

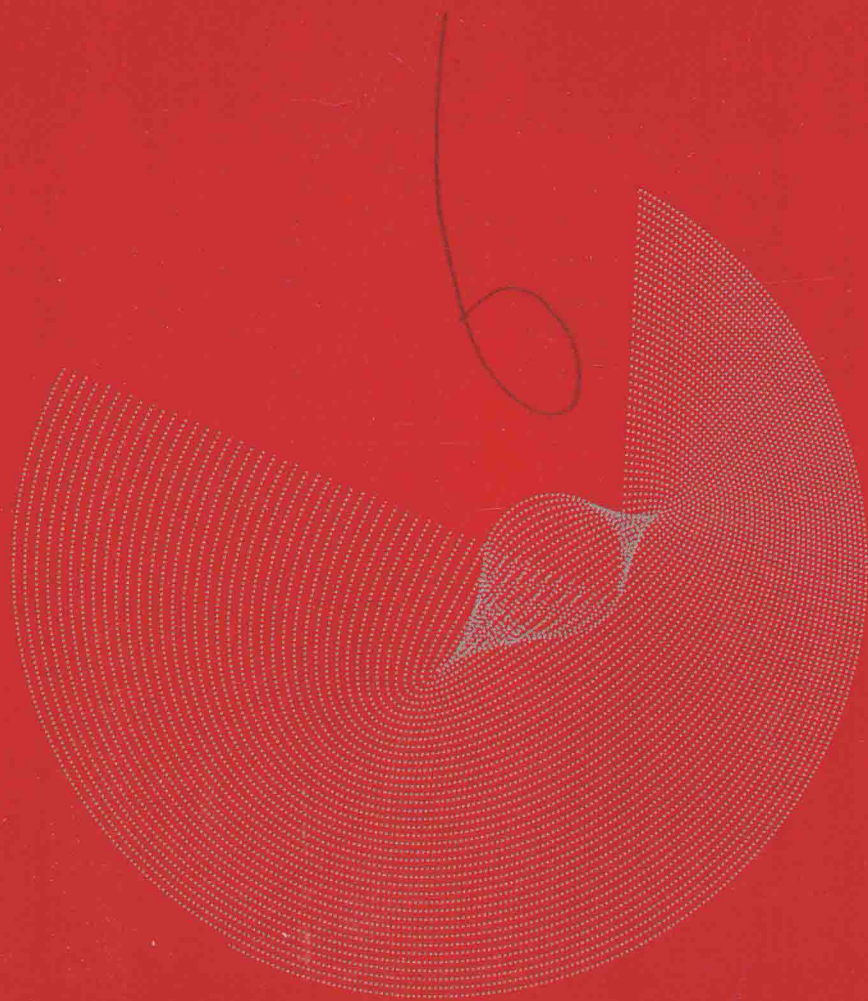
全国高校出版社优秀畅销书一等奖

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(第3版)

