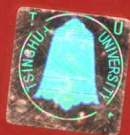
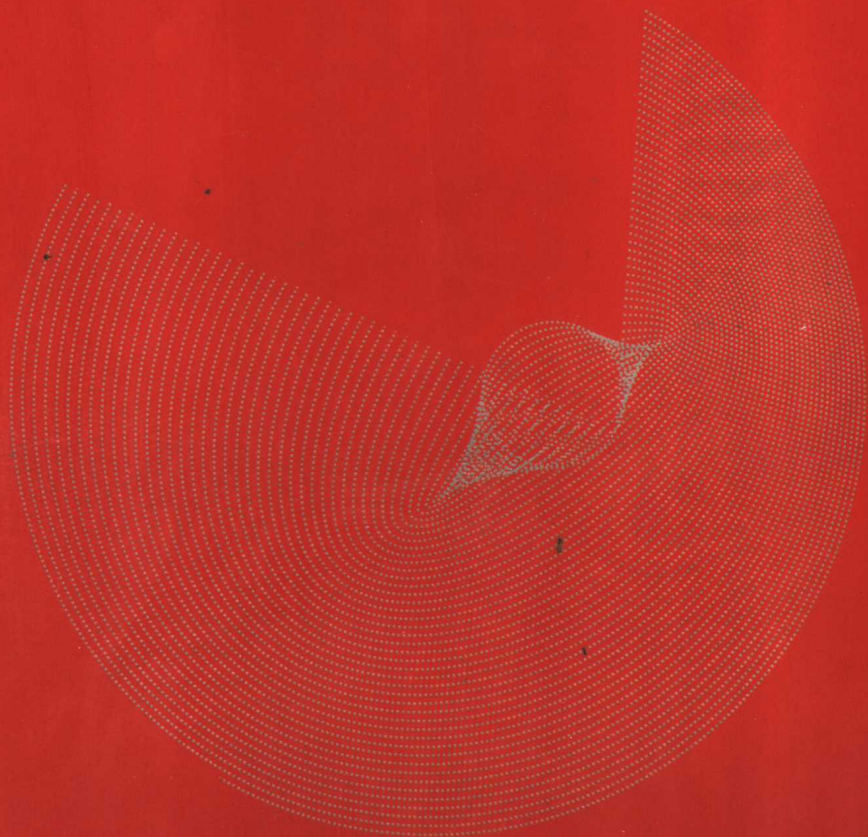


