

Oracle R&D Senior Director *Ernest Chen* 专业推荐

零起点 学通C++

-(多 教学)-



1DVD大型多媒体
教学课堂

真正适合中国人学习的**C++ Primer**

针对重要的概念精心设计了**666**个实用范例，囊括大量经验和技巧，重点难点全面透析

50小时1000多节专家视频讲解，部分视频已在网上发布，在**C++**开发者中广为流传，深受读者欢迎

范磊 编著



零起点 学通C++

-(多媒体范例教学)-

 科学出版社

内 容 提 要

本书总结了十几本 C++ 图书及教材, 从零开始、由浅入深、层层递进、细致而又详尽地讲解 C++ 这门大型编程语言。

本书知识系统全面, 拥有字典般的容量, 可随用随查, 涵盖指针、面向对象、操作符重载、流、命名空间、模板、异常处理、宏等主流 C++ 开发技术。为了使读者能够活学活用, 本书针对重要的概念精心设计了 438 个实用范例, 囊括大量经验和技巧, 即使是已从事 C++ 工作多年的朋友, 也能从中汲取新的“养料”。

本书适合从未学习过任何编程语言的新手, 以及学习 C++ 多年, 仍旧不能融会贯通的读者, 对于正在使用 C++ 进行开发的程序员也有很好的参考价值。

图书在版编目 (CIP) 数据

零起点学通 C++: 多媒体范例教学/范磊编著. —北京:
科学出版社, 2010. 7

ISBN 978-7-03-028209-5

I. ①零… II. ①范… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 127254 号

责任编辑: 赵东升 王海霞 / 责任校对: 刘雪连

责任印制: 新世纪书局 / 封面设计: 周智博

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

中国科学出版集团新世纪书局策划

北京市艺辉印刷有限公司印刷

中国科学出版集团新世纪书局发行 各地新华书店经销

*

2010 年 10 月 第 一 版

开本: 16 开

2010 年 10 月第一次印刷

印张: 62.5

印数: 1—3 000

字数: 1520 000

定价: 108.00 元 (含 1DVD 价格)

(如有印装质量问题, 我社负责调换)

前言

本书的配套视频在学吧学吧 (www.xue8xue8.com) 和网易学院 (tech.163.com) 发布后, 很快在各大网站流传开, 不少读者来信和我交流 C++ 的学习心得。为了方便大家学习, 我编写了《从新手到高手: C++全方位学习》一书。自该书出版后, 给我来信的读者更多了, 他们提出了不少宝贵的意见和建议, 因此在《从新手到高手: C++全方位学习》的基础上进行修订并增加了一些内容后, 推出了本书。

来信的朋友普遍提到的问题是: C++ 很难学, 看了好几本书, 有不少地方还是难以理解。C++ 真的这么难, 有必要看很多的教材和资料吗?

为什么会感到 C++ 难学

C++ 其实不难学, 只是由于许多书的组织框架和表达方式等方面的原因无意中增加了学习及研究 C++ 的难度。

就拿众多人推崇为圣经的 *C++ Primer* 来说, 与其把它看作一本 C++ 教材, 不如把它当作 C++ 字典。这是因为该书没有分清主次轻重, 在读者不了解模板为何物时, 对 STL 过早地进行了讲解, 而一些相对比较简单概念, 却又放到后面, 导致读者要来回跳跃式阅读, 类似于查阅字典。对于初学者来说, 这样的字典式图书是不需要的, 他们需要的是一本能够循序渐进、快速、扎实地掌握 C++ 语言的图书。

国外图书的特点是喜欢用大量晦涩而又抽象的语言来解释一个概念, 而要了解这些概念, 又需要查阅其他相关书籍。当你彻底明白作者的意图时, 你不禁错愕, 这些晦涩而又抽象的语言完全可以用一句简单的话来代替。

这是因为中国人和外国人在思维和表达上不同, 外国人喜欢用抽象的语言来解释抽象的概念, 这是他们几百年来养成的习惯。习惯了形象思维的中国读者在阅读他们的书籍时自然会不适应, 这就在无形中增加了学习 C++ 的难度。

写作本书的目的

C++ 是一门非常重要的语言, 它甚至可以上升到思想的高度, 它的思想被其他语言继承并沿袭。每一种新语言的诞生, 都可以找到 C++ 的影子, 或者说它们都来自 C++, 只不过穿的外衣不同而已。

