



普通高等教育“十一五”国家级规划教材

全国高校出版社优秀畅销书一等奖

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 程序设计教程(第3版)

谭浩强 著

- 《C程序设计》(发行逾1400万册)的姊妹篇
- 内容更精练,重点更突出
- 使C语言更容易学习
- 紧扣最基本的教学要求

清华大学出版社





普通高等教育“十一五”国家级规划教材

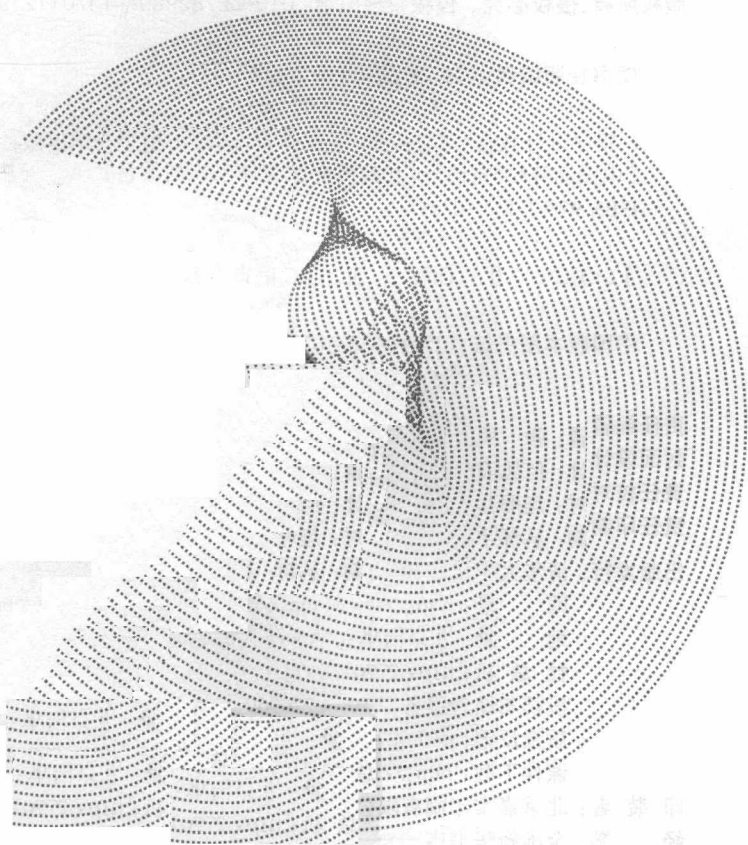
全国高校出版社优秀畅销书一等奖

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 程序设计教程 (第3版)

谭浩强 著



清华大学出版社
北京

内 容 简 介

C语言是国内外广泛使用的计算机语言,学会使用C语言进行程序设计是计算机工作者的一项基本功。本书系统地介绍怎样使用C语言进行程序设计。通过本书的学习,可以基本掌握C语言的主要内容和使用方法,同时学习程序设计的方法及有关算法的基本知识,初步掌握用计算机解题的全过程。

本书作者具有丰富的教学经验和编写教材的经验,善于把复杂的问题简单化,化解了学生学习过程中的许多困难。作者所著的《C程序设计》一书受到专家和读者的一致好评,公认为是学习C程序设计的理想教材,被全国大多数高校选用,是学习C语言的主流用书,已累计发行逾1400万册,创同类书的国内外发行最高纪录。

根据国内一般大学非计算机专业的教学需要,作者在《C程序设计》一书的基础上重新改写并出版了《C程序设计教程》,适当减少内容,紧扣基本要求,突出重点,适合“少学时”的需要。根据近年来的教学实践,本书重新构建教材体系,进一步压缩内容,突出重点,加强算法训练,更加实用。用通俗易懂的方法和语言阐明复杂的概念,使读者更加容易学习。同时采取适当的方法,注意培养包括计算思维在内的科学思维。