中国高等院校计算机基础教育课程体系规划教材

C++ 程序设计

谭浩强 编著

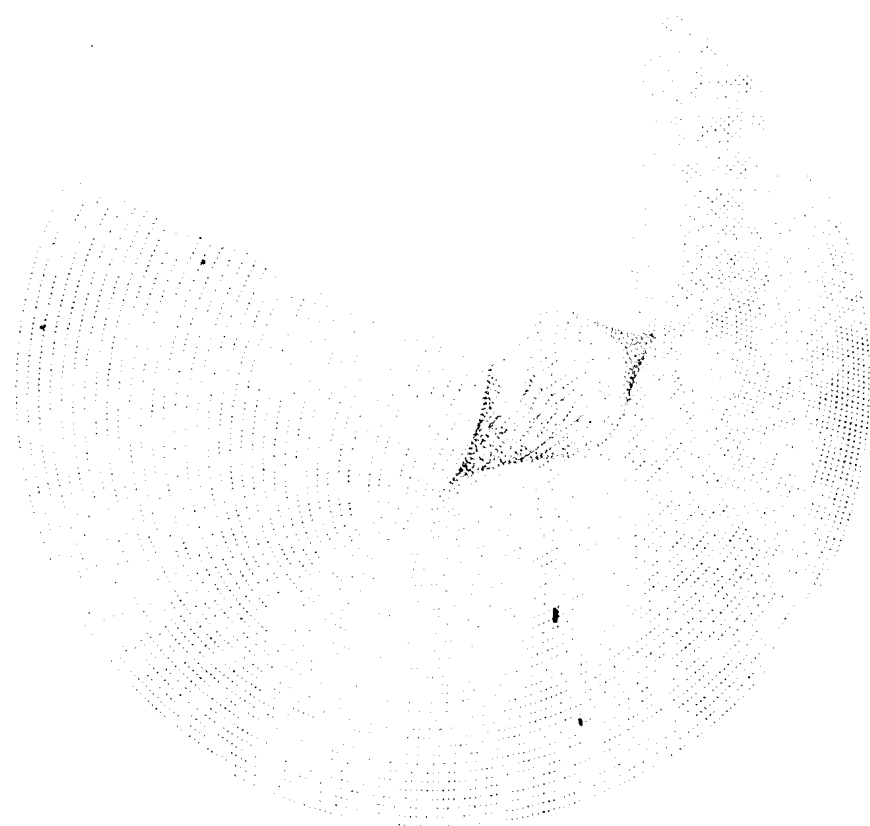


清华大学出版社

中国高等院校计算机基础教育课程体系规划教材

C++ 程序设计

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,它既支持面向过程的程序设计,也支持基于对象和面向对象的程序设计。国内许多高校已陆续开设了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我国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。在本书中做到准确定位,合理取舍内容,设计了读者易于学习的教材体系,并且以通俗易懂的语言化解了许多复杂的概念,大大减少了初学者学习C++ 的困难。

考虑到许多读者在学习C++ 前未学过其他语言,本书把入门起点降低,读者不需要具备C 语言的基础。本书包括14 章,分为4 篇:基本知识;面向过程的程序设计;基于对象的程序设计;面向对象的程序设计。本书内容全面,例题丰富,概念清晰,循序渐进,易于学习。

本书是依据ANSI C++ 标准进行介绍的,引导读者从一开始就按标准C++ 的规定编程。本书配有两本辅导教材,即《C++ 程序设计题解与上机指导》和《C++ 编程实践指导》。

本书是大学各专业学生学习C++ 的基础教材,也是适于初学者学习的教材。即使没有教师讲授,读者也能看懂本书的大部分内容。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C++ 程序设计/谭浩强编著. —北京:清华大学出版社,2004

(中国高等院校计算机基础教育课程体系规划教材)

ISBN 7-302-08599-4

I. C… II. 谭… 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(2004)第041528 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

组稿编辑:焦金生 张 民

文稿编辑:张 民

封面设计:孟繁聪

版式设计:肖 米

印 刷 者:北京市昌平环球印刷厂

装 订 者:北京市密云县京文制本装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:32.25 字数:755 千字

版 次:2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷

书 号:ISBN 7-302-08599-4/TP·6169

印 数:1~10000

定 价:36.00 元

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

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而成就

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

编审委员会

主 任：谭浩强

委 员：(按姓氏笔画为序)

冯博琴 刘瑞挺 吴文虎 张 龙

张 森 高 林 焦金生

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。特别是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。高校开展计算机基础教育的状况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) 从 21 世纪初开始,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,于 2004

年春提出了新的高校计算机基础教育改革思路和课程方案，并编写了《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。该课题受到各方面的关注、支持和欢迎，大家一致认为 CFC 2004 提出了一个既体现先进又切合实际的思路和解决方案。

为了实现课题研究组提出的要求，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。要十分明确本教材是为哪一部分读者写的，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 2004 的思路和课程要求。本套教材的作者多数是课题研究组的成员或参加过课题研讨的专家，对计算机基础教育改革的方向和思路有深切的体会和清醒的认识。因而可以说，本套教材是 CFC 2004 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

