面向21世纪课程教材
Textbook Series for 21st Century

高级语言C++程序设计

刘璟 编著

高等教育出版社
HIGHER EDUCATION PRESS

面向21世纪课程教材
Textbook Series for 21st Century

高级语言 C++程序设计

刘璟 编著

高等教育出版社
HIGHER EDUCATION PRESS

图书在版编目(CIP)数据

高级语言 C++ 程序设计/刘璟编著. —北京:高等教育出版社,2001.1

面向 21 世纪课程教材

ISBN 7-04-008905-X

Ⅰ. 高… Ⅱ. 刘… 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(2000)第 77522 号

高级语言 C++ 程序设计

刘 璟 编著

出版发行	高等教育出版社		
社 址	北京市东城区沙滩后街 55 号	**邮政编码**	100009
电 话	010-64054588	**传 真**	010-64014048
网 址	http://www.hep.edu.cn http://www.hep.com.cn		
经 销	新华书店北京发行所		
印 刷	中国科学院印刷厂		
开 本	787×960 1/16	**版 次**	2001 年 1 月第 1 版
印 张	20.25	**印 次**	2001 年 1 月第 1 次印刷
字 数	350 000	**定 价**	17.30 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

内 容 提 要

本书是教育部“高等教育面向21世纪教学内容和课程体系改革计划”的研究成果，是面向21世纪课程教材。

本书内容包括：第一章 绪论；第二章 C++语言初步；第三章 基本数据类型与基本运算；第四章 基本控制结构与导出数据类型；第五章 函数，函数与运算符的重载；第六章 指针，引用与动态内存分配；第七章 类与对象；第八章 继承与派生；第九章 模板；第十章 输入输出流和第十一章 用C++语言设计面向对象程序。

本书结构清晰、系统性强、语言的叙述简洁；适合作为计算机专业与非计算机专业学生学习高级语言程序设计及面向对象技术的教材。

作者简介

刘　璟　南开大学教授，博士生导师，计算机科学与技术系主任，教育部全国高等学校计算机科学与技术教学指导委员会软件专家组成员，天津市高等学校计算机基础教学指导委员会副主任。主要研究领域为算法设计与分析、面向对象程序设计技术、并行与分布式处理及应用、网络应用软件技术等。

前　言

本教材是为高等学校计算机及相关专业的高级语言程序设计课程编写的。

用C++语言取代Pascal语言(或者C语言),使教学内容上发生了巨大的变化;把程序设计技术的重点从结构程序设计转向面向对象程序设计,是一个重大变革。作者曾经与一些计算机专家讨论,大家比较一致的看法是:

·面向对象程序设计(Object Oriented Programming,OOP)技术是目前程序设计与软件开发的主流,是培养计算机专业人才的重点内容,过去只在高年级开一个选修课的方法不能继续下去了。现在国家把发展软件产业作为科教兴国的重大战略任务提出来,培养高级编程人才首当其冲。程序设计课程的变革可以说是当务之急。

·讲授OOP技术最合适的语言是C++语言。一方面它是应用最广的OOP语言,另一方面,也有利于学生学习一般的编程技巧。不过,一定不要把它作为C语言的扩充来讲授。

然而,在如何进行以OOP技术为主的程序设计教学问题上,专家们的意见却不尽统一。问题主要在于,C++语言的规模和C++程序的复杂程度,往往使得教师们在面对刚刚步入大学的学生时感到困难重重,比起十几年前讲BASIC语言不知要难多少倍!

不过,从目前国内外的计算机教学的发展来看,从一开始就讲授C++语言是大势所趋,因为这是现代软件技术的需要。早在20世纪90年代中期,南开大学、中山大学等高校已经开始采用C++语言作为程序设计主干课的内容,并取得了一些经验。尽管如此,教材的编写仍然是一个艰巨而具有探索性的工作。在本书的编写过程中,作者始终把讲授编程技术作为主线,努力使其不要淹没在大量语法规则的讲解之中;虽然一些C++的实用程序都很长,但仍然尽量多引用高水平的程序实例,避免像许多C++教材中那样把实例简化到仅用来说明语法,成了一本语法说明书。我们的教材就是为了培养高水平编程人员的。另一方面也应指出,掌握现代编程技术是一个长期的过程,同

学们应在后继课程(如算法与数据结构、编译技术、数据库技术、软件工程、JAVA 语言等)中不断地提高和完善。

在编写本书的过程中,得到教育部高等学校计算机科学技术教学指导委员会和高等教育出版社的专家和编辑的鼓励,天津大学边奠英教授详细审阅了全文,并提出了许多宝贵意见,南开大学周玉龙、于春凡副教授在原稿中发现了许多错误和不妥之处,邵秀丽副教授和研究生熊伟、罗艳以及董沙沙则为本书选编了习题,编写了三个附录,检查了书中的应用程序,并完成了全部的整理工作,在此向他们表示衷心的感谢。

在编写本书的过程中,作者参阅了国内外许多程序设计语言与数据结构方面的书籍,从中吸收了新的思想、新的内容,同时又力图有所突破、有所创新,然而能力和水平有限,可能有许多错误和不足之处,敬请阅读本书的老师和同学予以指正。