谭浩强 编著



清华大学出版社



全国

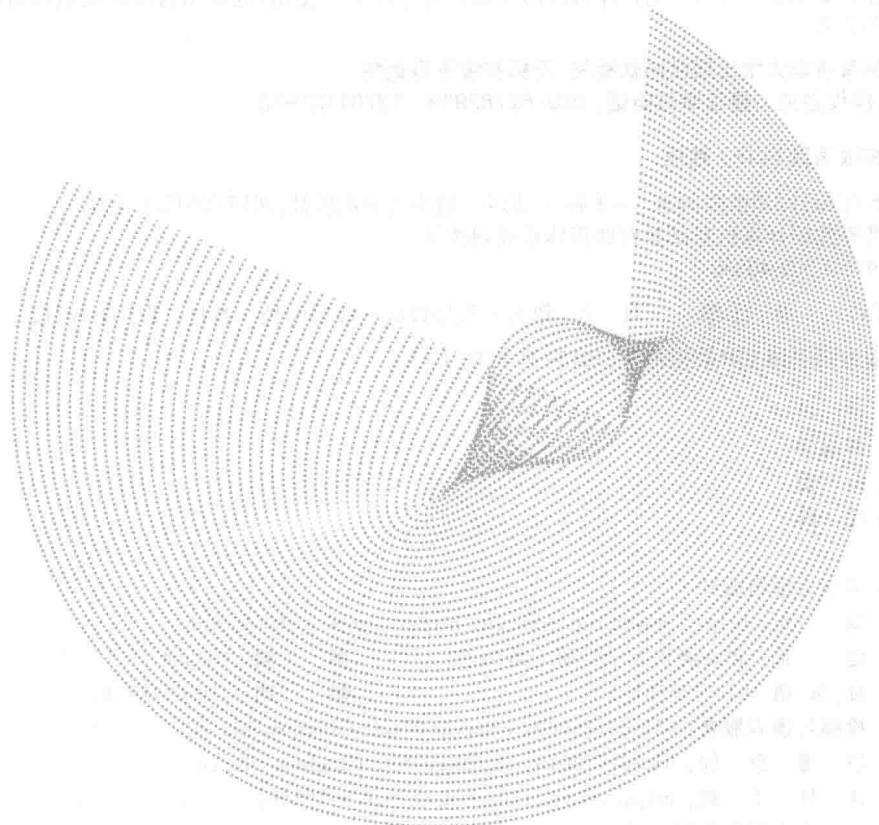
一等奖

中国高等院校计算机基础课程规划教材

丛书主编 谭浩强

C++ 程序设计 (第3版)

谭浩强 编著



清华大学出版社
北京

内 容 简 介

C++是近年来国内外广泛使用的现代计算机语言,它既支持基于过程的程序设计,也支持面向对象的程序设计。国内许多高校陆续开设了C++程序设计课程。但是,由于C++涉及概念很多,语法比较复杂,内容十分广泛,使不少人感到学习难度较大,难以入门。

本书作者深入调查了大学的程序设计课程的现状和发展趋势,参阅了国内外数十种有关C++的教材,认真分析了读者在学习中的困难和认识规律,设计了读者易于学习的教材体系,于2004年出版了《C++程序设计》一书。该书降低入门起点,不需要C语言的基础,从零起点介绍程序设计和C++。广大师生用后反映非常好,认为该书定位准确,概念清晰,深入浅出,取舍合理,以通俗易懂的语言对C++的许多难懂的概念作了透彻而通俗的说明,大大降低了初学者学习的困难,是初学者学习C++的一本好教材。

根据教学实践的需要,作者在2011年对该书进行了修订,出版了《C++程序设计(第2版)》,现在又进行了一次修订,出版了《C++程序设计(第3版)》,内容更加丰富,讲解更加清晰,学习更加容易,依据ANSI C++标准进行介绍,引导读者从一开始就按C++的要求编程,而不是过多地迁就C语言的习惯。全书分为4篇:基本知识、基于过程的程序设计、基于对象的程序设计和面向对象的程序设计。

为了便于教学,本书有《C++程序设计题解和上机指导(第3版)》和《C++程序设计实践指导》两本配套教材,旨在帮助学生通过实践掌握C++的编程方法。

本书内容全面,例题丰富,概念清晰,循序渐进,易于学习,即使没有教师讲授,读者也能看懂本书的大部分内容。本书是学习C++的入门教材,可供各类专业学生使用,也可作为计算机培训班的教材以及读者自学参考。

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/谭浩强编著. —3版. —北京:清华大学出版社,2015(2015.8重印)

中国高等院校计算机基础教育课程体系规划教材

ISBN 978-7-302-40830-7

I. ①C… II. ①谭… 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(2015)第153321号

责任编辑:张民

封面设计:傅瑞学

责任校对:白蕾

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:30.75

字 数:716千字

版 次:2004年6月第1版

2015年8月第3版

印 次:2015年8月第2次印刷

印 数:3001~8000

定 价:49.50元

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业绩

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

祝賀譚浩強教授著作創世界發行紀錄

光榮的紀錄
傑出的業績

一九九五年九月

盧嘉錫

▲ 原全国人民代表大会副委员长、中国科学院院长卢嘉锡给谭浩强教授的题词

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),由清华大学出版社出版。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。在此基础上,根据计算机基础教育的发展,全国高等院校计算机基础教育研究会先后多次发布了 CFC 的新

