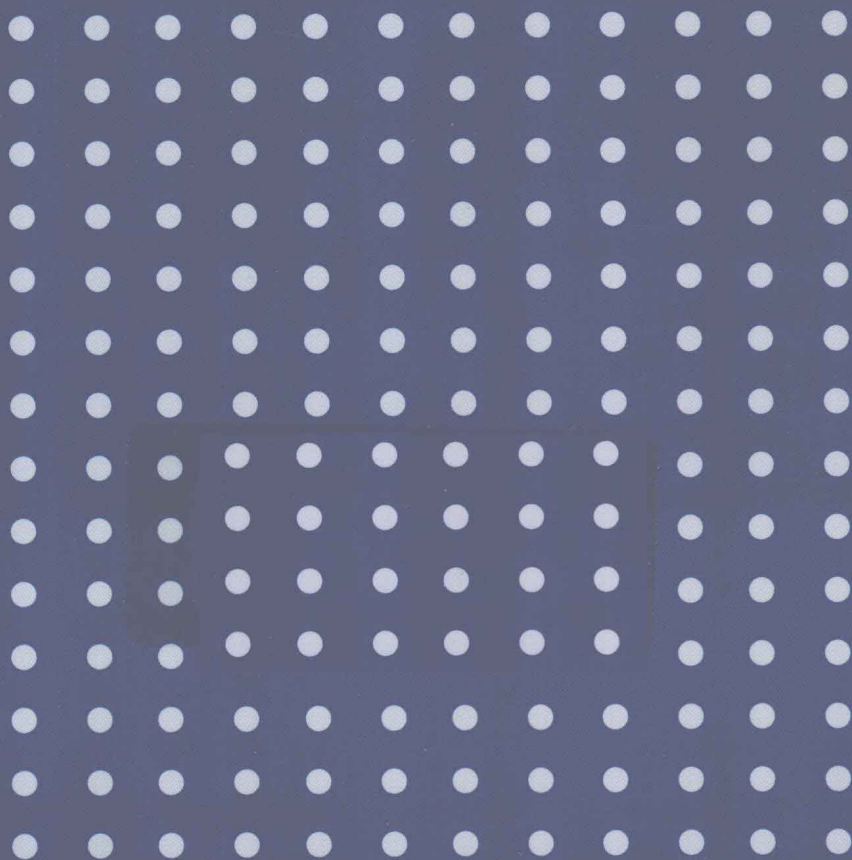


重点大学计算机专业系列教材

新标准C++程序设计教程

郭炜 编著



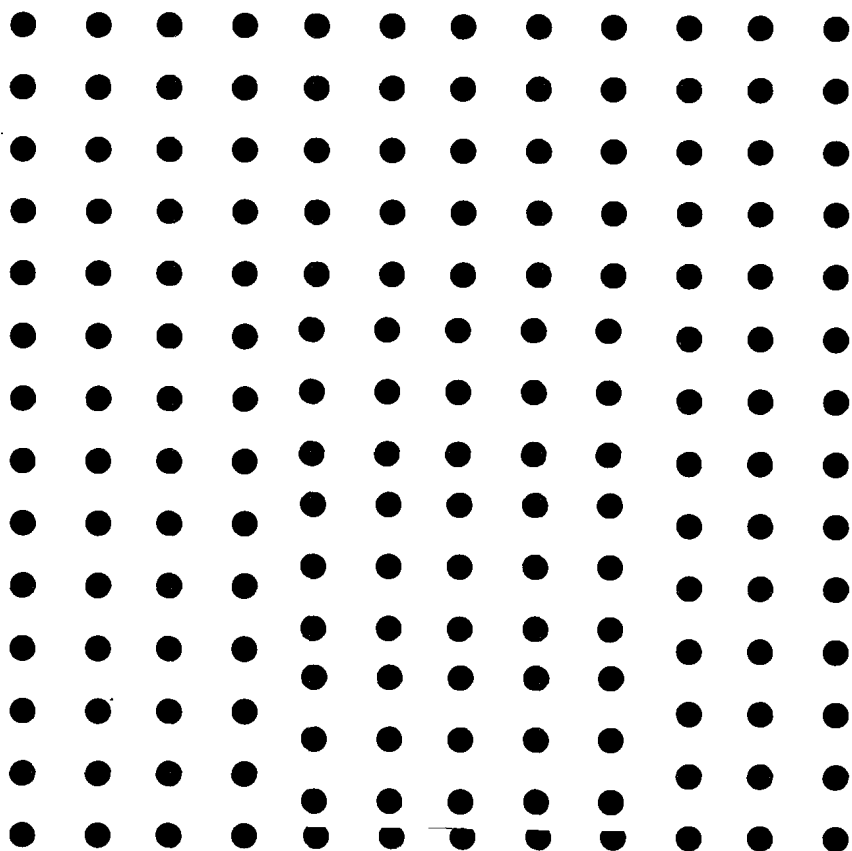
清华大学出版社



重点大学计算机专业系列教材

新标准C++程序设计教程

郭炜 编著



清华大学出版社

北京

内 容 简 介

本书基于新的 C++ 标准“C++ 11”,从全新的思路出发,融合作者丰富的编程实践经验,深入浅出地全面介绍 C++ 程序设计的过程,包含丰富的样例程序,强调实践性和专业性。

本书适合作为高等院校理工类专业程序设计课程的教材、学生自学和做毕业设计的参考书,也可供 IT 从业人员和编程爱好者参考。

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

新标准 C++ 程序设计教程/郭炜编著. —北京:清华大学出版社,2012.8

重点大学计算机专业系列教材

ISBN 978-7-302-28380-5

I. ①新… II. ①郭… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 050118 号

责任编辑:付弘宇 王冰飞

封面设计:常雪影

责任校对:李建庄

责任印制:何 芊

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:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28 字 数:696 千字

版 次:2012 年 8 月第 1 版 印 次:2012 年 8 月第 1 次印刷

印 数:1~3000

定 价:44.50 元

产品编号:045620-01