本书可用作全国高等学校各专业的正式教材,尤其适合一般院校的非计算机专业使用,同时也是一本供广大读者自学的好教材。本书还配有辅助教材《C程序设计教程(第3版)学习辅导》。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

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程序设计教程/谭浩强著. —3版. —北京:清华大学出版社,2018
(中国高等院校计算机基础教育课程体系规划教材)
ISBN 978-7-302-50382-8

I. ①C… II. ①谭… III. ①C语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第122970号

责任编辑:张民

封面设计:何凤霞

责任校对:焦丽丽

责任印制:丛怀宇

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:19.25

字 数:469千字

版 次:2007年7月第1版 2018年8月第3版

印 次:2018年8月第1次印刷

定 价:39.80元

产品编号:078357-01

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

▲ 全国政协原副主席、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

序

PREFACE

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。

三十多年来,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。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研究组”,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,课题研究

组先后于 2006、2008 和 2014 年发布了《中国高等院校计算机基础教育课程体系》的新版本,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的基本建设。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区、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。

本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及时修订,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

C语言是国内外广泛使用的一种计算机语言。学会使用C语言进行程序设计是计算机工作者的一项基本功。

1991年,作者所著的《C程序设计》由清华大学出版社出版。该书出版后反映很好。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”。该书被全国大多数高校选为正式教材,并被许多高校指定为研究生入学考试必读教材。该书已成为国内读者学习C语言的主流用书。20多年来,该书已先后出了5版,重印120多次,累计发行1400多万册,位居国内外同类书之首。作者到全国各高校和各企事业单位访问时,许多在校师生和已毕业参加工作的人士都说他们学过这本书,印象很深,作者在内心深切地感受到广大读者的殷切期望。

各校师生普遍认为该书内容系统,讲解详尽,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,被认为是学习C语言程序设计的理想教材。同时有的学校提出,由于各校情况不完全相同(例如,学校的类型不同、教学要求不同、安排的学时数不同、学生的基础不同),希望在保持原有的优点的基础上,能提供适用于不同要求的版本。作者和出版社征求了多方面的意见,进行了反复的研究,除了继续出版和完善《C程序设计》以外,还针对学时较少的学校,于2007年出版了《C程序设计教程》。该教材以《C程序设计》为基础,紧扣最基本的要求,适当减少内容,压缩篇幅,突出重点。出版后受到广泛欢迎,认为内容适当,概念清晰,被教育部评为普通高等教育“十一五”国家级规划教材,向全国各高校推荐。

经过几年的教学实践,作者于2013年对《C程序设计教程》一书进行了修订。现在又进行一次修订,在修订的过程中,作者思考了以下几个方面的问题。

1. 程序设计课程的作用与要求

近年来,在讨论C程序设计课程改革时,有的老师主张要学深学透;有的认为不能要求太高,主要是打好基础;有的认为有一些了解、初识即可;有的则认为大学生毕业后由自己编程序的机会不多,因此可不必学,课程可以取消。这些引起人们深入思考:大学生要不要上程序设计课程?程序设计课程的目的和作用是什么?学习程序设计课程的要求是什么?程序设计课程的内容应该是什么?

作者认为,学习程序设计能够使大学生更好地理解计算机和应用计算机。

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,懂得计算机是怎样工作的,才能较好地懂得计算机。通过学习程序设计,能使学生学习到用计算机处理问题的方法,培养学生提出问题、分析问题和解决问题的能力,并且具有编制程序的初步能力,能较好地应用计算机。即使将来不是计算机专业人员,由于学过程序设计,了解软件的特点和生产过程,也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。对我国所有理工类学生都开设程序设计课程,并且把它作为进一步学习与应用计算机的基础,是十分必要的。