版本。

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。

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

前 言

FOREWORD

20世纪90年代,我曾经写过一本《C程序设计》(现已更新至第四版),由清华大学出版社出版。该书出版后,社会各界反映强烈,许多人说,C语言原来是比较难学的,自从《C程序设计》出版后,C语言变得不难学了。该书已先后重印200多次,累计发行1250多万册,创造了科技书籍的最高纪录,成为广大初学者学习C语言的主要用书。这使我深受鼓舞和鞭策,感受到广大读者的殷切期望,看到了计算机教育工作者身上的重任。

近年来,支持面向对象程序设计的C++语言迅速地在国内推广,不少高校开设了C++课程,由于C++涉及概念很多,语法比较复杂,内容十分广泛,不少人(尤其是非计算机专业的初学者)感到学习难度较大,难以入门。许多热情的读者希望我能在《C程序设计》的基础上,再写一本易于入门的《C++程序设计》教材,以帮助更多的初学者更顺利地迈进C++的大门。我于2004年写成了《C++程序设计》一书,由清华大学出版社正式出版。该书出版后,受到各高校的欢迎,许多师生认为该书定位准确,概念清晰,深入浅出,取舍合理,以通俗易懂的语言对C++许多难懂的概念作了透彻而通俗的说明,大大降低了初学者学习的困难程度,是一本初学者学习C++的好教材。

根据教学的实践,作者于2011年对该书进行了修订,出版《C++程序设计(第2版)》,现在又在此基础上修订出版《C++程序设计(第3版)》,讲解更加清晰,使学习更加容易。

要写好和教好C++程序设计,决不是一个纯技术问题,需要综合考虑多方面因素,作出合理的决策和安排。作者调查了我国大学的程序设计课程的现状和发展趋势,了解了国内外C++的教学和使用情况,认真分析了学习者在学习过程中遇到的困难,研究了初学者的认识规律,并且参阅了国内外数十种有关C++的教材,形成了以下几点看法,并体现在本教材中。

(1)教材必须准确定位,要求恰当,合理取舍。写书首先要弄清楚本书是为什么人写的,他们学习C++的目的和要求是什么?应当学到什么程度?

目前学习C++的包括以下三部分人:

- ① 当前需要用C++编写面向对象程序的软件开发人员。
- ② 将来有可能成为软件开发人员,需要熟练掌握面向对象程序设计的知识和C++语言工具,打好进一步发展基础的人员,其中包括部分计算机专业的学生。

③ 希望初步学习面向对象程序设计的知识,了解用 C++ 语言进行面向对象程序设计方法的人员。这部分人中的大多数将来并不是专业的软件开发人员,不要求熟练掌握 C++ 语言进行程序设计。

应当说明:本书的对象不是 C++ 软件开发专业人员,而是大学各专业(尤其是非计算机专业)的大学生,他们目前尚无程序设计的实际经验,将来也不一定从事 C++ 程序开发。他们对 C++ 程序设计的学习是入门性、基础性的,主要是初步了解软件开发的方法,了解 C++ 语言的特点,扩大知识面,以利于将来的工作。

众所周知,研发 C++ 的初衷是为了解决大型软件开发中遇到的问题,提高软件的开发效率。只有参加过研制大型软件的人才能真正体会到 C++ 的优越性,并善于利用 C++ 的独特机制进行软件开发。

对多数大学生(尤其是非计算机专业的学生)在校期间初学 C++ 程序设计的要求要恰当。主要是初步学习面向对象程序设计方法,能够初步利用 C++ 编写简单的程序,为以后(如果需要的话)进一步学习和应用打下初步的基础。不可能设想,通过几十个小时的学习,能使一个没有程序设计基础的人变成一个熟练的 C++ 开发人员。应当有一个实事求是的分析和估计。因此,本书着力于使读者对面向对象程序设计的概念有清晰的了解,对 C++ 语言的全貌和使用方法有基本的认识,用容易理解的方法讲清楚有关的基本概念和基本方法,而不去罗列 C++ 语法中各种烦琐的细节。否则将会使篇幅过大(常见国外的书厚达近千页),而且会使读者感到枯燥无味,冲淡重点,主次不分。