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

一、本书的写作背景

C++功能强大、运用广泛,许多大学都将其作为入门的程序设计语言进行教学。笔者在北京大学信息科学技术学院讲授 C++ 程序设计已有 10 年,随着时间的推移,渐觉现有的教材已经不能满足教学的需要,于是萌生了自己编写一本教材的想法。

C++有两大特点:支持面向对象的程序设计和支持泛型程序设计。然而,国内大部分教材往往对“泛型程序设计”这部分内容基本忽略,或只是略作交待。这导致许多学过或准备学 C++ 的学生有如下印象:C++是为了编写大型的程序而设计的,如果编写一个十几、几十行的小程序,没有必要用 C++,用 C 语言就足够了。实际上,编写很小的程序用面向对象的程序设计方法确无必要,但不等于用 C++ 没必要。C++ 中的标准模板库(STL)是泛型程序设计的最成功应用,其中包含许多常用的数据结构(如动态数组、栈等)和算法(如排序、二分查找等),STL 即便应用于十几行的程序中,也能有效地提高编程效率。对熟练的 C++ 程序员来说,编写一个十几行的程序多半不会考虑到面向对象的程序设计,但会很自然地用到 STL。在笔者看来,如果计算机专业的学生学了 C++ 却不会用 STL,那么找工作面试的时候是会受影响的。C++ 标准委员会成员 Andrew Koenig 有句名言“库设计就是语言设计,语言设计就是库设计。”学了 C++ 语言,却不会用该语言的核心库,对于计算机专业的学生来说,这样的教学很难说是成功的。

国内大多数 C++ 教材对泛型程序设计和 STL 讲述甚少,国外虽有几部经典教材,全面覆盖了 C++ 的两大特点,但是都卷帙浩繁,动辄近千页,不适合初学者。为解决这个矛盾,笔者编写了本教材,篇幅适当,全面讲述了 C++ 面向对象的各种特性,此外还覆盖了标准模板库 90% 以上的内容。初学者通过本书的学习,可以比较全面地掌握 C++ 程序设计语言的精髓。