2004年5月

前言

FOREWORD

十几年前，我曾经写过一本《C 程序设计》(清华大学出版社出版)，当时只是为了教学的需要。由于许多人反映 C 语言难学，我力图编写出一本容易入门、容易学习的教材，以解决教学中的困难。该书出版后，反映很强烈，不少人说，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 100 多次，累计发行 600 多万册，平均每年发行 50 万册，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，我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++ 牵涉概念很多，语法比较复杂，内容十分广泛，不少人感到学习难度较大，难以入门。许多热情的读者多次向我提出，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》，以帮助更多的初学者更顺利地迈进 C++ 的大门。我感到热情难却，责任重大，但由于工作和活动较忙，心有余而力不足，迟迟未能动笔。在各高校老师和出版社同仁的多次催促下，我于 2002 年 1 月 1 日午夜开始动笔，写出了第一页。其后利用工作和活动的空隙断断续续地进行，2003 年春季，北京为防治非典(SARS)停止一切集会，使我有机会集中时间和精力，加快了进度。其间数易其稿，终于在 2004 年春完成了第 4 稿。

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，做出合理的决策和安排。我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，合理取舍内容。写书首先要弄清楚本书是为什么人写的，他们学习 C++ 的目的是什么，应当学到什么程度。应当声明：本书的对象不是 C++ 的专业开发人员，而是高校各专业的大学生，他们并无从事程序设计的实际经验，将来也不一定是 C++ 的专业开发人员。众所周知，研制 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++ 的独特机制去进行软件开发。

大学生在学校学习期间学习 C++ 的目的是了解面向对象程序设计方法，能够利用 C++ 编写相对简单的程序，为以后进一步学习和应用打下良好的基础。不可能设想，

通过几十小时的学习,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C++的全貌和作用有基本的认识,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(许多国外的书厚达近千页),而且会冲淡重点,主次不分,使读者感到枯燥无味。

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。读者在掌握基本的概念和方法的基础上,在从事实际程序开发工作过程中遇到具体问题时,查阅一下手册即可。

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。有的不介绍面向过程的程序设计,一开始就直接介绍类和对象;有的先介绍面向过程的程序设计,然后介绍面向对象的程序设计,各有特点。作者认为:不应当把面向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面向过程的知识。作为一个程序设计人员,要掌握面向过程的程序设计,也要掌握面向对象的程序设计。以前,人们通过C语言学习面向过程的程序设计,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++既可以编写面向过程的程序,也可以编写面向对象的程序。在参考、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;第2部分介绍C++面向过程的程序设计;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,以面向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,由面向过程到面向对象,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语言学习面向过程的程序设计,也可以直接通过C++去学习面向过程和面向对象的程序设计。读者可以根据需要选择。

3. 应设计合适的教材体系。C++程序设计牵涉到面向对象程序设计的理论、C++语言的语法以及算法等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,而是在介绍C++语言的特点和应用过程中自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。在介绍程序设计过程中,介绍有关的算法,引导读者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。

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,如果对它缺乏系统的了解,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。本书的做法是全面而系统地介绍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，掌握更多更深入的算法。这样的方法可能符合大多数学习者的情况，降低了学习难度。

4. 应当体现C++标准。C++是从C语言发展而来的，为了与C兼容，C++保留了C的许多用法，人们在编写C++程序时也常常沿用C的某些传统用法。例如，主函数为void类型，无返回值；头文件带后缀.h；使用系统库时不使用命名空间等。但是，ANSI C++标准有新的规定，要求主函数为int类型，如果程序正常执行则返回0值；系统头文件不带后缀.h；使用系统库时使用命名空间std；增加了字符串类型string等。

虽然C++仍然允许使用从C继承来的传统用法，但作者认为：作为教材，应提倡C++的新方法，而不应过分迁就C的习惯。应当引导大学生和所有读者从一开始就按照C++标准编写程序，养成C++的编程风格。本书各章都是依据C++标准介绍的，同时也说明允许使用的C的用法。在本书中，程序的形式如下：

```
#include <iostream>           // 头文件不带后缀.h
#include <string>              // 包含 string 头文件
using namespace std;         // 使用系统库时使用命名空间 std
int main( )                  // 主函数为 int 类型
{
    :
    string str;              // 定义字符串变量 str
    :
    return 0;                // 程序正常执行则返回 0
}
```