如果以后从事 C++ 程序开发工作,应当在此基础上再深入学习 C++ 程序设计提高课程。

(2) 需要选好学习 C++ 的切入点。对于怎样介绍 C++, 国内外的教材有多种模式。有的不介绍基于过程的程序设计,一开始就直接介绍类和对象,有的先介绍基于过程的程序设计,然后介绍面向对象的程序设计,各有特点。作者认为:不应当把基于过程和面向对象的程序设计对立起来,任何程序设计都需要用到过程化的知识。作为一个程序设计人员,要掌握过程化的程序设计,也要掌握面向对象的程序设计。以前,人们通过 C 语言学习过程化程序设计,再通过 C++ 学习面向对象的程序设计。其实,C++ 并不是纯粹的面向对象的语言,它是一种混合语言。学习 C++ 既可以编写过程化的程序,也可以编写面向对象的程序。实际上,C 语言相当于 C++ 的过程化部分。

在参考和比较国内外多种教材的基础上,作者确定了本书的结构。全书分为 4 部分:第 1 部分介绍 C++ 的基本知识和基本语法;第 2 部分介绍 C++ 基于过程的程序设计;第 3 部分介绍 C++ 基于对象的程序设计;第 4 部分介绍 C++ 面向对象的程序设计。

许多教师的实践表明,以基于过程的程序设计作为切入点,从编写简单的程序开始,循序渐进,由基于过程到面向对象,逐步深入,比较符合读者的认识规律,每一步的台阶都比较小,学习难度不大,读者容易理解。

这样,人们既可以通过 C 语言学习基于过程的程序设计,也可以直接通过 C++ 语言学习基于过程和面向对象的程序设计。因此,本书实际上是一本 C/C++ 教材。

(3) 设计合适的教材体系。C++ 程序设计涉及面向对象程序设计的理论、C++ 语言的语法以及算法 3 方面的内容,其中每一方面都包含十分丰富的内容,都可以分别单

独成书。显然在一本教材中深入、详细地介绍以上3个方面的知识是不可能的,必须把它们有机地结合起来,综合应用。不同的书对此采取不同的写法,侧重点有所不同,各有道理,也各有优缺点,适合于不同的读者。需要在教学实践中检验,取长补短,不断完善。

作者认为:要进行C++程序设计,当然需要了解面向对象程序设计的有关概念,但是本课程毕竟不是一门面向对象程序设计的理论课程,在本书中不是抽象地介绍面向对象程序设计的理论,而是结合C++的使用自然而然地引出面向对象程序设计的有关概念,通过C++的编程过程理解面向对象程序设计方法。在介绍程序设计的过程中,介绍有关算法,引导读者思考怎样构造一个算法。

要用C++编程序,最基本的要求是正确掌握和运用C++,由于C++语法复杂,内容又多,如果对它缺乏系统了解,将难以真正应用,编出来的程序将会错误百出,编译出错,事倍功半。本书的做法是比较全面地介绍C++的主要特点和功能,引导读者由简而繁地学会编写C++程序。有了C++编程的初步基础后,再进一步提高。这样的方法可能符合大多数学习者的情况。

作者认为:决不能使读者陷于语法细节的汪洋大海之中。在教学中,对于C++的众多功能和语法细节,在初学阶段没有必要全部细讲,必须有所选择,合理取舍,不应追求深而全。建议读者对于语言的细节不必深究,更不必死记,知道怎么用即可。对有些细节,可以在以后编写程序遇到问题时回过头再查阅本书的介绍,这样可以有效地降低当前学习的难度。

(4)重要的是要建立清晰的概念。由于人们习惯于过程化的编程方法,在开始学习时往往对面向对象的原理和实现机制理解不深,本书把面向对象程序计划分为两个有机联系的阶段——基于对象的程序设计和面向对象的程序设计。从类和对象开始逐步深入地引出面向对象程序设计的各个概念以及用C++实现的方法。凡引入一个新的概念,都作了通俗而透彻的讲解,把来龙去脉交代清楚,讲清楚“这是什么”,“为什么需要它”,“怎样使用它”,并举例说明。使读者建立起清晰的概念,知其然也知其所以然,而不是盲目地照葫芦画瓢。

本书是入门性、基础性的教材,任务是帮助读者顺利入门,打好基础。有了好的基础,以后根据需要再深入学习和实践,是不困难的。

(5)努力化解难点,把复杂的问题简单化。为了使学习C++更容易,作者在写作过程花了很大的精力去考虑怎样使读者易于接受和理解。作者一贯认为,教材编著者应当与读者换位思考,要站在读者的立场上思考问题,帮助他们排除学习中的困难,要善于把复杂的问题简单化,而不应把简单的问题复杂化,要善于化解难点,深入浅出。一定不要难倒读者,更不应吓跑读者。我们的任务是要使“难”变成“不难”,循序渐进地引导初学者进入C++的大门。