另外,大多数 C++ 教材依据的是 1998 年的 C++ 标准(一般称为“C++ 98”)。而在 1998 年后,C++ 标准进行了一些修订,加入了一些新特性,2011 年 C++ 标准委员会又通过了新的 C++ 标准“C++ 11”。这些变化在大多数教材中没有

体现。让教材与时俱进,也是笔者编写本教材的初衷之一。

二、本书的特点

1. 内容深广却通俗易懂,入门与提高并重

本书面向大学计算机专业的低年级学生,或非计算机专业但对编程能力要求较高的学生。本书可作为入门的程序设计语言教学之用,没学过 C 语言的读者可以直接学习本书。

本书内容很广,覆盖了 C++ 语言的方方面面,全面讲述了标准模板库 STL 的用法,几乎可以作为 C++ 语言的参考手册来查阅。第 4 篇“C++ 高级主题”中的内容更是大多数同等篇幅的教材所不曾涉及的。而且,由于本书的宗旨是让读者不但要知其然,还要知其所以然,因此对于 C++ 的一些语法特性,不但介绍如何使用,还会讲解 C++ 为什么会有这个语法特性,甚至该特性是如何实现的,如“多态”的实现方法。

笔者有 10 年的 C++ 语言第一线教学经验,非常清楚学生在学习 C++ 时哪些地方不易掌握,会提出什么样的问题,以及他们的问题应该如何回答。因此,在本书写作时,就已经将学生困惑的解答融入其中,用精简的语言直指问题的重点、难点和本质,可以说将有限的文字都用在了刀刃上。笔者的目标是力图做到“读者不用教师讲授,也能独立看懂本书”。

2. 紧扣 C++ 标准

国内大多数 C++ 教材依据的是 1998 年的 C++ 标准“C++ 98”。许多教材甚至都不能完全符合“C++ 98”的标准,这从其声称“程序都在 Visual C++ 6.0 中编译通过”就可看出——Visual C++ 6.0 并不是严格遵循“C++ 98”标准的编译器。目前,对 C++ 标准支持最好的编译器有 GNU gcc 和微软的 Visual C++ 10.0(包含在 Visual Studio 2010 中)等。本书中的所有程序除个别有特殊说明的以外,都同时在 Visual C++ 10.0 和 Dev C++ 4.9.9.2(其内核编译器是 gcc)中编译通过,并且运行结果相同,可以保证是符合 C++ 标准的,而不是某种“C++ 方言”。

最新的“C++ 11”标准通过的时间很短,目前还没有编译器能完全支持它。因此,本书不可能完全以“C++ 11”作为依据。本书的主要依据依然是“C++ 98”,但是收录了几个“C++ 98”之后新引入的特性,如“long long”数据类型、无序容器(即哈希表)、智能指针 shared_ptr 等。本书中的头文件都是 C++ 风格的,不像某些教材仍然使用 C 语言风格的头文件。另外,许多教材中的程序在用到字符串时往往使用 C 语言风格的字符数组,而本书则尽量使用 C++ 风格的 string 对象处理字符串。

3. 程序实例丰富实用,贴近编程实践

笔者不仅具有丰富的 C++ 教学经验,还有着数十万行的 C++ 商业软件开发经验。笔者独立开发了多种流行的英语学习软件,如《我爱背单词》、《我爱背句子》、《我爱学语法》、《我爱学音标》、《角斗士超级复读机》等,还开发了背单词的网站“爱单词网”。这些软件大量使用 C++ 语言进行开发,有的完全用 C++ 写成。因此,笔者自认为在对 C++ 语言的运用和理解方面,比一般的教材作者多了一些心得,更能从实践的角度阐述如何运用 C++ 的各种特性。本书中所有样例程序都由笔者精心编写,绝非网上复制所得。程序风格优美,贴近现实,对实践的指导意义很强。