精通了 C++, 学习别的语言就很容易了, 所以我们迫切需要一本适合中国人思维的

C++ Primer，它不是字典，却需要字典那样的容量；它不是小说，却需要像小说那样通俗易懂、引人入胜，以至于可以让从未学习过编程语言的初学者也能成为高手；同时，本书并不会因为它的通俗而流俗，资深 C++ 程序员在阅读该书时仍然能够获得质的飞跃。

基于这个理念，我编写了本书，其中的艰辛自不必说，不过令人欣慰的是，本书确实能够行之有效地帮助后来人少走弯路。

本书特色

本书不按照以往的模式，而是从一个最简短的 C++ 程序讲起，通过对这个程序的编写引申出一系列相关知识，然后不断地扩大和完善该程序，让读者循序渐进地进行学习，同时也能深刻理解 C++ 的各个语法要领。相对于同类书来说，本书具备以下特点。

1. 实用性强

毛主席说过：“要想知道梨子的味道，就要亲口尝一尝。”本书在讲述完一个新概念后，都会用一个实例来让读者检验学习成果。比如介绍“链表”时创建了一个简单的图书管理系统，在讲解“代码重用”时又用面向对象的思想重写了这个系统，在后面又将图书管理系统的数据库部分做成了一个类，以实现代码与数据的分离。而通过不断完善这个图书管理系统，可让读者把各部分知识系统地结合起来。

2. 章节安排合理

本书不会像 *C++ Primer* 一样，在读者不了解模板为何物时，过早地讲解 STL，而是根据读者的理解能力合理安排学习进度。

比如，本书先讲解简单的函数，了解类的概念后，再讲类的函数，学习了模板后，再去讲解模板函数，这样依次递进，使读者感觉不到难度的提高。同时，这些概念的传授并不是为了讲而讲，而是因为需要。比如说，当普通变量解决不了问题的时候，出现了数组，而数组解决不了问题的时候，又出现了链表。我们按照需求关系将它们的顺序合理排列起来，让读者认识到 C++ 这门语言是为了解决实际问题而创造出来的。

3. 通俗易懂

为了使大多数读者都看懂，本书尽量使用生活中常见的例子来举例说明。比如讲解友元一章时，列举了朋友的例子来说明两个友元类的关系。

我们把类 A 设置为类 B 的友元，这并不会使类 B 成为类 A 的友元。换句话说，你愿意把你的秘密告诉我，但是这并不意味着我愿意把秘密告诉你。

而讲解友元类的作用时，则用了一个电视机与遥控器的实例：将遥控器声明为电视机的友元后，遥控器就可以操作电视机了。

4. 观其大略的教学方法

毛主席说过：“看问题、解决问题要抓主要矛盾，主要矛盾抓住了，一切问题都会迎刃而解。”本书讲解从问题的本质入手，对问题的旁枝末叶则不予过多计较。

比如说虚函数那一章，大部分书籍都是从内存的角度来研究虚函数，这样研究的结果就是导致几十页也解释不清虚函数的概念。不光读者看得头晕，而且作者自己也犯怵。本书从全局分析问题，使读者迅速看清虚函数的本质，只用一个拳击游戏的实例就演示了虚函数的作用，同时借助游戏提高了读者的兴趣。

人生在世只有短暂的三万多天，即使你一刻不停地阅读书籍，也不能读遍世上所有的书籍，因此我们一定要掌握一种良好的学习方法，即要善于分清事物的本质，从大处着眼，而不是过分地去追求细枝末节，这样才能融会贯通，一通百通。

5. 用最简单的语言阐述最复杂的问题

尽量用通俗易懂的话来概括复杂事物，而不是去将它神话。比如，我们在“多态性”一章中如下概括多重继承的作用：

为什么要使用多重继承呢？举个非常简单的例子，一对夫妻生了一个儿子，假设这个儿子只继承了父亲的特点，那么就是单一继承。但是这种情况往往是不多见的，通常都是儿子既继承了父亲的特点，又继承了母亲的特点。

在没有多重继承之前，假如这个儿子既继承了父亲的聪明，又继承了母亲的美丽，那么就只好在父亲类中再添加一项功能，就是美丽。

假设这个父亲生了很多儿子和女儿，那么事情就更麻烦了，每个儿子或者女儿都要在父亲类中放一个作为接口的虚函数，那么父亲类将会变得臃肿不堪。而且所有的儿子和女儿都可以共享这些作为接口的函数，那么儿子和女儿也将变得臃肿不堪。