2. 要不要学习 C 语言

进行程序设计,必须用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。C 语言是基础而实用的语言,并不是每一种语言都具有此特点,有的语言实用,但不能作为基础语言(如 FORTRAN),有的语言可以作为基础,但实际应用不多(如 Pascal)。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。C 语言是多年来国内外使用最广泛的语言,国内外许多专家认为,C 语言是最基本的通用语言,有了 C 语言的基础后,掌握任何一种语言都不困难。C 语言被认为是计算机专业人员的基本功。

有人认为有了 C++ 语言以后,C 语言就过时了,这是一种误解。C++ 语言是为设计大型程序应运而生的。将来从事系统开发的人员以及计算机专业学生需要学习 C++ 语言或其他面向对象的语言。面向对象编程使用的是复杂的类层次结构与对象,适于处理大型的模块程序,但是在某些情况下并不比 C 语言程序更为有效。C 语言作为传统的面向过程的程序设计语言,更适于解决某些小型程序的编程。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。

对复杂的问题,面向对象方法符合人们的思维方式。对简单的问题,面向过程方法符合人们的思维方式,而面向过程是最基本的。对初学者来说,学习 C 语言显然比学习 C++ 语言容易得多,许多学校把 C 语言作为大学学生的第一门计算机语言,是比较合适的。有了 C 语言的基础,学习 C++ 语言也是很容易的。目前,如果有些非计算机专业学生学习 C++ 语言,其作用应当是了解面向对象的程序设计方法,为以后需要时进一步学习打下初步基础,要求不宜太高。

本书选择 C 语言为学习程序设计使用的语言。

3. 程序设计课程的性质和体系,正确处理算法与语法的关系

关于 C 程序设计课程的性质,应该说,它既有基础的性质(了解计算机处理问题的方式,学习算法),又有应用和工具的性质(掌握语言工具,具有编程的初步能力,能具体应用),二者兼顾。因此,既要注意讲清概念,使学生建立正确的概念,又要培养学生实际处理问题的能力。

程序设计有 4 个要素: ① 算法——程序的灵魂; ② 数据结构——加工的对象; ③ 语言——编程工具(算法要通过语言来实现); ④ 合适的程序设计方法。程序设计教学是否成功取决于能否将以上 4 个要素紧密结合。

本教材自始至终把这四方面自然、有机地结合,全面兼顾。不是孤立地介绍语法,也不是全面系统地介绍算法。本书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,而是从应用的角度出发,以编程为目的和主线,由浅入深地介绍怎样用C语言处理问题。把算法和语法紧密结合,同步展开,步步深入。精心安排顺序,算法的选择由易到难,细心选择例子,使读者容易学习。在此基础上,构造了新的教学和教材体系。具体的做法是:在每一章中,首先举几个简单的例子,引入新的问题,接着介绍怎样利用C语言解决简单的问题,然后再循序渐进地介绍较深入的算法和程序。使学生在富有创意、引人入胜的编程中,学会算法,掌握语法,领悟程序设计的思想和方法,把枯燥无味的语法规则变成生动活泼的编程应用。多年的实践表明,这种做法是成功的。

建议教师在讲授时,以程序为中心展开,着重讲清解题思路以及怎样用程序实现它,不要孤立介绍语法规则,教材中叙述的语法规则可以在介绍编写程序的过程中加以说明,或在简单介绍后请学生自己阅读,并通过上机实践掌握。

4. 在程序设计课程中注意培养科学思维

学习程序设计的另一个重要作用是可以培养学生的科学思维能力。近年来,国内外有些专家提出要重视和研究计算思维,认为计算思维是运用计算机科学的基础概念进行问题求解、系统设计和理解人类行为的思维活动。

计算思维是科学思维的组成部分。人们在学习和应用过程中已经认识到:计算机不仅是工具,而且是可以启发人们思考问题的科学方法。通过学习和应用计算机,人们改变了旧的思维方式和工作方式,逐步培养了现代的科学思维方式和工作方式,懂得现代社会处理问题的科学方法,这个意义比掌握工具更为深远。计算思维是信息时代中的每个人都应当具备的一种思维方式,要让思维具有计算的特征。