4. 强调程序设计基本思想的培养

笔者担任北京大学 ACM 国际大学生程序设计竞赛队教练已有 8 年,手下的队员都是

北京大学最出色的编程高手。他们的成长、求学、求职经历可以验证,算法才是程序设计的核心。学习程序设计决不是仅仅掌握一门语言的语法,更重要的是掌握算法。只掌握语言的语法,在碰到具体问题时往往还是不知道如何去编程解决。作为入门的程序设计语言教材,本书不可能讲述太多的算法,但是程序设计的基本思想是必定要涉及的。为此,本书专门辟出一章,讲述了枚举、递归、二分 3 种基本的程序设计思想。此外,其他章节的一些例题和程序也能体现程序设计的基本思路。

三、内容编排

本书主要分为以下四篇。

第 1 篇 结构化程序设计:是对 C 语言的继承以及在 C 语言基础上的一些扩充。学过 C 语言的读者可以粗略阅读甚至跳过此篇大部分内容,但是需要学习目录中带“*”的章节,这些章节仍然是 C 语言中没有的内容。

第 2 篇 面向对象的程序设计:包含类和对象、运算符重载、继承、多态等内容,是 C++ 语言的学习重点。

第 3 篇 泛型程序设计:包括如何编写模板,以及如何使用标准模板库 STL。即便不想用面向对象的程序设计方法编程,学习 STL 也是大有裨益的。

第 4 篇 C++ 高级主题:涉及了异常处理、名字空间、C++ 风格的强制类型转换等 C++ 语言中比较深入的内容,以及 C++ 11 标准引入的几个新特性。

本书各章最后都有小结和习题。各章内的讲述中还穿插了许多思考题,思考题难度较大,很适合作为启发式教学的讨论话题。

四、总结

总而言之,本书书名中的“新标准”有两层含义:第一层含义是指本书所有的讲述和程序都是严格遵循 C++ 标准的,而且提及了一些新 C++ 标准的内容;第二层含义更为重要,指的是希望读者通过本书的学习,对 C++ 的掌握程度能够达到更高的标准,除了深入理解面向对象的程序设计外,还能够进行泛型程序设计并熟练使用 STL。

本书的配套课件和书中的例题程序代码可以从清华大学出版社网站(www.tup.com.cn)下载。如果在本书和课件的下载使用中遇到问题,请联系 fuhuy@tup.tsinghua.edu.cn。

五、鸣谢

编写本书的动力来自作者在北京大学信息学院“程序设计实习”课程的教学经历。感谢课程主持人李文新教授对作者在教学中的支持和指导。还要感谢多年来共同讲授此课程的余华山老师、田永鸿老师,和他们的讨论使我获益良多。三位老师编写的讲义,也是本书的重要参考。

感谢您选用本书。由于笔者水平所限,不足之处在所难免,欢迎读者及同仁们批评指正,笔者不胜感激。笔者的 E-mail: gwpl@pku.edu.cn。

郭 炜

2012 年 6 月于北京大学

第 1 篇 结构化程序设计

| | |
|--------------------------------|----|
| 第 1 章 计算机基础知识 | 3 |
| 1.1 信息在计算机中的表示和存储 | 3 |
| 1.1.1 如何用 0 和 1 表示各种信息 | 3 |
| 1.1.2 二进制和十六进制 | 4 |
| 1.1.3 整数和小数的计算机表示 | 6 |
| 1.2 计算机程序设计语言 | 7 |
| 1.2.1 机器语言 | 7 |
| 1.2.2 汇编语言 | 8 |
| 1.2.3 高级语言 | 9 |
| *1.3 C++ 语言的历史 | 10 |
| 1.4 小结 | 11 |
| 习题 | 12 |
| 第 2 章 C++ 语言的基本要素 | 13 |
| 2.1 C++ 的标识符 | 13 |
| 2.2 C++ 的关键字 | 13 |
| *2.3 最简单的 C++ 程序 | 14 |
| 2.4 变量 | 16 |
| 2.4.1 变量的定义 | 16 |
| 2.4.2 变量的初始化 | 16 |
| 2.4.3 变量的赋值 | 17 |
| 2.4.4 常变量 | 17 |
| 2.5 C++ 的数据类型 | 17 |
| 2.5.1 C++ 基本数据类型 | 17 |
| 2.5.2 数据类型自动转换 | 18 |

