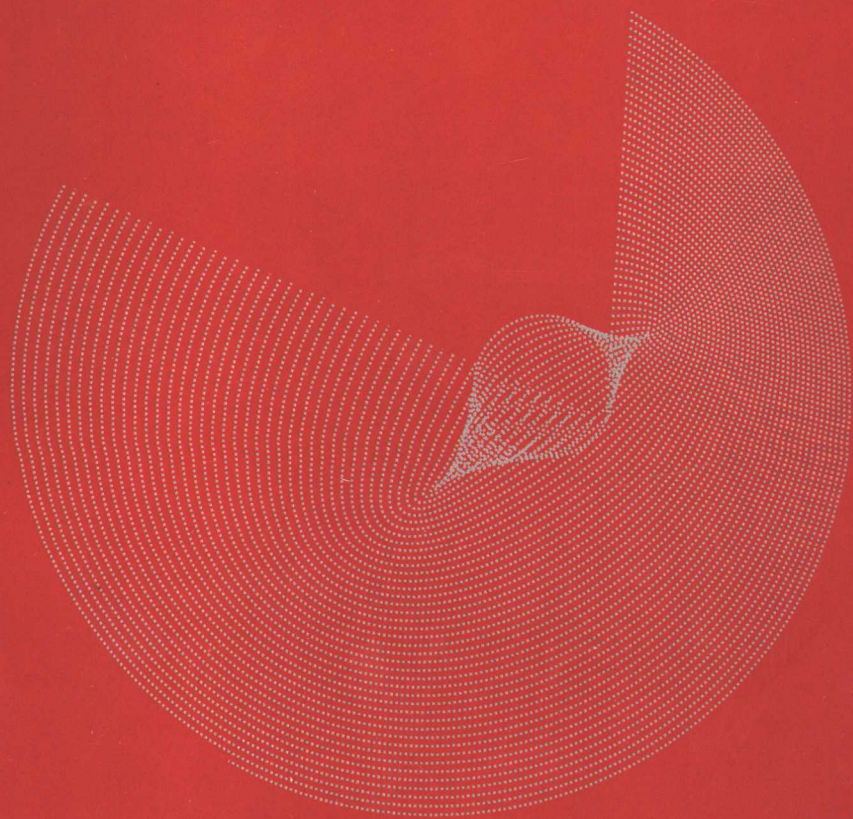


中国高等院校计算机基础教育课程体系规划教材

C++ 面向对象程序设计

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++ 面向对象程序设计

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学习和使用的现代计算机语言,国内许多高校已陆续开设了C++程序设计课程。但是,由于C++牵涉概念很多,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。本书做到准确定位,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。

本书是为已有C语言基础的读者编写的,主要介绍面向对象程序设计的知识和方法。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++面向对象程序设计题解与上机指导》和《C++程序设计实践指导》。

本书是大学各专业学生学习C++的基础教材,也是适于初学者自学的教材,即使没有教师讲授,读者也能看懂本书的大部分内容。使用本书教学将会大大降低教学难度。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C++面向对象程序设计/谭浩强编著. —北京:清华大学出版社,2006.1

(中国高等院校计算机基础教育课程体系规划教材)

ISBN 7-302-12315-2

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(2005)第160115号

出版者:清华大学出版社

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

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:张民

印刷者:北京四季青印刷厂

装订者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:19.75 字数:457千字

版 次:2006年1月第1版 2006年1月第1次印刷

书 号:ISBN 7-302-12315-2/TP·7904

印 数:1~10000

定 价:26.00元

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、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

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。

从2002年起,我花了两年时间,完成了《C++程序设计》一书,由清华大学出版社出版。该书出版后受到广大初学者的欢迎,认为定位准确,取舍合理,概念清晰,逻辑严密,通俗易懂,容易入门。很多教师反映用它作为教材,易教易学,效果很好。很多学生反映,即使没有老师讲授,也能基本看懂该书的内容。该书出版后在一年左右时间内重印了7次,发行6万多册,许多高校选用为教材。