计算机不仅为不同专业提供了解决专业问题的有效方法和手段,而且提供了一种独特的处理问题的思维方式。把计算机处理问题的方法和技术用于各有关领域,有助于提升各个领域的科学水平,开拓新的领域。积极在计算机的教学中引入跨学科元素,启迪跨学科计算思维,会对各个学科的发展产生深远的影响。

计算思维不是悬空的抽象概念,是体现在各个环节中的。算法思维就是典型的计算思维。学习程序设计就是培养计算思维的有效途径。

计算思维的培养不是孤立进行的,不是依靠另开专门课程讲授的,而是在学习和应用计算机的过程中培养的。多年来,人们在学习和应用计算机过程中不断学习和培养了计算思维,正如学习数学培养了理论思维,学习物理培养了实证思维一样。对计算机的学习和应用越深入,对计算思维的认识也越深刻。

培养计算思维不是目的,正如学习哲学不是目的的一样。学哲学的目的是认识世界、改造世界。培养计算思维的目的是更好地应用计算技术,推动社会各领域的发展与提高。要正确处理好培养计算思维与计算机应用的关系。

程序设计的各个环节都体现了计算思维。没有必要去声明或争论:这个问题是计算思维,那个问题属于其他什么思维。属于计算思维的就重视,否则就不重视,这是书生气十足的做法。只要有利于培养大学生科学思维,都应当大力提倡,大学生需要

培养多种思维的能力。

本教材注意在教学过程中努力培养学生的科学思维(包括计算思维)。在介绍每一个问题时,都采取以下步骤:提出问题→解题思路→编写程序→运行结果→程序分析→有关说明。在“解题思路”中,分析问题,介绍算法,建立数学模型。使读者首先把注意力放在处理问题的思路和方法上,而不是放在语法细节上。在确定算法之后,再使用 C 语言编写程序就顺理成章了。在“程序分析”中,再进一步分析程序的思路及其实现方法。这样,思路清晰,逻辑性强,有利于形成科学的思维方法。希望读者不仅要注重学习知识,更要注重学习方法,掌握规律,举一反三。

5. 从实际出发,区别对待

学习程序设计的人群中,有的是计算机专业学生,有的是非计算机专业的学生;有的是本科生,有的是专科(高职)学生;有的是重点大学的学生,有的是一般大学的学生。情况各异,要求不同,必须从实际出发,制订切实可行的教学方案,对象不同,要求也应不同,并非越深越好。切忌脱离实际的一刀切。

例如,对计算机专业学生,应有较高的要求,应作为基本功来要求。尤其是对算法的要求应当高一些,需要较系统学习各种算法,不仅会用现成的算法,还应当会设计一般的算法:熟练掌握语言工具;了解软件开发的方法和规范;掌握程序设计的全过程;具有一定的编程实践经验;能够举一反三,掌握几种计算机语言。最好能在学完本课程后独立完成一个有一定规模的程序。

对一般大学非计算机专业的学生,多数人将来工作中不一定要用 C 语言编程,本课程的目的不是培养熟练的程序员。学习程序设计的目的是学习计算机处理问题的方法,具有一定的编程知识和应用能力。对非计算机专业的学生,对算法的要求不能太高。算法选择典型的、难度不太大的,只要求掌握基本的算法和设计算法的思路,为以后进一步学习和应用打下基础。没有必要把语言的每一个细节都学透,而是围绕程序设计,使用语言工具中基本的、常用的部分,对初学者不常用的部分可暂时不学。有些部分概念很重要,但难度较大,初学时用得不多,但以后会用到,可作简单介绍,打下基础。在非计算机专业中不宜提“学深学透”的要求。要求是相对的,一切以实际条件为转移。

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的,而应培养切实掌握语言工具,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,可以采取少讲多练,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掌握知识,尽可能完成一些难度较高的习题。