| | | |
|--------------|-----------------------------|-----------|
| 2.5.3 | 用 cin 读入类型不同的变量 | 20 |
| 2.6 | 常量 | 20 |
| 2.6.1 | 整型常量 | 21 |
| 2.6.2 | 实数型常量 | 21 |
| 2.6.3 | 布尔型常量 | 22 |
| 2.6.4 | 字符型常量 | 22 |
| 2.6.5 | 字符串常量 | 22 |
| 2.6.6 | 符号常量 | 23 |
| 2.7 | 运算符和表达式 | 23 |
| 2.7.1 | 算术运算符 | 24 |
| 2.7.2 | 赋值运算符 | 26 |
| 2.7.3 | 关系运算符 | 27 |
| 2.7.4 | 逻辑运算符和逻辑表达式 | 27 |
| 2.7.5 | 位运算符 | 28 |
| 2.7.6 | 条件运算符 | 32 |
| 2.7.7 | sizeof 运算符 | 33 |
| 2.7.8 | 强制类型转换运算符 | 33 |
| 2.7.9 | 逗号运算符 | 34 |
| 2.7.10 | 运算符的优先级和结合性 | 34 |
| 2.8 | 注释 | 35 |
| 2.9 | 小结 | 36 |
| | 习题 | 36 |
| 第 3 章 | C++ 语言的控制结构 | 39 |
| 3.1 | 用 if 语句实现选择结构 | 39 |
| 3.2 | 用 switch 语句实现选择结构 | 43 |
| 3.3 | 用 for 语句实现循环结构 | 46 |
| 3.4 | 用 while 语句实现循环结构 | 49 |
| 3.5 | 用 do...while 语句实现循环结构 | 50 |
| 3.6 | 用 break 语句跳出循环 | 50 |
| 3.7 | continue 语句 | 52 |
| 3.8 | goto 语句 | 52 |
| 3.9 | 使用 freopen 方便程序调试 | 53 |
| 3.10 | 小结 | 54 |
| | 习题 | 54 |
| 第 4 章 | 函数 | 57 |
| 4.1 | 函数的定义和调用 | 57 |
| 4.1.1 | 函数的定义 | 57 |

| | | |
|--------------|------------------------|-----------|
| 4.1.2 | 函数调用和 return 语句 | 58 |
| 4.1.3 | 函数使用实例 | 58 |
| 4.2 | 函数的声明 | 60 |
| 4.3 | main 函数 | 61 |
| * 4.4 | 函数参数的默认值 | 61 |
| * 4.5 | 引用和函数参数的传递 | 62 |
| 4.5.1 | 引用的概念 | 62 |
| 4.5.2 | 引用作为函数的返回值 | 63 |
| 4.5.3 | 参数传值 | 63 |
| 4.5.4 | 参数传引用 | 64 |
| 4.5.5 | 常引用 | 65 |
| * 4.6 | 内联函数 | 66 |
| * 4.7 | 函数的重载 | 67 |
| * 4.8 | 库函数和头文件 | 68 |
| 4.9 | 小结 | 70 |
| 习题 | | 70 |
| 第 5 章 | 数组 | 72 |
| 5.1 | 一维数组 | 72 |
| 5.2 | 数组的大小限制 | 74 |
| 5.3 | 二维数组 | 75 |
| 5.4 | 数组的初始化 | 76 |
| 5.5 | 数组作为函数的参数 | 78 |
| 5.6 | 数组越界 | 79 |
| 5.6.1 | 什么是数组越界 | 79 |
| 5.6.2 | 数组越界的后果 | 80 |
| 5.7 | 小结 | 81 |
| 习题 | | 81 |
| 第 6 章 | 字符串 | 83 |
| 6.1 | 字符串常量 | 83 |
| 6.2 | 用字符数组存放的字符串 | 84 |
| 6.2.1 | 用一维数组存放字符串 | 84 |
| 6.2.2 | 用二维数组存放字符串 | 86 |
| 6.3 | 字符串函数用法示例 | 86 |
| * 6.4 | 用 string 对象处理字符串 | 87 |
| 6.4.1 | 定义 string 对象 | 88 |
| 6.4.2 | string 对象的输入输出 | 88 |
| 6.4.3 | string 对象的赋值 | 88 |