当初在编写《C++程序设计》时,考虑到有些学校并不是把C和C++分别作为独立的两门课,而是希望在C++课程中包括C语言程序设计的基本内容。因此该书的定位是为没有学过C语言的读者写的,不要求学生在学该书之前具有程序设计的基础。在教材内容安排上,先介绍面向过程程序设计,然后再介绍面向对象程序设计。这样的安排,对于未学过面向过程程序设计的同学来说,是比较合适的。

目前在大学生中,也有不少人是学过面向过程的C语言程序设计的,许多学校的师生希望我能在该书的基础上出版一本以C语言为起点的C++教材,这样可以节省教学时间,更好地满足广大师生的要求。本书就是在这种情况下问世的。

本书是在《C++程序设计》一书的基础上改写而成的。考虑到本书的读者已学习过C语言程序设计,因此不必再重新学习面向过程程序设计的部分,而直接从面向对象程序设计入手,将《C++程序设计》一书中介绍面向对象程序设计的第8章到第14章抽出来改写后单独成书,并且重新写了第1章,介绍C++面向对象程序设计的初步知识以及C++对C在面向过程程序设计方面的改进与增强,为以后各章的学习打下基础。

这样的安排使本书内容更集中,篇幅更紧凑。本书仍然保持了通俗易懂、贴近读者、容易入门的特点。希望有助于大学生学习C++课程,初步掌握面向对象程序设计的方法。

作者认为: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。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会真正体会到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，就能使一个没有程序设计基础的初学者变成一个熟练的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时遇到具体问题，查阅一下手册即可。

2. 设计合适的教材体系

C++程序设计牵涉面向对象程序设计的理论、C++语言的语法以及算法3个方面的内容，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。在各章中结合例题介绍有关的算法，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，如果对它缺乏系统了解，将难以真正应用，编出来的程序将会错误百出，通不过编译，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，引导读者由简到繁地学会编写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。

3. C++ 教材应当体现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++标准在一些方面有新的规定,例如要求主函数为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。应当引导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,程序的形式大致如下:

```
#include <iostream>           //头文件不带后缀.h
#include <string>             //包含 string 头文件,以便能在程序中使用字符串变量
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;             //可以定义字符串变量 str
    :
    return 0;              //程序正常执行则返回 0
}
```

4. 引导学生学习标准C++,而不是又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。

5. 本教材力图使学习C++更容易

作者在写作过程中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,而可以选择重点讲授,其余内容由学生自学。

6. 本教材中所举的程序,是从教学的角度考虑的

本书中的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,力求使读者能读懂程序。有些在专业人员看来很“幼稚”的程序,对学习来说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一些,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。

7. 教材要有完整的配套

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++面向对象程序设计题解与上机指导》。提供主教材各章中的全部习题的解答。由于教材的篇幅有限,有些很好的例子无法在教材中列出,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++面向对象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,随该书赠送。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主教材中全部例题的源程序,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++程序设计实践指导》。这是为学习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。

本书由谭浩强编写，谭亦峰和薛淑斌高级工程师参加了写作思路的研讨、收集材料、调试部分程序等工作。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作了周到的安排，使本书得以在短时间内出版。对一切曾经鼓励、支持和帮助过我的同志和朋友，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

2006年1月1日于清华园

目 录

CONTENTS

第1章 C++的初步知识	1
1.1 从C到C++	1
1.2 最简单的C++程序	2
1.3 C++对C的扩充	8
1.3.1 C++的输入输出	8
1.3.2 用const定义常变量	11
1.3.3 函数原型声明	11
1.3.4 函数的重载	12
1.3.5 函数模块	14
1.3.6 有默认参数的函数	16
1.3.7 变量的引用	17
1.3.8 内置函数	23
1.3.9 作用域运算符	25
1.3.10 字符串变量	26
1.3.11 动态分配/撤销内存的运算符new和delete	29
1.3.12 小结	30
1.4 C++程序的编写和实现	31
1.5 关于C++上机实践	33
习题	33
第2章 类和对象	37
2.1 面向对象程序设计方法概述	37
2.1.1 什么是面向对象的程序设计	37
2.1.2 面向对象程序设计的特点	40
2.1.3 类和对象的作用	41
2.1.4 面向对象的软件开发	42
2.2 类的声明和对象的定义	44
2.2.1 类和对象的关系	44
2.2.2 声明类类型	44

