

PEARSON

C和C++实务精选
品味岁月积淀，读享技术菁华

C++ Primer Plus

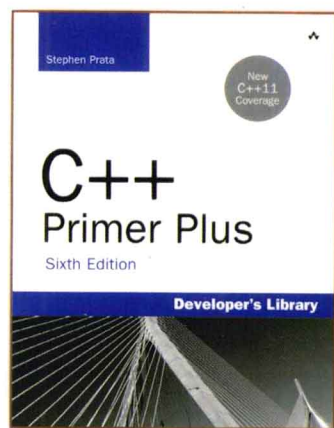
中文版
累计销量超
80 000册

(第6版) 中文版

[美] Stephen Prata 著 张海龙 袁国忠 译

C++ Primer Plus Sixth Edition

- 经久不衰的C++畅销经典教程
- 涵盖C++11新标准



NLIC2970801660

人民邮电出版社
POSTS & TELECOM PRESS

PEARSON

C++ Primer Plus

(第6版) 中文版

[美] Stephen Prata 著 张海龙 袁国忠 译

C++ Primer Plus Sixth Edition



人民邮电出版社
北京

图书在版编目 (C I P) 数据

C++ Primer Plus中文版 : 第6版 / (美) 普拉达
(Prata, S.) 著 ; 张海龙, 袁国忠译. -- 北京 : 人民邮
电出版社, 2012. 7
ISBN 978-7-115-27946-0

I. ①C… II. ①普… ②张… ③袁… III. ①
C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第065092号

版 权 声 明

Authorized translation from the English language edition, entitled C++ Primer Plus (sixth edition), 9780321776402 by Stephen Prata, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2011 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education Inc. CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2012.

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签。无标签者不得销售。

C++ Primer Plus (第 6 版) 中文版

-
- ◆ 著 [美] Stephen Prata
译 张海龙 袁国忠
责任编辑 傅道坤
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京新华印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 59.5
字数: 1 946 千字 2012 年 7 月第 1 版
印数: 1-10 000 册 2012 年 7 月北京第 1 次印刷
著作权合同登记号 图字: 01-2012-0244 号

ISBN 978-7-115-27946-0

定价: 99.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

内 容 提 要

C++是在 C 语言基础上开发的一种集面向对象编程、泛型编程和过程化编程于一体的编程语言，是 C 语言的超集。本书是根据 2003 年的 ISO/ANSI C++标准编写的，通过大量短小精悍的程序详细而全面地阐述了 C++的基本概念和技术，并专辟一章介绍了 C++11 新增的功能。

全书分 18 章和 10 个附录。分别介绍了 C++程序的运行方式、基本数据类型、复合数据类型、循环和关系表达式、分支语句和逻辑运算符、函数重载和函数模板、内存模型和名称空间、类的设计和使用、多态、虚函数、动态内存分配、继承、代码重用、友元、异常处理技术、string 类和标准模板库、输入/输出、C++11 新增功能等内容。

本书针对 C++初学者，书中从 C 语言基础知识开始介绍，然后在此基础上详细阐述 C++新增的特性，因此不要求读者有 C 语言方面的背景知识。本书可作为高等院校教授 C++课程的教材，也可供初学者自学 C++时使用。

作者简介

Stephen Prata 在美国加州肯特菲尔得的马林学院教授天文、物理和计算机科学。他毕业于加州理工学院，在美国加州大学伯克利分校获得博士学位。他单独或与他人合作编写的编程图书有十多本，其中《New C Primer Plus》获得了计算机出版联合会 1990 年度最佳“**How-to**”计算机图书奖，《C++ Primer Plus》获得了计算机出版联合会 1991 年度最佳“**How-to**”计算机图书奖提名。

前 言

学习 C++ 是一次探索之旅，因为这种语言容纳了好几种编程范式，其中包括面向对象编程、泛型编程和传统的过程化编程。本书第 5 版是基于 ISO C++ 标准编写的，该标准的官方名称为 C++99 和 C++03 (C++99/C++03)，其中 2003 标准主要是对 1999 标准的技术修正，并没有添加任何新功能。C++ 在不断发展，编写本书时，新标准获得了 C++ 国际标准委员会的批准。在制定期间，该标准名为 C++0x，但现已改名为 C++11。大多数编译器都能很好地支持 C++99/03，而本书的大多数示例都遵守该标准。有些实现中已显现了新标准的很多功能，而本书也对这些新功能进行了探索。

本书在介绍 C++ 特性的同时，讨论了基本 C 语言，使两者成为有机的整体。书中介绍了 C++ 的基本概念，并通过短小精悍的程序来阐明，这些程序都很容易复制和试验。书中还介绍了输入和输出，如何让程序执行重复性任务，如何让程序做出选择，处理数据的多种方式，以及如何使用函数等内容。另外，本书还讲述了 C++ 在 C 语言的基础上新增的很多特性，包括：

- 类和对象；
- 继承；
- 多态、虚函数和 RTTI（运行阶段类型识别）；
- 函数重载；
- 引用变量；
- 泛型（独立于类型的）编程，这种技术是由模板和标准模板库（STL）提供的；
- 处理错误条件的异常机制；
- 管理函数、类和变量名的名称空间。