| | | |
|--------------|-------------------------------|------------|
| 6.4.4 | string 对象的运算 | 89 |
| 6.4.5 | string 对象用法示例 | 89 |
| 6.5 | 小结 | 90 |
| | 习题 | 90 |
| 第 7 章 | 指针 | 92 |
| 7.1 | 指针的基本概念 | 92 |
| 7.2 | 指针的作用 | 94 |
| 7.3 | 指针的互相赋值 | 94 |
| 7.4 | 指针运算 | 95 |
| 7.5 | 空指针 | 96 |
| 7.6 | 指针作为函数参数 | 96 |
| 7.7 | 指针和数组 | 98 |
| 7.8 | 常量指针 | 100 |
| 7.9 | 字符串和指针 | 101 |
| 7.9.1 | 普通字符串和指针的关系 | 101 |
| 7.9.2 | string 对象和 char * 指针的关系 | 101 |
| 7.9.3 | 字符串操作库函数 | 102 |
| 7.10 | void 指针和内存操作库函数 | 106 |
| 7.11 | 函数指针 | 107 |
| 7.11.1 | 函数指针的定义 | 107 |
| 7.11.2 | 函数指针的应用 | 108 |
| 7.12 | 指针和动态内存分配 | 109 |
| 7.13 | 指向指针的指针 | 111 |
| 7.14 | 指针数组 | 112 |
| 7.15 | 误用无效指针 | 117 |
| 7.16 | 小结 | 117 |
| | 习题 | 118 |
| 第 8 章 | 自定义数据类型 | 120 |
| 8.1 | 结构 | 120 |
| 8.1.1 | 结构的定义和使用 | 120 |
| 8.1.2 | 访问结构变量的成员变量 | 122 |
| 8.1.3 | 结构变量的初始化 | 122 |
| 8.1.4 | 结构数组 | 123 |
| 8.1.5 | 指向结构变量的指针 | 123 |
| 8.1.6 | 动态分配结构变量和结构数组 | 125 |
| 8.1.7 | 结构变量或引用作为函数形参 | 125 |
| 8.2 | 联合 | 126 |

| | | |
|---------------|------------------------------|------------|
| 8.3 | 枚举类型 | 128 |
| 8.4 | 用 typedef 定义类型 | 129 |
| 8.5 | 小结 | 130 |
| | 习题 | 130 |
| 第 9 章 | 程序设计的基本思想 | 133 |
| 9.1 | 枚举 | 133 |
| 9.2 | 递归 | 138 |
| 9.3 | 二分 | 144 |
| 9.4 | 算法的时间复杂度及其表示法 | 145 |
| 9.5 | 小结 | 147 |
| | 习题 | 147 |
| 第 10 章 | C++ 程序结构 | 148 |
| 10.1 | 全局变量和局部变量 | 148 |
| 10.2 | 静态变量、自动变量和寄存器变量 | 149 |
| 10.3 | 标识符的作用域 | 152 |
| 10.4 | 变量的生存期 | 153 |
| 10.5 | 预编译 | 153 |
| | 10.5.1 宏定义 | 154 |
| | 10.5.2 文件包含 | 154 |
| | 10.5.3 条件编译 | 156 |
| 10.6 | 命令行参数 | 157 |
| 10.7 | 多文件编程 | 159 |
| | 10.7.1 C++ 程序的编译过程 | 159 |
| | 10.7.2 多文件共享全局变量 | 160 |
| | 10.7.3 静态全局变量和静态全局函数 | 161 |
| | 10.7.4 多文件编程中的内联函数 | 162 |
| | 10.7.5 用条件编译避免头文件的重复包含 | 162 |
| 10.8 | 小结 | 163 |
| | 习题 | 163 |