在学习过程中不应被一大堆高深莫测的名词术语吓唬住,有些问题看起来很深奥,其实换一个角度解释就很容易理解,甚至用一个通俗的例子就可以把问题说清楚。本书尽量用通俗易懂的方法和语言叙述复杂的概念,力求减少初学者学习C++的困难。

为了便于读者理解,本书在介绍编程例题时,一般采取以下步骤展开:给出任务—解题思路—编写程序—运行结果—程序分析—说明与注意。以帮助读者清晰地掌握程

序设计的思路与方法。

本书便于自学,即使没有老师讲解,读者也能看懂本书的大部分内容。这样,老师教学时就可以不必完全按照教材的内容和顺序详细讲解,而可以有选择地重点讲授,其余内容由学生自学。

(6) 按照教学的特点组织教材。不应当把教材写成手册,教材的任务是用读者容易理解的方法讲清基本的概念和方法,而手册的任务则是给出一个包罗万象的备查资料。读者在学习时应重点掌握基本的概念和方法,以后如果从事软件开发,在遇到具体问题时,进一步学习或查阅一下手册即可。

本教材所举的程序,是从教学的角度考虑的,是为了帮助读者更好地理解和应用某一方面的教学内容而专门编写的,并不一定是实际应用的程序。一个实际的C++程序需要考虑许多因素,综合各部分知识,有许多注释行,而且一般是多文件的程序,篇幅较长,往往不适合作为教学程序。教学程序对问题作了简化,尽量压缩不必要的语句,减少注释行,篇幅一般不长,力求使读者能读懂程序。有些在专业人员看来很“幼稚”的程序,在学习者看来可能是一个很合适的教学程序。教材必须通俗易懂,使人能看懂。在初步掌握C++编程方法后,可以逐步使程序复杂些,长一些,更接近真实程序一些。

考虑到教学的不同要求,我们提供了3个层次的程序实例:(1)教材各章中的例题。这是最基本的,对此作了比较详细的分析介绍,读者容易理解与掌握;(2)教材各章所给出的习题。由于教材的篇幅有限,有些很好的例子无法在教材中列出,则将其作为习题,习题的难度比例题大一些,希望读者在学习教材的基础上自己完成。在本书的配套教材《C++程序设计题解与上机指导(第3版)》中提供了全部习题的参考解答。教师可以从中选择一些习题作为例题讲授。建议读者除了完成教师指定的习题外,把习题解答中的程序全部看一遍,以更好地理解C++程序。(3)为了提供更丰富、更贴近实际的实例,在本书的另一本配套教材《C++程序设计实践指导》中提供了一批C++应用程序。教师可以指定学生阅读其中一些程序实例,也可选择一些在课堂上讲授。使学生扩大眼界,启迪思路,丰富知识,增长能力。

关于如何使用本教材:

(1) 学习本教材可以有两个入口:未学过C语言的读者从第1章学起;已学过C语言的读者可以从第8章学起。由于C++是从C语言发展而来的,它保留了C语言的大部分内容。本书第1~7章介绍的内容主要是C++从C语言继承来的部分,因此,学过C语言的读者可以跳过这几章,而直接学习第8章。但最好在学习第8章之前,简单浏览一下前7章,以对C++与C的异同有所了解。尤其应看一下前7章中带星号(*)的部分,它们是C++对C语言的发展,是C语言中没有的。

由于《C程序设计(第四版)》一书已比较成熟,读者反映很容易看懂,因此本书前7章主要是根据《C程序设计(第四版)》一书进行改写的。这样,把C语言和C++很自然地衔接起来,无论学过C语言还是未学过C语言的读者都会感到本教材容易入门,易于学习。

(2) 本教材提供两本配套参考书:

