,除了给出程序外,还作了比较详细的说明。这些习题解答实际上是作者对本教材例题的补充,希望读者能充分利用它。学生即使没有时间自己做全部习题,如果能把全部习题的参考解答都看一遍,而且都能看懂,也会很有收获,能扩大眼界,丰富知识。教师也可以挑选一些习题解答在课堂上讲授,作为补充例题。

(5) 预处理指令往往是 C 程序中必要的部分,尤其是用 `#include` 指令来包含头文件和用 `#define` 指令定义符号常量。在本教材中结合编写程序,介绍了怎样使用这两种预处理指令。在《C 程序设计(第四版)学习辅导》一书中,专门有一章系统地、详细地介绍各种预处理指令的使用,以供使用参考。教师可在介绍 `#include` 指令和 `#define` 指令时,说明还有其他预处理指令,请同学们自己学习参考。

(6) “位运算”是 C 语言区别于其他高级语言的一个重要特点。C 语言能对“位”进行操作,使得 C 具有比较接近机器的特点。考虑到非计算机专业学生的情况,在这次修订时在教材中不再包括位运算的内容。但是,在编写系统软件和数据采集、检测与控制中往往需要用到位运算。信息类专业的学生需要学习这方面的知识,因此,把位运算的内容放到《C 程序设计(第四版)学习辅导》一书中,计算机和其他信息类专业可以把它列入教学内容,其他读者可以选学。

(7) 为了便于教学,本教材中的例题程序的规模一般都不大。在学完各章内容之后,需要综合应用已学过的知识,编写一些应用程序,以提高编程能力。在《C 程序设计(第四版)学习辅导》一书中专门有一章“C 程序案例”,这些案例很有实用价值,对于读者在学习本书后提高编程能力会有很大的帮助。要善于利用这些资源,教师可以指定学生阅读这些程序。

(8) 由于学时少,只靠几十小时的教学就能使学生真正掌握 C 程序设计是困难的,如果有条件,最好在学完本教材后安排一次课程设计,要求学生独立完成一个有一定规模的程序设计,这是一个重要的教学实践环节,能大大提高学生的独立编程能力。

(9) 有的学校由于分配给本课程的学时有限,可能不能讲完教材中的全部内容。建议不要把本书后面几章舍弃,应当让学生基本学完第 1~10 章,使学生对 C 语言有全面的了解。例如,文件的概念是很重要的,宁可作简单的介绍,也不要放弃。前 5 章的进度可以快些,有些程序可指定学生自学。

第 1 章 程序设计和 C 语言	1
1.1 什么是计算机程序	1
1.2 什么是计算机语言	1
1.3 C 语言的发展及其特点	3
1.4 最简单的 C 语言程序	5
1.4.1 最简单的 C 语言程序举例	6
1.4.2 C 语言程序的结构	10
1.5 运行 C 程序的步骤与方法	12
1.6 程序设计的任务	14
习题	15
第 2 章 算法——程序的灵魂	16
2.1 什么是算法	16
2.2 简单的算法举例	17
2.3 算法的特性	21
2.4 怎样表示一个算法	22
2.4.1 用自然语言表示算法	22
2.4.2 用流程图表示算法	22
2.4.3 三种基本结构和改进的流程图	26
2.4.4 用 N-S 流程图表示算法	28
2.4.5 用伪代码表示算法	31
2.4.6 用计算机语言表示算法	32
2.5 结构化程序设计方法	34
习题	36
第 3 章 最简单的 C 程序设计——顺序程序设计	37
3.1 顺序程序设计举例	37
3.2 数据的表现形式及其运算	39
3.2.1 常量和变量	39