第 2 篇 面向对象的程序设计

| | | |
|---------------|----------------------|------------|
| 第 11 章 | 类和对象初步 | 167 |
| 11.1 | 结构化程序设计的不足 | 167 |
| 11.2 | 面向对象程序设计的概念和特点 | 169 |
| 11.3 | 类的定义和使用 | 170 |
| 11.4 | 类的示例程序剖析 | 171 |

| | | |
|---------------|------------------------|------------|
| 11.5 | 访问对象的成员 | 172 |
| 11.6 | 类成员的可访问范围 | 173 |
| 11.7 | 内联成员函数 | 176 |
| 11.8 | 小结 | 176 |
| | 习题 | 176 |
| 第 12 章 | 类和对象进阶 | 179 |
| 12.1 | 构造函数 | 179 |
| 12.1.1 | 构造函数的概念和作用 | 179 |
| 12.1.2 | 构造函数在数组中的使用 | 181 |
| 12.1.3 | 复制构造函数 | 183 |
| 12.1.4 | 类型转换构造函数 | 187 |
| 12.2 | 析构函数 | 188 |
| 12.3 | 构造函数、析构函数和变量的生存期 | 190 |
| 12.4 | 静态成员变量和静态成员函数 | 191 |
| 12.5 | 常量对象和常量成员函数 | 194 |
| 12.6 | 成员对象和封闭类 | 196 |
| 12.7 | const 成员和引用成员 | 199 |
| 12.8 | 友元 | 199 |
| 12.8.1 | 友元函数 | 200 |
| 12.8.2 | 友元类 | 201 |
| 12.9 | this 指针 | 202 |
| 12.9.1 | C++ 程序到 C 程序的翻译 | 202 |
| 12.9.2 | this 指针的作用 | 203 |
| 12.10 | 在多个文件中使用类 | 204 |
| 12.11 | 小结 | 204 |
| | 习题 | 205 |
| 第 13 章 | 运算符重载 | 209 |
| 13.1 | 运算符重载的概念和原理 | 209 |
| 13.2 | 重载赋值运算符“=” | 210 |
| 13.3 | 浅复制和深复制 | 213 |
| 13.4 | 运算符重载为友元函数 | 215 |
| 13.5 | 实例——长度可变的整型数组类 | 215 |
| 13.6 | 重载流插入运算符和流提取运算符 | 218 |
| 13.7 | 重载强制类型转换运算符 | 220 |
| 13.8 | 重载自增、自减运算符 | 221 |
| 13.9 | 运算符重载的注意事项 | 222 |
| 13.10 | 小结 | 223 |

| | |
|---|------------|
| 习题 | 223 |
| 第 14 章 继承与派生 | 227 |
| 14.1 继承和派生的概念 | 227 |
| 14.2 正确处理类的复合关系和继承关系 | 231 |
| 14.3 protected 访问范围说明符 | 233 |
| 14.4 派生类的构造函数和析构函数 | 234 |
| 14.5 多层次的派生 | 236 |
| 14.6 包含成员对象的派生类 | 237 |
| 14.7 公有派生的赋值兼容规则 | 237 |
| 14.8 基类与派生类的指针的互相转换 | 238 |
| 14.9 私有派生和保护派生 | 240 |
| 14.10 派生类和赋值运算符“=” | 241 |
| 14.11 多重继承 | 241 |
| 14.11.1 多继承的概念及其引发的二义性 | 241 |
| 14.11.2 用“虚继承”解决二义性 | 244 |
| 14.12 小结 | 245 |
| 习题 | 245 |
| 第 15 章 多态与虚函数 | 247 |
| 15.1 多态的基本概念 | 247 |
| 15.2 多态的作用 | 250 |
| 15.3 多态的实现原理 | 256 |
| 15.4 关于多态的注意事项 | 258 |
| 15.5 虚析构函数 | 261 |
| 15.6 纯虚函数和抽象类 | 263 |
| 15.7 小结 | 264 |
| 习题 | 264 |
| 第 16 章 输入输出流 | 269 |
| 16.1 流类 | 269 |
| 16.2 标准流对象 | 270 |
| 16.3 使用流操纵算子控制输出格式 | 271 |
| 16.4 调用 cout 的成员函数 | 274 |
| 16.5 cin 的高级用法 | 275 |
| 16.5.1 判断输入结束 | 275 |
| 16.5.2 istream 类的成员函数 | 277 |
| 16.6 printf、scanf 等 C 语言标准输入输出库函数 | 282 |
| 16.7 小结 | 286 |