而对于抽象类，则是这样概括的：

由于该类没有什么实际作用，因此它是一个抽象类。就像人类，它不是某个具体的人，没有实际功能，只是在人的大脑中抽象出用来概括人类的名词。不过它可以保证所有进入人类这个级别的对象都具有人类的行为或特征，不可能存在狮子或者恐龙也成为人类的这种情况。

同样是阐述以上问题，*C++ Primer* 却用了长达数十页甚至上百页，甚至还用了大量的篇幅来演示那些已经淘汰的编译器编译后的结果，对于已经淘汰的编译器，我们还需要浪费时间对它们研究吗？

6. 始终站在存在即合理的角度来阐述问题

C++ Primer 的优点是知识系统全面，就像一本大型的字典，很多内容都可以在该书中查到，甚至是已经淘汰的知识。本书并不会因为是“零起点”而借故将内容缩水，相反，由于本书语言精练，淘汰的知识不再讲述，因此用与 *C++ Primer* 相同的篇幅却讲出了 *C++ Primer* 所不能表达的知识，我们可将该书看作一本中国人自己的 *C++ Primer*，一本 C++ 的圣经。

升级修订说明

任何事物都不是一下就完善的，而是逐渐完善的，本书在经过数十次整理和修改后，终于近乎完美。

那么本书及配套光盘与《从新手到高手：C++全方位学习》相比，到底有哪些不同呢？以下简单列出了两者的不同之处。

- ◆ 高级篇增加了实例，用来巩固并掌握高级篇的内容，每章都有一个比较大的实例应用并总结学习过的知识。
- ◆ 文字描述更加精准，更加易懂，错误几乎全部修正。
- ◆ 增加了专用播放器，解决了视频播放问题。
- ◆ 增加了高级篇的视频。
- ◆ 对部分视频进行修改，消除了一些语误。

光盘内容

为了达到教学的最佳效果，作者综合了 1000 多个读者的反馈意见后，整理出 50 小时 1000 多节视频教程，并包含 666 个实例源程序，与图书内容对应，可让读者大幅度提高学习效率。

本书是作者长期教学和 C++ 开发经验的总结，在编写过程中参阅了下列书籍。

- ◆ 《C++ PRIMER 中文版（第 4 版）》，STANLEY B. LIPPMAN 等。
- ◆ *The C++ Programming Language*, BJARNE STROUSTRUP.
- ◆ *Effective C++*, SCOTT MEYERS.
- ◆ 《设计模式：可复用面向对象软件的基础》，ERICH GAMMA 等。
- ◆ *International Standard for C++*, ISO/IEC.
- ◆ *Modern C++ Design*, ANDREI ALEXANDRESCU.
- ◆ 《C++编程金典》，H.M.DEITEL.
- ◆ 《21 天学通 C++》，JESSE LIBERTY, BRADLEY JONES.
- ◆ *Exceptional C++*, HERB SUTTER.
- ◆ 《数据结构（C 语言版）》，ELLIS HOROWITZ.
- ◆ *Effective STL*, MEYERS, SCOTT.

本书也得到了软件行业的一些朋友的大力支持，在此深表感谢。在本书的编写过程中，我们力求精益求精，但难免存在一些不足之处，敬请广大读者批评指正。联系方式：najiushifeng@126.com。

编 者

2010 年 8 月

读者来信

本书的上一版《从新手到高手：C++全方位学习》出版后，不少读者给我写信，有的来咨询技术问题，有的给予了热情洋溢的赞扬，这里感谢大家的支持。下面收录了部分读者的来信和网络上的评论。

发件人：wangzhenjixie

发送时间：2009 年 12 月 9 日 20:20

收件人：najiushifeng

主题：衷心感谢您，范老师

范老师，您好！

我是您的《C++全方位学习》的受益者。我是黑龙江省哈尔滨市哈尔滨量具刃具集团的一名技术人员。07 年大学毕业后参加工作的。我大学学的是机械设计，之前几乎没有接触过计算机编程，因为工作的需要（我们公司需要齿轮自动化测量软件），我们公司急需计算机编程人员。我一直想学习编程，也在网上找过很多教程，但讲座讲得很繁琐，所以一直没什么收获。上个月偶然间发现您的教程，感觉受益颇深。您的讲座很直观，对于我这样的初学者来说真的是帮助很大。我每天下班后一直学习到晚上 12 点多，虽然很疲倦，但很有收获，很充实。

我会继续边听您的讲座边练习编程，直到把您所有的教程都学完。

虽然没有见过您本人，但您是我计算机编程的启蒙老师。在这里我对您说一声：Mr Fan, Thank you very much!