全国各校的情况不同,学生的基础和学习要求也不尽相同,不可能都采用同一本教材。教材应当服务于教学,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决定出版 C 程序设计的系列教材,目前已出版的有以下 3 种:

(1) 《C 程序设计(第五版)》。该书系统全面,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(第 3 版)》,即本书。它是在《C 程序设计(第五版)》的基础上改编的,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广大高校使用。本书为普通高等教育国家级规划教材,推荐给各校使用。

(3) 《C 语言程序设计(第 3 版)》。内容更加精练,要求适当降低,适合程度较好的高职院校使用。该书亦为普通高等教育国家级规划教材和国家级精品教材。

6. 本次修订版的特点

在本次修订中保持了本书概念清晰、通俗易懂的特点,体现了以下特点:

(1) **按照 C 99 标准进行介绍**,以适应 C 语言的发展,使编写程序更加规范。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 根据 C 99 的建议,main 函数的类型一律指定为 int 型,并在函数的末尾加返回语句“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,会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。过去用 C 89 编写的程序在 C 99 环境下仍然可以运行。C 99 所增加的许多新的功能和规则,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,在初学时可以不涉及,因此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。

(2) **加强算法,强化解题思路**。在各章中由浅入深地结合例题介绍各种典型的算法。对穷举、递推、迭代、递归、排序(包括比较交换法、选择法、起泡法)、矩阵运算、字符处理应用等算法作了详尽的介绍,对难度较大的链表处理算法的思路作了清晰说明,使读者逐步建立算法思维。

介绍例题时,在给出问题后,先是进行问题分析,探讨解题思路,构造算法,然后才根据算法编写程序,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

合读者认知规律,更容易理解算法,有利于培养计算思维。引导读者在看到题目后,先考虑算法再编程,而不是坐下来就写程序,以培养好的习惯。

(3) **对指针作了更明确详尽的说明。**指针是学习 C 语言的重点,也是难点。不少读者反映难以掌握指针的实质和应用。作者在《C 程序设计》和本书中,明确指出了“指针就是地址”,许多读者反映这是“画龙点睛,点出了问题的实质”,觉得一通百通,许多问题迎刃而解了。许多学校的师生反映,原来在学习指针时感到特别难懂,后来看了《C 程序设计》后豁然开朗了。希望作者保持这一正确做法,并能对指针再作更详尽的说明。作者根据各校教学中的情况和一些师生提出的问题,在本次修订中对指针的性质作了进一步说明,指出:我们所说的指针就是地址,这个地址不仅是在内存中的位置信息(即纯地址),而且包括在该存储单元中的数据的类型信息,并对此作了有力而明确的说明,使读者对指针的性质有进一步的认识。请读者阅读本书时加以注意。

(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。实践证明这样做符合读者的认知规律,读者很容易理解。

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**,对一些典型的算法,还有**算法分析**,使读者更好理解。

把算法与语言二者紧密而自然地结合,而且通过运行程序,看到结果,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。

本书便于自学。具有高中以上文化水平的人,即使没有教师讲解,也能基本上掌握本书的内容。这样就有可能做到:教师少讲,提倡自学,上机实践。

考虑到教学的基本要求,本书适当降低难度,对以下几个问题进行了适当处理:

① **简化输入输出格式。**C 语言的输入输出格式比较烦琐复杂,初学者往往感到难以掌握。本次修订时,只介绍最基本的格式(%d,%f,%e,%c,%s),能够进行输入输出就行,其他附表供查用。

② 在函数一章中,简化一些初学者不常用的内容,如“内部函数和外部函数”,对存储类别的介绍也从简。

③ 指针一章主要介绍一级指针,关于二级指针只介绍有关二维数组的内容。对“指向函数的指针”“返回指针值的函数”“指针数组和多重指针”“动态内存分配与指向它的指针变量”等较深入而初学者用得不多的内容不再介绍。