| | |
|----------------------------------|------------|
| 习题 | 287 |
| 第 17 章 文件操作 | 288 |
| 17.1 文件的概念 | 288 |
| 17.2 C++ 文件流类 | 289 |
| 17.3 文件的打开和关闭 | 290 |
| 17.4 文件的读写 | 292 |
| 17.4.1 文本文件的读写 | 292 |
| 17.4.2 二进制文件的读写 | 294 |
| 17.4.3 操作文件读写指针 | 298 |
| 17.5 文本方式打开文件与二进制方式打开文件的区别 | 299 |
| 17.6 小结 | 300 |
| 习题 | 300 |

第 3 篇 泛型程序设计

| | |
|------------------------------------|------------|
| 第 18 章 泛型程序设计与模板 | 305 |
| 18.1 函数模板 | 305 |
| 18.2 类模板 | 310 |
| 18.3 类模板中的非类型参数 | 314 |
| 18.4 类模板与继承 | 315 |
| 18.5 类模板和友元 | 316 |
| 18.6 类模板中的静态成员 | 320 |
| 18.7 在多个文件中使用模板 | 321 |
| 18.8 小结 | 321 |
| 习题 | 321 |
| 第 19 章 标准模板库 STL | 324 |
| 19.1 STL 中的基本概念 | 324 |
| 19.1.1 容器 | 325 |
| 19.1.2 迭代器 | 326 |
| 19.1.3 算法 | 331 |
| 19.1.4 STL 中的“大”、“小”和“相等”的概念 | 332 |
| 19.2 顺序容器 | 333 |
| 19.2.1 动态数组 vector | 333 |
| 19.2.2 双向链表 list | 336 |
| 19.2.3 双向队列 deque | 339 |
| 19.3 函数对象 | 339 |
| 19.4 关联容器 | 346 |
| 19.4.1 关联容器的预备知识: pair 类模板 | 347 |

| | | |
|--------|----------------|-----|
| 19.4.2 | multiset | 348 |
| 19.4.3 | set | 351 |
| 19.4.4 | multimap | 352 |
| 19.4.5 | map | 355 |
| 19.5 | 容器适配器 | 357 |
| 19.5.1 | stack | 357 |
| 19.5.2 | queue | 358 |
| 19.5.3 | priority_queue | 359 |
| 19.6 | STL 算法分类 | 360 |
| 19.7 | 不变序列算法 | 361 |
| 19.8 | 变值算法 | 364 |
| 19.9 | 删除算法 | 369 |
| 19.10 | 变序算法 | 372 |
| 19.11 | 排序算法 | 376 |
| 19.12 | 有序区间算法 | 378 |
| 19.13 | string 类详解 | 382 |
| 19.14 | bitset | 387 |
| 19.15 | 小结 | 389 |
| | 习题 | 389 |

第 4 篇 C++ 高级主题

| | | |
|---------------|-----------------|------------|
| 第 20 章 | C++ 高级主题 | 395 |
| 20.1 | 强制类型转换 | 395 |
| 20.2 | 运行时类型检查 | 399 |
| 20.3 | 智能指针 auto_ptr | 400 |
| 20.4 | C++ 异常处理 | 402 |
| 20.5 | 名字空间 | 411 |
| 20.6 | C++ 11 新特性管窥 | 418 |
| 20.6.1 | 智能指针 shared_ptr | 418 |
| 20.6.2 | 无序容器(哈希表) | 419 |
| 20.6.3 | 正则表达式 | 420 |
| 20.6.4 | Lambda 表达式 | 421 |
| 20.7 | 小结 | 422 |
| | 习题 | 422 |
| 附录 A | 魔兽世界大作业 | 426 |
| | 参考文献 | 428 |