发件人：654662702@qq.com

发送时间：2009 年 12 月 18 日 5:36

收件人：木马

主题：答复读者

老师，你好！我反复看着您的回复，虽然是以前的，但每每都感觉有收获。

我现在一边在第二遍学习您的书，一边想着自己的软件的雏形，想着想着，我终于明白过来了。您的书介绍了 C++ 的基础，打个通俗的比方，就像我要造一辆车，您先后介绍了要在哪个部位造，需要什么零件，各个零件的工作原理，等等，然后最终将零件组装起来。

这是我的一点浅谈，不知对不对，老师说有下部书要出，能详细介绍一下吗？能否提供书的目录？

我是 C++ 的初学者，只接触了您的书，如果我要成为一个真正的程序员，还需要学什么？买什么样的书？老师您能给我们打包一套教程吗？包括程序的编写、程序的反编译，总的思路是“从新手到骇客全方位学习电脑语言”？

好了，今天就说到这，我能理解老师，每天有那么多的工作，还要回复那么多网友的留言，知道老师是完全没有必要回复我的这封信，但我还是希望能接到老师的回复!!

谢谢! 这是元旦前给您的最后一封邮件。

发件人: 654662702@qq.com

发送时间: 2009 年 12 月 26 日 6:41

收件人: najiushifeng

主题: www_dssz_com_FinData1.0

老师，您好!

我本想在已有的软件源码中获得灵感，想到了 Linux 的源码是公开的，就搜了一下，结果又发现了您写的书，我好高兴啊，看来老师也是一位 Linux 高手，越发让我对老师崇敬和羡慕啦!

老师，今天去信是想了解一下您下本书的一些目录，好让我先大概了解一下。

老师，上次上传的附件不知有没有接到，不知能不能就附件获得的软件信息给我做个解答呢?

第三，也是最重要的，我想成为一名程序员，但在茫茫书海中又无法理智的选择，呵呵。我现在知道在学校学习和自学的区别了，学校是系统地教学，而自学只能是瞎子摸象——走一步算一步，最后学得多而散。所以我想请老师给我的学习做一个系统的大纲，这样可能学得更快一些，更容易去找些工作。谢谢老师!

发件人: JiN 者

发送时间: 2010-01-4 19:40:38

收件人: 范先生

主题: 不错的书

范先生，您好:

看到您出版的《C++全方位学习》，感觉它确实弥补了国内 C++图书的一个空白。

您在前言中提到了 *C++ Primer*，说这本书很火，我想最重要的原因还是因为写这本书的人资格够硬，如此而已。作为一个学过了 C、MFC、汇编、Windows 编程和一点驱动开发的人，我其实不是想说我知识有多么猛（我很菜的其实）。我只是想说，我真的看过了很多书。我之所以买 *C++ Primer* 这本书，的确，是因为这本书太火了。作为一个小学员（不是小学生，呵呵），好像我根本就没有资格来评论这本由 C 国际知名大师，还有什么 C 委员会的什么什么人写的书。但是，一本书，我觉得最有资格评论的，最终还是这本书的广大读者。

首先我想说的是，这本书绝不应该叫做 *C++ Primer*，我为它想了一个更合适的名称：《C 编译器参考手册》。当你写 C 代码，编译器无法通过编译的时候，你来查这本书就对了。整本书根本没有一丁点的原理，也没有任何的分析，读起来感觉好像是规范手册、流水账一般。学编程是这样学的吗？甚至连编译器会自动为类的某些成员函数添加 this 指针这一回事，书上好像都没有怎么去说明白，一些其实很简单的东西，硬是要说得不明不白（我并不是指翻译的问题），作者好像很热衷于把一些简单的概念（或方法）极度地复杂化，我想他是想照顾新手，结果却是适得其反。

我读一些地方的时候，往往读了两三遍都不知所云，导致我极度怀疑我以前所学的知识，到我

真正弄明白的时候，恍然大悟，原来说的是这个。随后我的问题就是：你为什么要这样说？你为什么要把它复杂到这种程度？你为什么越说越让人不明白？

C++ Primer 整本书都在用同一个例子，就是开篇时提到的书店的例子。整本书啊，这意味着什么？看上去是“紧紧围绕着一个中心点”，很好。但事实上是，除非你一口气读完这本书，并且能过目不忘，不然，到了后面提到的书店的例子里，你根本不知道他想说什么。另外，翻译也很成问题。作者是大师级人物，他认为他说话一定要完全标准，要天衣无缝，一定要严谨。然而，这样会把话说得很复杂，并且很啰唆，这样的书本身就不适合初学者。更甚，翻译的人并不是不会翻译，而是认为这样的一本书，他翻译起来得小心翼翼再小心，谨慎谨慎再谨慎！这样带来的后果就是：用汉字来说英语。