① 《C++程序设计题解与上机指导(第3版)》,谭浩强编著,清华大学出版社出

版。除了提供教材各章中的全部习题解外，还介绍在两种典型的环境下运行 C++ 程序的方法，一种是 Windows 环境下的 Visual C++ 2010，一种是 GCC。GCC 是自由软件，可以在 Windows 环境或非 Windows 环境(如 DOS, UNIX, Linux)下使用。此外，该书还给出上机实践任务，指导学生完成课后上机实践。

② 《C++ 程序设计实践指导》，谭浩强主编，陈清华、朱红编著，清华大学出版社出版。书中精心选择了五十多个不同类型、不同难度的 C++ 程序。可以供学生进一步深入学习的参考，以提高编程能力。

本书由谭浩强编著，薛淑斌和谭亦峰高级工程师参加了教学研讨、大纲讨论、收集材料、调试部分程序及部分编写工作。全国高等院校计算机基础教育研究会以及全国各高校老师几年来对本书的编写始终给予了热情的支持，清华大学出版社对本书的出版十分重视，使本书在短时间内得以出版。对一切曾经鼓励、支持和帮助过我的领导、组织、专家、朋友和读者，在此谨表示真挚的谢意。

本书肯定会有不妥甚至错误之处，诚盼专家和广大读者不吝指正。

谭浩强

2015年5月1日于清华园

第1篇 基本知识

第1章 C++的初步知识	3
*1.1 从C到C++	3
*1.2 最简单的C++程序	5
1.3 C++程序的构成和书写形式	12
1.4 C++程序的编写和实现	13
1.5 关于C++上机实践	15
习 题	15
第2章 数据的存储、表示形式和基本运算	18
2.1 C++的数据类型	18
2.2 常量	20
2.2.1 什么是常量	20
2.2.2 数值常量	20
2.2.3 字符常量	22
2.2.4 符号常量	25
2.3 变量	26
2.3.1 什么是变量	26
2.3.2 变量名规则	26
2.3.3 定义变量	27
2.3.4 对变量赋初值	28
2.3.5 常变量	28
2.4 C++的运算符	29
2.5 算术运算符与算术表达式	30
2.5.1 基本的算术运算符	30
2.5.2 算术表达式和运算符的优先级与结合性	30
2.5.3 表达式中各类数值型数据间的混合运算	30

2.5.4	自增(++)和自减(--)运算符	31
2.5.5	强制类型转换运算符	32
2.6	赋值运算符和赋值表达式	33
2.6.1	赋值运算符	33
2.6.2	赋值过程中的类型转换	33
2.6.3	复合赋值运算符	35
2.6.4	赋值表达式	35
2.7	逗号运算符和逗号表达式	36
	习题	37

第2篇 基于过程的程序设计

第3章	程序设计初步	41
3.1	基于过程的程序设计和算法	41
3.1.1	算法的概念	41
3.1.2	算法的表示	42
3.2	C++的程序结构和C++语句	43
3.3	赋值操作	46
3.4	C++的输入与输出	46
*	3.4.1 输入流与输出流的基本操作	47
*	3.4.2 在标准输入流与输出流中使用控制符	49
	3.4.3 用 getchar 和 putchar 函数进行字符的输入和输出	52
	3.4.4 用 scanf 和 printf 函数进行输入和输出	53
3.5	编写顺序结构的程序	54
3.6	关系运算和逻辑运算	55
	3.6.1 关系运算和关系表达式	55
*	3.6.2 逻辑常量和逻辑变量	57
	3.6.3 逻辑运算和逻辑表达式	58
3.7	选择结构和 if 语句	60
	3.7.1 if 语句的形式	60
	3.7.2 if 语句的嵌套	63
	3.7.3 条件运算符和条件表达式	64
	3.7.4 多分支选择结构与 switch 语句	65
	3.7.5 编写选择结构的程序	67
3.8	循环结构和循环语句	69
	3.8.1 用 while 语句构成循环	70
	3.8.2 用 do-while 语句构成循环	71
	3.8.3 用 for 语句构成循环	72
	3.8.4 循环的嵌套	74