5. 应引导学生学习标准C++，而不仅学习依赖于某一特定环境下的“方言化”的C++。本书介绍的是标准C++，读者在学习过程中可以在不同的环境下编译和运行自己的C++程序，但它们只是被当作编译器来使用。不宜喧宾夺主，把过多的精力放在学习某一编译系统的特殊环境和具体细节上，学习的重点应当是C++本身。读者应当能够在不同的平台上(例如Windows环境和非Windows环境)编译和运行C++程序。

6. 本教材力图使学习C++更容易。考虑到许多读者在学习C++之前未学过其他语言，本书把入门起点降低，不要求读者有C语言的基础。在写作过程中作者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将心比心，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，而不应当把简单的问题复杂化，要善于化解难点，深入浅出。不要让读者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，其实换一个角度去解释就很容易理解，甚至用一个通俗的例子就可以把问题说清楚了。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。本书十分便于自学，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。

7. 本教材中所举的程序，是从教学的角度考虑的。这些程序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,力求使读者能方便地读懂程序。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。不要忘记,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。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。在学完本课程后,最好完成一个实际的应用程序,以提高实际应用的能力。

8. 学习本书可以有两个入口: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章~第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。尤其应当看一下前7章中带星号(*)的章节,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》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》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书容易入门,易于学习。

9. 教材要配套,形成纵深。程序设计是一门实践性很强的课程,只靠听课和看书是学不好的。衡量学习好坏的标准不是“懂不懂”,而是“会不会干”。因此必须强调多编程,多上机实践。考虑到不同学校、不同专业、不同读者对学习C++有不同的要求,因此要提供配套用书,不同的读者可以根据要求选用。

本书有两本配套的教学参考书:

(1)《C++程序设计题解与上机指导》。提供主教材各章中全部习题的解答。由于教材的篇幅有限,有些很好的例子无法一一在主教材中列出,我们把它们作为习题,希望读者自己完成,教师也可以从中选择一些习题作为例题讲授。学生除了完成教师指定的习题外,最好把习题解答中的程序全部看一遍,以更好地理解C++程序,扩大眼界,启迪思路,丰富知识,增长能力。在《C++程序设计题解与上机指导》中,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++ 6.0,一种是GCC。GCC是自由软件,不必购买,随本书赠送。GCC可以在Windows环境下使用,也可以在非Windows环境(如DOS, UNIX, Linux)下使用。此外,书中还给出了上机实践任务,要求学生完成若干次上机实践。

随该书赠送的光盘中包括GCC编译系统,主教材中全部例题和习题解答的源程序,以便于教师讲授,也便于学生在此基础上调试和修改程序,以学得更灵活深入。

(2)《C++程序编程实践指导》。这是为学习主教材后进一步深入学习的读者准备的,目的是提高编程能力。有的专业对学生使用C++编程有一定要求,可以在学完课程后,安排一段集中的时间进行一次编程实践,相当于课程设计。该书内容包括:怎样编写C++应用程序;提供若干个不同规模的实际的C++应用程序供分析阅读;安排并指导学生完成1~2个C++应用程序。经过这样的训练,学生的实际能力将会有较大的提高。

此外,向使用本书的教师免费提供讲课的电子演示文稿(素材),以节省教师的备课

时间。

本书由谭浩强编著，谭亦峰和薛淑斌参加了教材写作思路的研讨、收集材料、调试部分程序等工作。在初稿完成后，从事C++教学的周海燕副教授和陈清华副教授仔细阅读了书稿，提出了许多宝贵的意见，使本书更加完善。全国高等院校计算机基础教育研究会和浩强工作室的各位专家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并做了周到的安排，使本书得以在短期内出版。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