到目前为止，计算机事实上只能认 0 与 1，所谓的类，所谓的什么面向对象编程，这些全部都是编译器的事情。死记硬背地学习编程（*C++ Primer* 这本书无形中这样提倡了）。个人认为，这本书真的不适合初学者。

发件人：chendewu-001

发送时间：2010 年 1 月 7 日 22:32

收件人：najiushifeng@126.com

主题：C++问题

范老师：

您好！我是武汉大学印刷工程专业的陈德武，今年大四，被保送到兰州大学计算机专业上研究生。我们专业没开过 C++ 课程，所以我从 eNet 网络学院下载了您讲授的视频教程《从新手到高手：C++ 全方位学习》，自学 C++。您讲得很详细、生动，现在快学完了，自己也能编一些小程序了，感谢您将视频免费分享。但我在学习第十五章链表时，按您讲的内容编写的程序只能实现建立链表，显示、插入、删除、统计链表都不能实现，编译时程序无错误，运行时就崩溃，我将源程序给您发过来了，希望您在百忙之中抽出点时间帮我解决一下这个问题，谢谢！

发件人：shaojinzero2005

发送时间：2010 年 1 月 9 日 22:33

收件人：najiushifeng

主题：读者的来信

范磊老师，您好！

在今年的暑假期间，我在网上看到了您制作的 C++ 视频，感觉十分受用。虽然我之前没有学过编程，但我很快被 C++ 这一门博大精深的编程语言吸引住了，特别是您的教学方式很特别，我感觉受益匪浅。最近，我买了一本您写的《从新手到高手：C++ 全方位学习》，发现配套光盘中的视频只有 17 章，我想问的是，其他部分的视频我能否通过其他的途径获得呢？

感谢您在百忙之中抽空看我这封信，我是广东的一名初中生，今年 16 岁，也许我以后会踏上成为一名程序员的道路。尽管路途坎坷，可是我相信有您这样的先行者引导着我们这些初学者，我一定会把 C++ 学好的。谢谢您！

发件人：郁闷草

发送时间：2010-04-06 19:03:27

收件人：najiushifeng

主题：范老师

范老师：

您好！您的C++讲得很好，比我们老师讲得好多了，因为我们老师讲课，我都听不懂，不知道在讲什么。

看了您的几段视频教程，觉得您讲课通俗易懂，而且喜欢重复，这点我比较喜欢，因为老师讲东西，学生不可能一遍就能记住，是一个不断重复的过程。

我想问问范老师，您的C++视频除了在硅谷动力上发布了，还有在其他网站上发布吗？因为我们学校网速非常慢，在线看的话，缓冲不起来，所以我想打包下载。谢谢！

感 谢

一本书的出版凝聚了很多人的心血，很多人是本书幕后的英雄，他们付出了很多，在此我想对他们表达我的感激之情。

首先要感谢的是我的编辑兼策划赵东升。当一颗金子放在大家面前时，所有人都能发现，但是要从瓦砾中找寻金粉，却不是一件很容易的事情。赵编辑从瓦砾堆中发现了这本书，并且加以推荐，无论什么样的困难和阻力都没能影响他的判断，在物欲横流的今天，这是非常难能可贵的。

同样要感谢的是赵编辑的领导。当赵编辑推荐这本书时，领导丝毫没有犹豫就通过了。这需要相当大的勇气和魄力，毕竟作者不是耳熟能详的名人。

纵使你是千里马，假如没有伯乐来赏识，也终究会被埋没，我很幸运，我碰到了伯乐，而且有两个伯乐，这使我既感激又幸福。

接下来要对本书的责任编辑王海霞、责任校对刘雪连、封面设计周智博、责任印制科海新世纪书局全体员工表示衷心的感谢，该书印刷精美，排版工整，无论从编排还是封面设计以及纸张和印刷都可谓上乘，足以说明科学出版社对本书很重视，这使我倍感荣幸。

最后，我要感谢的是网络上支持我的网友和购买图书的读者，没有你们的支持，我没有动力来继续这一版的修订，没有你们的支持，我没有动力继续制作后面的视频，在这里，我也想对你们说一声：谢谢。