3.8.5	提前结束循环(break 语句和 continue 语句)	74
3.8.6	编写循环结构的程序	75
	习题	80
第 4 章	利用函数实现指定的功能	83
4.1	什么是函数	83
4.1.1	为什么需要函数	83
4.1.2	函数调用举例	84
4.1.3	函数的分类	85
4.2	定义函数的一般形式	85
4.2.1	定义无参函数的一般形式	85
4.2.2	定义有参函数的一般形式	86
4.3	函数参数和函数的值	86
4.3.1	形式参数和实际参数	86
4.3.2	函数的返回值	88
4.4	函数的调用	89
4.4.1	函数调用的一般形式	89
4.4.2	函数调用的方式	89
4.4.3	对被调用函数的声明和函数原型	90
4.5	函数的嵌套调用	92
4.6	函数的递归调用	95
*4.7	内置函数	98
*4.8	函数的重载	100
*4.9	函数模板	102
*4.10	有默认参数的函数	104
4.11	局部变量和全局变量	106
4.11.1	局部变量	106
4.11.2	全局变量	107
4.12	变量的存储类别	109
4.12.1	动态存储方式与静态存储方式	109
4.12.2	自动变量	110
4.12.3	用 static 声明静态局部变量	110
4.12.4	用 register 声明寄存器变量	113
4.12.5	用 extern 声明外部变量	113
4.12.6	用 static 声明静态外部变量	115
4.13	变量属性小结	115
4.14	关于变量的声明和定义	118
4.15	内部函数和外部函数	119
4.15.1	内部函数	119

4.15.2 外部函数	119
4.16 头文件	121
* 4.16.1 头文件的内容	121
4.16.2 关于 C++ 标准库和头文件的形式	121
习题	122
第 5 章 利用数组处理批量数据	124
5.1 为什么需要用数组	124
5.2 定义和引用一维数组	125
5.2.1 定义一维数组	125
5.2.2 引用一维数组的元素	125
5.2.3 一维数组的初始化	126
5.2.4 一维数组程序举例	126
5.3 定义和引用二维数组	128
5.3.1 定义二维数组	129
5.3.2 引用二维数组的元素	130
5.3.3 二维数组的初始化	130
5.3.4 二维数组程序举例	131
5.4 用数组作函数参数	133
5.5 字符数组	137
5.5.1 定义和初始化字符数组	137
5.5.2 字符数组的赋值与引用	138
5.5.3 字符串和字符串结束标志	139
5.5.4 字符数组的输入输出	140
5.5.5 使用字符串处理函数对字符串进行操作	141
5.5.6 字符数组应用举例	144
* 5.6 C++ 处理字符串的方法——字符串类与字符串变量	145
5.6.1 字符串变量的定义和引用	146
5.6.2 字符串变量的运算	147
5.6.3 字符串数组	147
5.6.4 字符串运算举例	148
习题	151
第 6 章 善于使用指针与引用	153
6.1 什么是指针	153
6.2 变量与指针	155
6.2.1 定义指针变量	155
6.2.2 引用指针变量	157
6.2.3 用指针作函数参数	159

6.3	数组与指针	164
6.3.1	指向数组元素的指针	164
6.3.2	用指针变量作函数形参接收数组地址	166
6.4	字符串与指针	169
6.5	函数与指针	171
6.6	返回指针值的函数	172
6.7	指针数组和指向指针的指针	173
6.7.1	指针数组	173
* 6.7.2	指向指针的指针	175
* 6.8	const 指针	177
* 6.9	void 指针类型	179
6.10	有关指针的数据类型和指针运算的小结	180
6.10.1	有关指针的数据类型的小结	180
6.10.2	指针运算小结	180
* 6.11	引用	182
6.11.1	什么是变量的引用	182
6.11.2	引用的简单使用	183
6.11.3	引用作为函数参数	184
	习题	188
第7章	用户自定义数据类型	190
7.1	结构体类型	190
7.1.1	为什么需要用结构体类型	190
7.1.2	结构体类型变量的定义方法及其初始化	191
7.1.3	引用结构体变量	193
7.1.4	结构体数组	195
7.1.5	指向结构体变量的指针	197
7.1.6	结构体类型数据作为函数参数	200
* 7.1.7	用 new 和 delete 运算符进行动态分配和撤销存储空间	203
7.2	枚举类型	205
7.3	用 typedef 声明新的类型名	208
	习题	211

第3篇 基于对象的程序设计

第8章	类和对象的特性	215
8.1	面向对象程序设计方法概述	215
8.1.1	什么是面向对象的程序设计	215
8.1.2	面向对象程序设计的特点	219