2004年春于清华园

目 录

CONTENTS

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	16
第2章 数据类型与表达式	19
2.1 C++的数据类型	19
2.2 常量	21
2.2.1 什么是常量	21
2.2.2 数值常量	21
2.2.3 字符常量	23
2.2.4 符号常量	26
2.3 变量	27
2.3.1 什么是变量	27
2.3.2 变量名规则	28
2.3.3 定义变量	28
2.3.4 为变量赋初值	29
2.3.5 常变量	30
2.4 C++的运算符	31
2.5 算术运算符与算术表达式	32
2.5.1 基本的算术运算符	32
2.5.2 算术表达式和运算符的优先级与结合性	32
2.5.3 表达式中各类数值型数据间的混合运算	32

2.5.4 自增和自减运算符	33
2.5.5 强制类型转换运算符	35
2.6 赋值运算符与赋值表达式	36
2.6.1 赋值运算符	36
2.6.2 赋值过程中的类型转换	36
2.6.3 复合的赋值运算符	38
2.6.4 赋值表达式	39
2.7 逗号运算符与逗号表达式	40
习题	41

第2篇 面向过程的程序设计

第3章 程序设计初步	45
3.1 面向过程的程序设计和算法	45
3.1.1 算法的概念	45
3.1.2 算法的表示	46
3.2 C++ 程序和语句	47
3.3 赋值语句	49
3.4 C++ 的输入与输出	50
*3.4.1 输入流与输出流的基本操作	51
*3.4.2 在输入流与输出流中使用控制符	53
3.4.3 用 getchar 和 putchar 函数进行字符的输入和输出	55
3.4.4 用 scanf 和 printf 函数进行输入和输出	57
3.5 编写顺序结构的程序	58
3.6 关系运算和逻辑运算	58
3.6.1 关系运算和关系表达式	59
3.6.2 逻辑常量和逻辑变量	60
3.6.3 逻辑运算和逻辑表达式	61
3.7 选择结构和 if 语句	63
3.7.1 if 语句的 3 种形式	64
3.7.2 if 语句的嵌套	66
3.8 条件运算符和条件表达式	67
3.9 多分支选择结构和 switch 语句	68
3.10 编写选择结构的程序	70
3.11 循环结构和循环语句	73
3.11.1 用 while 语句构成循环	73
3.11.2 用 do-while 语句构成循环	74
3.11.3 用 for 语句构成循环	75

3.11.4 几种循环的比较	78
3.12 循环的嵌套	79
3.13 break 语句和 continue 语句	79
3.14 编写循环结构的程序	80
习题	85
第4章 函数与预处理	88
4.1 概述	88
4.2 定义函数的一般形式	90
4.2.1 定义无参函数的一般形式	90
4.2.2 定义有参函数的一般形式	91
4.3 函数参数和函数的值	91
4.3.1 形式参数和实际参数	91
4.3.2 函数的返回值	93
4.4 函数的调用	94
4.4.1 函数调用的一般形式	94
4.4.2 函数调用的方式	94
4.4.3 对被调用函数的声明和函数原型	95
*4.5 内置函数	97
*4.6 函数的重载	99
*4.7 函数模板	102
*4.8 有默认参数的函数	103
4.9 函数的嵌套调用	105
4.10 函数的递归调用	109
4.11 局部变量和全局变量	112
4.11.1 局部变量	112
4.11.2 全局变量	113
4.12 变量的存储类别	115
4.12.1 动态存储方式与静态存储方式	115
4.12.2 自动变量	116
4.12.3 用 static 声明静态局部变量	116
4.12.4 用 register 声明寄存器变量	119
4.12.5 用 extern 声明外部变量	119
4.12.6 用 static 声明静态外部变量	121
4.13 变量属性小结	122
4.14 关于变量的声明和定义	124
4.15 内部函数和外部函数	125
4.15.1 内部函数	125
4.15.2 外部函数	125