目 录

第 1 篇 初级篇

第 1 章 初识 C++	4	4.6 整型概述	29
1.1 C++简介	4	4.7 整型变量的定义	31
1.2 C++与 C 的区别	5	4.8 浮点型变量	33
1.3 学习 C++之前需要先学 C 吗	6	4.9 常量	33
1.4 C++与其他语言的区别	6	4.10 枚举型常量	34
1.5 C++的版本以及安装问题	6	第 5 章 if 语句与运算符	36
第 2 章 做一个最简短的 C++程序	7	5.1 语句的定义	36
2.1 简单的屏幕输出小程序	7	5.2 块的定义	36
2.2 输出语句的使用	8	5.3 表达式的定义	37
2.3 std::介绍	9	5.4 运算符的定义	38
2.4 iostream 与 iostream.h 的区别	10	5.4.1 赋值运算符的定义	38
2.5 重名问题	11	5.4.2 数学运算符的定义	38
2.6 注释	13	5.4.3 赋值运算符与数学运算符的联合	39
2.7 总结	13	5.5 自加与自减	39
第 3 章 初步了解函数	14	5.5.1 前置	40
3.1 一个简单的函数	14	5.5.2 后置	40
3.2 函数的传参	15	5.6 表达式的优先级	41
3.3 函数的返回值、参数与变量	16	5.7 关系运算符	41
3.4 函数的声明与定义	17	5.8 if 语句	42
3.5 局部变量	19	5.8.1 else 语句	43
3.6 全局变量	20	5.8.2 else if 语句	44
3.7 总结	21	5.8.3 if 语句的嵌套	45
第 4 章 C++数据类型	22	5.9 逻辑运算符及其使用	48
4.1 变量的定义	22	5.9.1 逻辑“与”	49
4.2 将变量及数据存储在内存中	23	5.9.2 逻辑“或”	49
4.3 布尔型变量	24	5.9.3 逻辑“非”	50
4.4 字符型变量	25	5.9.4 逻辑运算符的优先级	51
4.5 wchar_t 双字符型变量	28	5.9.5 运算式的真假关系	53
		5.10 三目运算符	54
		5.10.1 三目运算符的优先问题	55

5.10.2 三目运算符的使用问题	56	7.3.4 continue 语句	83
5.10.3 三目运算符的型别问题	56	7.3.5 break 语句	84
5.10.4 三目运算符在字符型变量中的 应用	56	7.3.6 永不休止的 while 循环	85
5.11 复杂嵌套的 if 语句	57	7.4 do... while 循环	86
5.12 总结	58	7.5 for 循环	87
第 6 章 面向对象	59	7.5.1 灵活的 for 循环	89
6.1 面向对象程序语言的主要特征	59	7.5.2 条件为空的 for 循环	89
6.2 类、对象和成员	60	7.5.3 执行为空的 for 循环	90
6.3 类、对象和成员的使用方法及区别	61	7.5.4 嵌套的 for 循环	91
6.3.1 声明一个类	61	7.6 switch 语句	92
6.3.2 命名习惯	62	7.6.1 switch 语句常见错误	94
6.3.3 定义一个对象	62	7.6.2 switch 的菜单功能	95
6.3.4 类与对象的区别	62	7.7 总结	96
6.3.5 对象与成员的关系	63	第 8 章 指针	97
6.3.6 不要给类赋值	63	8.1 什么是地址	97
6.3.7 对象只能调用类中存在的方法	63	8.2 用指针来保存地址	98
6.4 公有	65	8.2.1 空指针	98
6.5 私有	65	8.2.2 指针与变量类型	99
6.6 成员函数的声明和定义	68	8.2.3 用指针来访问值	99
6.7 内联函数	70	8.2.4 指针地址、指针保存的地址和 该地址的值	100
6.7.1 普通内联函数	70	8.2.5 指针对数值的操作	101
6.7.2 内联成员函数	71	8.2.6 更换指针保存的地址	102
6.8 头文件与源文件	72	8.3 为什么使用指针	103
6.9 const 成员函数	73	8.3.1 栈和堆	103
6.10 构造函数	74	8.3.2 用指针创建堆中空间	107
6.11 默认构造函数	75	8.3.3 用指针删除堆中空间	108
6.12 析构函数	76	8.4 动态内存	110
6.13 析构对象数组	77	8.4.1 内存泄漏	110
6.14 总结	78	8.4.2 在堆中创建对象	110
第 7 章 循环语句	79	8.4.3 在堆中删除对象	111
7.1 循环语句的前身——goto 语句	79	8.4.4 访问堆中的数据成员	112
7.2 慎用 goto 语句	80	8.4.5 在构造函数中开辟内存空间	113
7.3 while 语句	80	8.4.6 对象在栈与堆中的不同	114
7.3.1 带运算符的 while 语句	81	8.5 this 指针	116
7.3.2 以字符为条件的 while 语句	81	8.6 指针的常见错误	117
7.3.3 限定 while 循环的次数	82	8.7 指针运算	118
		8.7.1 指针的加减运算	119