④ 只介绍结构体,不介绍共用体。

⑤ 链表处理(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,且不必要,因此精简了。只对链表做很简单的介绍,有一定了解

即可。

⑥ 文件只作简单介绍，有初步概念即可。

⑦ 由于许多学校把C语言的教学安排在一年级，而学生还未学完高等数学，因此本书不包括有关高等数学知识的例题。考虑到有部分读者在学习高等数学后可能对这方面的内容感兴趣，在习题部分列出有关的题目(如用二分法和牛顿迭代法求一元方程的根)，并在《C程序设计教程(第3版)学习辅导》中给出介绍和程序，可供自学参考。

⑧ 在各章节标题前加“*”号的，是比较深入的内容，在教学时可以不讲，由学生自学参考。

相信经过修改后，本书会更加容易学习，读者的基本功会更扎实，效果会更好。

7. 作者的心中永远要装着读者

作者从1978年开始从事计算机基础教育和计算机普及工作，40年来一直奋斗在这个平凡而重要的岗位上，把自己的后半生贡献给了我国的计算机教育和计算机普及事业，对这个事业有着深厚的感情和深切的体会。我最大的愿望是“把计算机从少数计算机专家手中解放出来，成为广大群众手中的工具”，使广大群众轻松愉快、兴趣盎然地进入计算机的天地。经过40年的努力，这个愿望正在变成现实。

我始终认为，作者心中要永远装着读者，处处为读者着想，和读者将心比心，善于换位思考。我在编写教材时，常常反问自己：“读者读到这里时会提出什么问题？”

“怎样讲才能使读者更容易明白？”我常常用这样一句话来鞭策自己：“只有明白不明白的人为什么不明白的人才是明白人。”写书不仅是简单地把有关的技术内容告诉读者，而且要考虑怎样写才能使读者容易理解。我写书，有一半的时间用来研究和处理技术方面的问题，还有一半时间用来考虑怎样讲才能使学生易于理解。有时为了找到一个好的例子或一个通俗的比喻，往往苦苦思索好几天，每一句话都要反复斟酌推敲。要善于把复杂问题简单化，而不能把简单问题复杂化。写一本书容易，写一本好书不容易，能讲一堂课很容易，要讲好一堂课并不容易，需要下很大的功夫。

这就要求我们(作者和老师)要深入了解自己工作的对象，有的放矢，准确定位；要根据应用的需要，合理取舍，精选内容；要认真研究学习者的认识规律，采用他们容易理解的方法，深入浅出，通俗易懂。清华大学一位已故的院士说得好：“什么叫水平高？只有能用通俗易懂的方法和语言阐述清楚复杂概念的人，才是水平高。那些把概念搬来搬去的人，不能算水平高。”

多年来，在广大师生的关心和支持下，我努力去做，取得一些成绩。有人称我是计算机界的“平民作家”。我乐于接受并很珍惜群众送给我的这一称谓，这是对我的莫大鞭策。希望所有的教师和作者共同努力，把每一本书、每一门课程都做成精品，得到千万学生和读者的肯定和赞扬，这才是对我们辛劳的最高奖赏。