刘 璟

2000 年于南开大学

目　录

第一章 绪 论

1.1 程序设计与程序设计语言

半个世纪以来，计算机技术无论是作为科学学科还是作为现代产业，已从一颗幼苗成长为枝繁叶茂的参天大树。回顾其发展历程，计算机科学与工程的发展（包括“网络时代”的今天）一直是围绕着程序设计这个中心课题进行的。

1.1.1 计算机与程序设计

计算机也许是20世纪人类带给21世纪的最有价值的礼物，是人类文明历史上最伟大的发明之一，现在估计它对人类生活将会产生多么大的影响也许还为时尚早。不过我们总是能够把计算机与人类其他伟大发明，如飞机、电灯、汽车、电视机等等相比较，从其差别中可以感到它的威力、影响和壮美前景。计算机与其他发明的主要差别是：

·人类的发明都是对自己的器官的延长或替代，别的发明都可归结为人的四肢和五官的延长或替代；而计算机则是人类大脑功能的延长或某种替代，所以被称为“电脑”。

·人类的发明可以应用在各个不同的局部领域，计算机与众不同之处在于它可以应用在几乎所有的人类活动的领域。

目前，计算机的这两个特征还在发展之中，计算机可以在怎样的程度上延长或代替大脑的活动，计算机可以在何种程度上被广泛而深入地应用于各个领域，谁也不能指出一个“到顶”不再发展的时间，不过我们现在可以指出的是，使计算机具有如此影响力的根本原因是，计算机不是一个一次性的直接服务产品，它为人类服务是有条件的，这个条件就是**程序设计**。

没有程序和程序设计，计算机可以说是一堆废物，也可以换一个方法说：程序（软件）是计算机的必要组成部分。计算机首先要求人们不断地在程序设计上付出大量的创造性劳动，然后才能享受到它的服务。

计算机本身是人类智慧的产物，它的诞生又导致了人们投入十倍，百倍的精力和智慧用于程序设计和软件开发，从而又引发出无穷无尽的新的发明创造。

有关计算机科学与技术的大部分研究工作都是围绕程序设计进行的，特别是形势发展到今天，新型计算机本身（主要指计算机硬件核心的芯片）的设计也

归结为使用高级硬件描述语言的“程序设计”，所以也可以说整个计算机产业（硬件和软件的研究、设计和生产）就是在进行程序的设计与开发。因此计算机专家和专业人员的培养和训练的最主要任务就是让学生掌握程序设计及其相关理论的研究和开发能力。

其实，把计算机比作电视机或电冰箱并不恰当，它更像是人所驯养的一匹马，能按照人的指令去完成各种任务，所谓**程序**，就是要计算机完成某一任务所规定的一系列动作步骤。计算机好像是唯人的命令是从的仆人，严格地按照程序规定的步骤完成任务。当然，计算机程序不是简单的几条或几十条命令放在一起，现代的计算机每秒钟可以执行成千上万条指令（最快的计算机每秒钟可处理超过 10 000 亿条指令）。因此，计算机程序规模很大，内容十分复杂。为计算机编程序是一种非常复杂，具有挑战性的工作。也可以说，自计算机问世的半个世纪以来，人们都是在研究设计各种各样的程序，使计算机完成各种各样的任务。

1946 年美国的 Mauchly 和 Eckert 研制的第一台电脑 ENIAC 上应用的程序是用来计算火炮的弹道函数的。微软公司开发的 Windows 系列是一种用来管理计算机资源的图形界面操作系统，它也是一个大规模的程序。这个程序已为开发商创造了数百亿美元的财富。1997 年 5 月份，另一个计算机程序的开发也引起了新闻界的注意，这就是击败了国际象棋世界冠军卡斯帕罗夫的计算机程序“Deeper Blue”（深蓝）。这是一个能够下棋的程序，它以 4:2 击败了当代最强的国际象棋大师，赢得了 70 万美元的奖金。（实际上，IBM 公司为开发“Deeper Blue”的投入比这笔奖金要多得多。）

因此，程序设计是一件工作量永无止境，极其困难复杂而又富有魅力和创造乐趣的工作。这样一项职业，每年都吸引着数以十万计的优秀人才投入其中，容纳下他们的全部智慧、想象力和创造性的工作，换来了计算机产业和计算学科的日新月异的发展。

1.1.2　程序设计语言

程序设计的任务就是用计算机懂得的语言（即程序设计语言）编写程序，然后交给它去执行。

1. 计算机指令系统

严格地说，一台“纯粹”的计算机（或称“裸机”）并没有特别高超的本领，它只能做如下工作：

· 完成几十种（或一百多种）不同的简单“动作”，例如把内存的某地址的数取到某寄存器；把某地址的两个数相加送到某寄存器；判断某个值是否为 0 等等。

· 计算机设计者把计算机可以完成的动作编辑成一个指令表，每种动作赋

予一个二进制代码，并为机器的每种动作设计一种通用的格式：由指令码和内存地址组成的指令。一条指令就是一个固定长度的由指令码和地址码组成的二进制位串，这就是计算机唯一可以读懂的语言。一般称作**机器语言**。