8.7.2 指针的赋值运算	119	10.3 重载构造函数	156
8.7.3 指针的相减运算	120	10.3.1 成员变量的赋值与初始化	156
8.7.4 指针的比较运算	121	10.3.2 成员变量的初始化与构造函数	157
8.8 指针	121	10.3.3 复制构造函数	160
8.8.1 常量指针	121	10.3.4 构造函数和 new 运算符	161
8.8.2 指向常量的指针	122	10.3.5 再谈默认构造函数	163
8.8.3 指向常量的常指针	123	10.4 析构函数和 delete 运算符	164
8.9 总结	124	10.4.1 默认析构函数	165
第 9 章 引用	125	10.4.2 调用构造函数进行类型转换	165
9.1 什么是引用	125	10.5 浅层复制构造函数	166
9.1.1 引用的地址	126	10.6 深层复制构造函数	168
9.1.2 引用就是别名常量	126	第 11 章 运算符重载	171
9.1.3 引用对象	127	11.1 运算符重载	171
9.1.4 空引用	128	11.2 在成员函数中实现自加	172
9.2 函数的参数传递	128	11.3 重载前置自加运算符	172
9.2.1 按值传递	128	11.4 创建临时对象	174
9.2.2 按址传递	129	11.5 创建无名临时对象	175
9.2.3 按别名传递	130	11.6 取消创建临时对象	176
9.2.4 让函数返回多个值	131	11.7 重载后置自加运算符	178
9.3 传递对象	134	11.8 重载加法运算符函数 operator+	179
9.3.1 按值来传递对象	134	11.9 重载赋值运算符函数 operator=	181
9.3.2 利用指针来传递对象	136	11.10 转换类型运算符	188
9.3.3 利用 const 指针来传递对象	137	11.10.1 温习调用构造函数实现的类型 转换	188
9.3.4 利用引用来传递对象	138	11.10.2 通过构造函数将变量转换为 一个对象的成员变量	189
9.3.5 到底是使用引用还是指针	140	11.10.3 通过 operator 关键字进行转换	190
9.3.6 引用和指针可以一块用	142	11.11 什么可以重载, 什么不可以重载	192
9.4 引用应注意的问题	143	第 12 章 继承	193
9.4.1 引用容易犯的错误	143	12.1 什么是继承和派生	193
9.4.2 引用一个按值返回的堆中对象	146	12.1.1 复杂的继承和派生	193
9.4.3 引用一个按别名返回的堆中对象	148	12.1.2 继承和派生如何在 C++ 中实现	194
9.4.4 在哪里创建, 就在哪里释放	149	12.1.3 继承的种类及语法	194
9.5 总结	150	12.1.4 单一继承	195
第 10 章 深入函数	152	12.2 公有型、保护型和私有型	197
10.1 函数重载	152	12.3 访问权限	198
10.1.1 普通函数的重载	152	12.4 多重继承	203
10.1.2 成员函数的重载	153		
10.2 函数的默认参数	154		