2.2.3	定义对象的方法	46
2.2.4	类和结构体类型的异同	48
2.3	类的成员函数	49
2.3.1	成员函数的性质	49
2.3.2	在类外定义成员函数	50
2.3.3	inline 成员函数	51
2.3.4	成员函数的存储方式	52
2.4	对象成员的引用	54
2.4.1	通过对象名和成员运算符访问对象中的成员	54
2.4.2	通过指向对象的指针访问对象中的成员	55
2.4.3	通过对象的引用来访问对象中的成员	55
2.5	类和对象的简单应用举例	55
2.6	类的封装性和信息隐蔽	62
2.6.1	公用接口与私有实现的分离	62
2.6.2	类声明和成员函数定义的分​​离	64
2.6.3	面向对象程序设计中的几个名词	66
习题	67
第3章	关于类和对象的进一步讨论	69
3.1	构造函数	69
3.1.1	对象的初始化	69
3.1.2	构造函数的作用	70
3.1.3	带参数的构造函数	72
3.1.4	用参数初始化表对数据成员初始化	73
3.1.5	构造函数的重载	73
3.1.6	使用默认参数的构造函数	75
3.2	析构函数	78
3.3	调用构造函数和析构函数的顺序	80
3.4	对象数组	81
3.5	对象指针	83
3.5.1	指向对象的指针	83
3.5.2	指向对象成员的指针	84
3.5.3	this 指针	87
3.6	共用数据的保护	88
3.6.1	常对象	89
3.6.2	常对象成员	90
3.6.3	指向对象的常指针	91
3.6.4	指向常对象的指针变量	91
3.6.5	对象的常引用	95

3.6.6	const 型数据的小结	96
3.7	对象的动态建立和释放	96
3.8	对象的赋值和复制	97
3.8.1	对象的赋值	97
3.8.2	对象的复制	99
3.9	静态成员	101
3.9.1	静态数据成员	102
3.9.2	静态成员函数	104
3.10	友元	107
3.10.1	友元函数	107
3.10.2	友元类	111
3.11	类模板	112
	习题	116
第4章	运算符重载	120
4.1	什么是运算符重载	120
4.2	运算符重载的方法	122
4.3	重载运算符的规则	124
4.4	运算符重载函数作为类成员函数和友元函数	127
4.5	重载双目运算符	129
4.6	重载单目运算符	134
4.7	重载流插入运算符和流提取运算符	137
4.7.1	重载流插入运算符“<<”	137
4.7.2	重载流提取运算符“>>”	139
4.8	不同类型数据间的转换	141
4.8.1	标准类型数据间的转换	141
4.8.2	转换构造函数	142
4.8.3	类型转换函数	144
	习题	149
第5章	继承与派生	150
5.1	继承与派生的概念	150
5.2	派生类的声明方式	153
5.3	派生类的构成	154
5.4	派生类成员的访问属性	155
5.4.1	公用继承	156
5.4.2	私有继承	158
5.4.3	保护成员和保护继承	161
5.4.4	多级派生时的访问属性	164

5.5	派生类的构造函数和析构函数	166
5.5.1	简单的派生类的构造函数	166
5.5.2	有子对象的派生类的构造函数	169
5.5.3	多层派生时的构造函数	172
5.5.4	派生类构造函数的特殊形式	175
5.5.5	派生类的析构函数	175
5.6	多重继承	176
5.6.1	声明多重继承的方法	176
5.6.2	多重继承派生类的构造函数	176
5.6.3	多重继承引起的二义性问题	179
5.6.4	虚基类	182
5.7	基类与派生类的转换	187
5.8	继承与组合	191
5.9	继承在软件开发中的重要意义	192
	习题	193
第6章	多态性与虚函数	198
6.1	多态性的概念	198
6.2	一个典型的例子	199
6.3	虚函数	204
6.3.1	虚函数的作用	204
6.3.2	静态关联与动态关联	208
6.3.3	在什么情况下应当声明虚函数	209
6.3.4	虚析构函数	210
6.4	纯虚函数与抽象类	211
6.4.1	纯虚函数	211
6.4.2	抽象类	212
6.4.3	应用实例	213
	习题	219
第7章	输入输出流	220
7.1	C++的输入和输出	220
7.1.1	输入输出的含义	220
7.1.2	C++的I/O对C的发展——类型安全和 可扩展性	220
7.1.3	C++的输入输出流	221
7.2	标准输出流	225
7.2.1	cout, cerr 和 clog 流	225
7.2.2	格式输出	228