·程序员把要计算机完成的任务分解为一系列其指令表（或指令系统）包括的“动作”，以指令序列的形式写出来，这就是机器语言程序设计。

1949 年，普林斯顿大学的冯·诺依曼（von Neumann）研制的 EDVAC 计算机，首次把上述指令序列形式的程序与数据一同置于磁芯存储器中。从那时开始，计算机通过识别置于存储器中的由 0 和 1 组成的二进制指令序列，来依次执行指定的操作，这种工作方式一直延续到今天。

在计算机应用的最初的十几年中，大多数计算机程序就是用所谓“机器语言”编写的。这种“语言”虽然十分简单，机器可以“看”懂，但对于程序员来说却很不方便。完成一个简单的计算公式也要编写几十条指令，编程工作枯燥而繁琐，程序冗长而难读，调试、修改、移植和维护都是难题。因此，早期的计算机应用不广，编程的专业性极强，人们逐渐感到用机器语言编程是计算机应用向前发展的瓶颈。

2. **低级编程语言**

20 世纪 50 年代的计算机专家们，一方面以“机器语言”的形式写出了大量的应用程序，例如在那个年代实现了天气预报和卫星上天，计算机曾做出了巨大的贡献。另一方面，他们也在研究，可否在语言问题上兼顾对话的双方：计算机与人。“机器语言”——太“低级”了，机器容易懂，但对人却太不方便。于是，一种“汇编系统”程序问世了，这种程序的功能是把一种“汇编语言”翻译为“机器语言”。

汇编语言是用人比较习惯的符号来代替指令编码，例如用“ADD”来代替“001”表示加法操作，用“Move”来代替“010”表示数据移动。用符号代替二进制地址表示参加操作的数据，这样大大减少了编程工作的困难。后来又改进为“宏汇编语言”。一条宏汇编指令可以代替多条机器指令。人们用汇编语言或宏汇编语言写程序，通过汇编系统（Assembler）把它们翻译成计算机唯一“看”得懂的机器语言程序，然后再令其执行。

使用汇编语言编程比使用机器语言编程要容易，另外由于汇编语言指令与机器语言指令基本上一条对一条或一条对几条，所以汇编系统的程序开发也不太复杂。因此，汇编语言编程很快取代了机器语言编程。到了 60 年代，机器语言编程已经比较少了。汇编逐渐取代机器语言，成为主要编程语言。汇编语言编程延续了相当长的时间，后来以 FORTRAN 和 ALGOL_60 为代表的高级语言逐渐流行，汇编语言的应用从应用程序的开发转向了系统程序，如操作系统（Operating System，OS），编译系统（Compiler）的开发。到了 70 年代，新一代的高级语

言 Pascal 和 C 语言问世，又逐渐在系统程序的开发领域取代了汇编语言。然而到了 80 年代，计算机一些新的应用形式如单板机、单片机等的简单编程仍然以汇编语言为主。到了 90 年代，汇编语言的应用才相对减少了。

汇编语言和机器语言都属于**低级语言**，这是因为其语言的结构都是以面向机器的指令序列形式为主，与人的习惯语言方式距离较远，所以它们的共同缺点是：

- 依赖于机器，可移植性差；
- 代码冗长，不易于编写大规模程序；
- 可读性差，可维护性差。

总之，即使是汇编语言，也没有解决计算机编程难的基本问题。低级语言必然为高级语言取代，没有高级语言就没有今天的计算机应用的网络化、多媒体化、智能化等等。

3. **高级程序设计语言**

有了计算机以后，许多科学与工程计算问题交给计算机完成，大多获得了成功。计算机比人们用手工计算速度快，精度高，为人们帮了大忙，那么编程序是否也可以请计算机帮忙呢？

从 20 世纪 50 年代开始，作为当时的尖端科学技术，许多计算机科学家开始研究一种“可以编程序的程序”，有些研究者把它称为“程序设计自动化”。这项工作包括两个部分：

(1) 设计一种人们习惯的高级语言规范。由于在当时计算机完成的主要是一些工程计算任务，因此这种用来描述计算任务的高级语言的主要部分就是数学表达式。

(2) 设计一种程序，它能够把用高级语言写出的对计算任务的描述翻译为机器语言的指令序列。这种程序被称为**编译程序**、**编译系统**或**编译器**。

最早取得成功的是 IBM 公司的 John Backus，他领导的研究小组于 1954 年提出了一个高级语言规范“IBM mathematical FORmula TRANslation system”(FORTRAN)，两年后完成了编译系统的研制。而正式公布的 FORTRAN I 语言和编译系统则是在 1957 年 4 月。

FORTRAN 语言的诞生标志着计算机技术的一个新的里程碑。有了高级程序设计语言及其编译系统的帮助，人们可编制出规模越来越大，结构越来越复杂的程序。计算机的应用领域不断扩展，计算机技术能够发展到今天的局面，没有高级语言及其编译技术的突破是不可能实现的。

1.1.3 程序设计方法理论的发展

程序设计技术的发展，是一个逐步提高的过程，是一个与实际应用需要互相