12.5 继承的构造与析构	205	14.2.2 兔子繁殖问题	252
12.6 合理利用基类构造函数	209	14.2.3 数字排序问题	254
12.7 继承和重载的两义性问题	212	14.3 数组在内存中的分布	256
12.7.1 多重继承容易产生两义性	212	14.4 输出数组名	258
12.7.2 继承中的重载	214	14.5 数组名与函数	258
12.7.3 两义性的归属问题	217	14.6 传递与接收	259
12.7.4 减少两义性产生的混淆问题	218	14.7 数组与函数	260
12.7.5 虚基类不会产生两义性	220	14.7.1 函数传参实例一——求数组所有元素的和	261
12.8 总结	221	14.7.2 函数传参实例二——用递増法查找数据	262
第 13 章 虚函数	223	14.7.3 函数传参实例三——用二分算法查找数据	263
13.1 指向子对象的父指针	223	14.7.4 函数传参实例四——判断数组是否按照顺序排列	264
13.2 虚函数	225	14.7.5 函数传参实例五——判断数组排列方式后执行不同的函数	265
13.3 拳击游戏	227	14.8 数组在对象中的传参	266
13.4 继承是否可以实现多态性	229	14.9 对象数组	267
13.5 在编译时的静态联编	231	14.10 在对象数组中初始化成员变量	268
13.6 在运行时的静态联编	231	14.11 指针数组	270
13.7 在运行时的动态联编	232	14.12 枚举常量与数组	275
13.8 在编译时的动态联编	235	14.13 多维数组	276
13.9 调用虚函数	235	14.14 多维数组的初始化	276
13.9.1 在虚函数中调用成员函数	235	14.15 字符数组	278
13.9.2 3 种调用虚函数的方式比较	237	14.16 重载数组下标操作符	287
13.10 被继承的虚函数仍然是虚函数	239	14.17 总结	289
13.11 系统是如何调用虚函数的	241	第 15 章 链表	290
13.12 在虚函数中使用成员名限定	241	15.1 声明链表结构	290
13.13 虚析构造函数	242	15.2 简单的图书链表	291
13.14 总结	244	15.2.1 图书链表	291
第 14 章 数组	245	15.2.2 类的链表	292
14.1 数组的基本用法	245	15.3 动态链表	292
14.1.1 什么是数组	245	15.3.1 动态链表的建立	293
14.1.2 数组元素	246	15.3.2 解决输入字符造成死循环的问题	295
14.1.3 数组下标越界	246	15.3.3 动态链表的显示	299
14.1.4 倒序输出	247	15.3.4 动态链表的删除	299
14.1.5 将数组的下标定义为常量	248		
14.1.6 手动操作数组元素	248		
14.1.7 数组的初始化	249		
14.2 数组的用途	250		
14.2.1 求平均考试成绩	250		

15.3.5 动态链表的插入	301
15.3.6 链表统计	302
15.3.7 使用链表	303
15.4 完整的动态链表清单	306
15.5 链表使用案例——走迷宫	310
15.5.1 创建 Windows 应用程序项目	310
15.5.2 创建窗口	311
15.5.3 加载图片	313
15.5.4 句柄是什么	315
15.5.5 显示图片	315
15.5.6 动画	319
15.5.7 键盘控制人物移动	323
15.5.8 迷宫墙壁	327
15.5.9 走迷宫	333
15.5.10 用链表记录行走路线	336
第 16 章 多态性	341
16.1 为什么要使用多重继承	341
16.2 在派生类中增加函数	343
16.3 使用多重继承	345
16.4 多重继承的初始化	347
16.5 多个子基类共享一个父基类	349
16.6 虚基类	352
16.7 慎用多重继承	356
16.8 空的虚函数	356
16.9 模拟抽象类	358
16.10 纯虚函数与抽象类	360
16.11 复杂的抽象结构	363
16.12 总结	366
第 17 章 类的特殊成员	367
17.1 静态成员变量	367
17.2 私有静态成员变量	370
17.3 静态成员函数	370
17.4 静态成员的使用	372
17.5 函数指针	374
17.6 函数指针数组	378
17.7 函数指针作为函数的参数	379
17.8 利用 typedef 简化函数指针	381
17.9 类的函数指针	383
17.10 成员函数指针数组	385
第 18 章 字符串	388
18.1 char 型字符串	388
18.2 string 型字符串	391
18.2.1 string 型字符串的赋值	396
18.2.2 string 型字符串的合并	398
18.2.3 string 型字符串的部分合并	400
18.2.4 string 型字符串的替换	401
18.2.5 string 型字符串的复制	404
18.2.6 string 型字符串的插入	405
18.2.7 string 型字符串的删除	406
18.2.8 string 型字符串的查找	407
18.2.9 string 型字符串的比较	408
18.2.10 判断 string 型字符串是否为空	409
18.3 字符串的使用	410
18.3.1 交换两个字符串的内容	410
18.3.2 将 string 型字符串转为 char 型字符串	411
18.3.3 char 型字符串与函数	411
18.3.4 函数如何返回字符串	413
18.4 结构体	414
18.4.1 结构体的赋值	418
18.4.2 结构体与函数	418
18.4.3 结构体与 string	421
18.5 string 数组与函数	422
18.6 流的使用	423
18.6.1 重载输出运算符<<	424
18.6.2 用友元的方式重载输出运算符	426
18.6.3 重载自加运算符的执行次序	426
18.6.4 重载输入运算符>>	428
18.7 编写一个 String 类	429
18.7.1 创建 String 类	429
18.7.2 创建可自动调节大小的 String 类字符串对象	430
18.7.3 限制数组越界	432
18.7.4 用复制构造函数实现字符串的初始化功能	434