7.2.3 用流成员函数 put 输出字符	232
7.3 标准输入流	233
7.3.1 cin 流	233
7.3.2 用于字符输入的流成员函数	235
7.3.3 istream 类的其他成员函数	239
7.4 文件操作与文件流	242
7.4.1 文件的概念	242
7.4.2 文件流类与文件流对象	243
7.4.3 文件的打开与关闭	244
7.4.4 对 ASCII 文件的操作	246
7.4.5 对二进制文件的操作	251
7.5 字符串流	256
习题	261
第 8 章 C++ 工具	262
8.1 异常处理	262
8.1.1 异常处理的任务	262
8.1.2 异常处理的方法	263
8.1.3 在函数声明中进行异常情况指定	270
8.1.4 在异常处理中处理析构函数	270
8.2 命名空间	272
8.2.1 为什么需要命名空间	273
8.2.2 什么是命名空间	276
8.2.3 使用命名空间解决名字冲突	278
8.2.4 使用命名空间成员的方法	280
8.2.5 无名的命名空间	282
8.2.6 标准命名空间 std	282
8.3 使用早期的函数库	283
习题	284
附录 A 常用字符与 ASCII 代码对照表	286
附录 B 运算符与结合性	287
参考文献	289

第1章

C++的初步知识

1.1 从C到C++

C语言是结构化和模块化的语言,它是面向过程的。在处理较小规模的程序时,程序员用C语言较为得心应手。但是当问题比较复杂、程序的规模比较大时,结构化程序设计方法就显出它的不足。C程序的设计者必须细致地设计程序中的每一个细节,准确地考虑程序运行时每一时刻发生的事情,例如各个变量的值是如何变化的,什么时候应该进行哪些输入,在屏幕上应该输出什么等。这对程序员的要求是比较高的,如果面对的是一个复杂问题,程序员往往感到力不从心。当初提出结构化程序设计方法的目的是解决软件设计危机,但是这个目标并未完全实现。

为了解决软件设计危机,在20世纪80年代提出了面向对象的程序设计(object oriented programming, OOP)思想,这就需要设计出能支持面向对象的程序设计方法的新语言。Smalltalk就是当时问世的一种面向对象的语言。而在实践中,人们发现由于C语言是如此深入人心,使用如此广泛,以至最好的办法不是另外发明一种新的语言去代替它,而是在它原有的基础上加以发展。在这种形势下,C++应运而生。C++是由AT&T Bell(贝尔)实验室的Bjarne Stroustrup博士及其同事于20世纪80年代初在C语言的基础上开发成功的。C++保留了C语言原有的所有优点,增加了面向对象的机制。由于C++对C的改进主要体现在增加了适用于面向对象程序设计的“类(class)”,因此最初它被Bjarne Stroustrup称为“带类的C”。后来为了强调它是C的增强版,用了C语言中的自加运算符“++”,改称为C++。

AT&T发布的第一个C++编译系统实际上是一个预编译器(前端编译器),它把C++代码转换成C代码,然后用C编译系统编译,生成目标代码。第一个真正的C++编译系统是1988年诞生的。C++ 2.0版本于1989年出现,它作了重大的改进,包括了类的多继承。1991年的C++ 3.0版本增加了模板,C++ 4.0版本则增加了异常处理、命名空间、运行时类型识别(RTTI)等功能。ANSI C++标准草案是以C++ 4.0版本为基础制定的,1997年ANSI C++标准正式通过并发布。但是目前使用的C++编译系统中,有一些是早期推出的,并未全部实现ANSI C++标准所建议的功能。

C++是由C发展而来的,与C兼容。用C语言写的程序基本上可以不加修改地用