为了帮助读者学习本书，作者还编写了一本《C程序设计教程(第3版)学习辅导》，提供本书中各章习题的参考答案以及上机实践指导。该书由清华大学出版社于2018年出版。

南京大学金莹副教授、薛淑斌高级工程师和谭亦峰工程师参加了本书的策划、调研、收集资料、研讨以及编写部分程序的工作。

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和指正。

谭浩强谨识

2018年3月

于清华园

第 1 章 程序设计和 C 语言	1
1.1 计算机与程序、程序设计语言	1
1.2 C 语言的特点	2
1.3 简单的 C 语言程序	3
1.4 C 语言程序的结构	6
1.5 运行 C 程序的步骤与方法	8
1.6 程序设计的任务	10
1.7 算法——程序的灵魂	11
1.7.1 程序是什么	11
1.7.2 什么是算法	12
1.7.3 怎样表示一个算法	13
1.8 结构化程序设计方法	19
1.9 学习程序设计,培养科学思维	21
本章小结	23
习题	23
第 2 章 最简单的 C 程序设计——顺序程序设计	25
2.1 顺序程序设计举例	25
2.2 数据的类型及存储形式	29
2.2.1 C 语言的数据类型	29
2.2.2 数据的表现形式——常量和变量	29
2.2.3 整型数据	31
2.2.4 字符型数据	36
2.2.5 浮点型数据	41
2.3 用表达式进行数据的运算	43
2.3.1 C 表达式	43
2.3.2 C 运算符	44
2.3.3 运算符的优先级与结合性	46

2.3.4	不同类型数据间的混合运算	46
*2.3.5	强制类型转换	47
2.4	最常用的 C 语句——赋值语句	48
2.4.1	C 语句综述	48
2.4.2	赋值表达式	50
2.4.3	赋值语句	53
2.5	数据的输入输出	56
2.5.1	C 语言中输入输出的概念	56
2.5.2	用 printf 函数输出数据	57
2.5.3	用 scanf 函数输入数据	62
2.5.4	字符数据的输入输出	65
	本章小结	67
	习题	68
第 3 章	选择结构程序设计	71
3.1	简单的选择结构程序	71
3.2	选择结构中的关系运算	73
3.2.1	关系运算符及其优先次序	73
3.2.2	关系表达式	73
3.3	选择结构中的逻辑运算	74
3.3.1	逻辑运算符及其优先次序	75
3.3.2	逻辑表达式	76
3.4	用 if 语句实现选择结构	78
3.4.1	if 语句的三种形式	78
3.4.2	if 语句的嵌套	80
3.5	用条件表达式实现选择结构	83
3.6	利用 switch 语句实现多分支选择结构	86
3.7	选择结构程序综合举例	88
	本章小结	94
	习题	94
第 4 章	循环结构程序设计	96
4.1	程序需要循环	96
4.2	用 while 语句和 do...while 语句实现循环	96
4.2.1	用 while 语句实现循环	96
4.2.2	用 do...while 语句实现循环	98
4.2.3	while 循环和 do...while 循环的比较	99
4.2.4	递推与迭代	101
4.3	用 for 语句实现循环	104

4.3.1	for 语句的执行过程	104
4.3.2	for 语句的各种形式	106
4.3.3	for 循环应用举例	108
4.4	循环的嵌套	110
4.5	用 break 语句和 continue 语句改变循环状态	110
4.5.1	用 break 语句提前退出循环	110
4.5.2	用 continue 语句提前结束本次循环	111
4.6	几种循环的比较	113
4.7	循环程序举例	113
	本章小结	116
	习题	117
第 5 章	利用数组处理批量数据	119
5.1	数组的作用	119
5.2	怎样定义和引用一维数组	120
5.2.1	怎样定义一维数组	120
5.2.2	怎样引用一维数组元素	120
5.2.3	一维数组的初始化	121
5.2.4	利用一维数组的典型算法——递推与排序	122
5.3	怎样定义和引用二维数组	125
5.3.1	怎样定义二维数组	125
5.3.2	怎样引用二维数组的元素	126
5.3.3	二维数组程序举例	126
5.3.4	二维数组的初始化	129
5.4	利用字符数组处理字符串数据	130
5.4.1	怎样定义字符数组	130
5.4.2	字符数组的初始化	131
5.4.3	引用字符数组的元素	132
5.4.4	字符串和字符串结束标志	133
5.4.5	字符数组的输入输出方法	135
5.4.6	有关字符处理的算法	136
5.4.7	利用字符串处理函数	139
	本章小结	142
	习题	143
第 6 章	利用函数进行模块化程序设计	145
6.1	为什么要使用函数	145
6.1.1	函数是什么	145
6.1.2	程序和函数	146