初级教程方法

大约 20 年前，《C Primer Plus》开创了优良的初级教程传统，本书建立在这样的基础之上，吸收了其中很多成功的理念。

- 初级教程应当是友好的、便于使用的指南。
- 初级教程不要求您已经熟悉相关的编程概念。
- 初级教程强调的是动手学习，通过简短、容易输入的示例阐述一两个概念。
- 初级教程用示意图来解释概念。
- 初级教程提供问题和练习来检验您对知识的理解，从而适于自学或课堂教学。

基于上述理念，本书帮助您理解这种用途广泛的语言，并学习如何使用它。

- 对何时使用某些特性，例如何时使用公共继承来建立 is-a 关系，提供了概念方面的指导。
- 阐释了常用的 C++ 编程理念和技术。
- 提供了大量的附注，如提示、警告、注意等。

本书的作者和编辑尽最大的努力使本书简单、明了、生动有趣。我们的目标是，您阅读本书后，能够编写出可靠、高效的程序，并且觉得这是一种享受。

示例代码

本书包含大量的示例代码，其中大部分是完整的程序。和前一版一样，本书介绍的是通用 C++，因此适用于任何计算机、操作系统和编译器。书中的示例在 Windows 7 系统、Macintosh OS X 系统和 Linux 系统上进行了测试。

使用了 C++11 功能的程序要求编译器支持这些功能，但其他程序可在遵循 C++ 99/03 的任何系统上运行。

书中完整程序的源代码可从配套网站下载，详情请参阅封底的链接信息。

本书内容

本书分为 18 章和 10 个附录。

- 第 1 章 预备知识：本章介绍 Bjarne Stroustrup 如何在 C 语言的基础上添加对面向对象编程的支持，来创造 C++ 编程语言。讨论面向过程语言（如 C 语言）与面向对象语言（如 C++）之间的区别。您将了解 ANSI/ISO 在制定 C++ 标准方面所做的工作。本章还讨论了创建 C++ 程序的技巧，介绍了当前几种 C++ 编译器使用的方法。最后，本章介绍了本书的一些约定。

- 第 2 章 开始学习 C++：本章介绍创建简单 C++ 程序的步骤。您可以学习到 `main()` 函数扮演的角色以及 C++ 程序使用的一些语句。您将使用预定义的 `cout` 和 `cin` 对象来实现程序输出和输入，学习如何创建和使用变量。最后，本章还将介绍函数——C++ 的编程模块。

- 第 3 章 处理数据：C++ 提供了内置类型来存储两种数据：整数（没有小数的数字）和浮点数（带小数的数字）。为满足程序员的各种需求，C++ 为每一种数据都提供了几个类型。本章将要讨论这些类型，包括创建变量和编写各种类型的常量。另外，还将讨论 C++ 是如何处理不同类型之间的隐式和显式转换的。

- 第 4 章 复合类型：C++ 让程序员能够使用基本的内置类型来创建更复杂的类型。最高级的形式是类，这将在第 9 章~第 13 章讨论。本章讨论其他形式，包括数组（存储多个同类型的值）、结构（存储多个不同类型的值）、指针（标识内存位置）。您还将学习如何创建和存储文本字符串及如何使用 C-风格字符串数组和 C++ `string` 类来处理文本输入和输出。最后，还将学习 C++ 处理内存分配的一些方法，其中包括用于显式地管理内存的 `new` 和 `delete` 运算符。

- 第 5 章 循环和关系表达式：程序经常需要执行重复性操作，为此 C++ 提供了 3 种循环结构：`for` 循环、`while` 循环和 `do while` 循环。这些循环必须知道何时终止，C++ 的关系运算符使程序员能够创建测试来引导循环。本章还将介绍如何创建逐字符地读取和处理输入的循环。最后，您将学习如何创建二维数组以及如何使用嵌套循环来处理它们。

- 第 6 章 分支语句和逻辑运算符：如果程序可以根据实际情况调整执行，我们就说程序能够智能地行动。在本章，您将了解到如何使用 `if`、`if else` 和 `switch` 语句及条件运算符来控制程序流程，学习如何使用逻辑运算符来表达决策测试。另外，本章还将介绍确定字符关系（如测试字符是数字还是非打印字符）的函数库 `cctype`。最后，还将简要地介绍文件输入/输出。

- 第 7 章 函数——C++ 的编程模块：函数是 C++ 的基本编程部件。本章重点介绍 C++ 函数与 C 函数共同的特性。具体地说，您将复习函数定义的通用格式，了解函数原型是如何提高程序可靠性的。同时，还将学习如何编写函数来处理数组、字符串和结构。还要学习有关递归的知识（即函数在什么情况下调用自身）以及如何用它来实现分而治之的策略。最后将介绍函数指针，它使程序员能够通过函数参数来命令函数使用另一个函数。

- 第 8 章 函数探幽：本章将探索 C++ 中函数新增的特性。您将学习内联函数，它可以提高程序的执行速度，但会增加程序的长度；还将使用引用变量，它们提供了另一种将信息传递给函数的方式。默认参数使函数能够自动为函数调用中省略的函数参数提供值。函数重载使程序员能够创建多个参数列表不同的同名函数。类设计中经常使用这些特性。另外，您还将学习函数模板，它们使程序员能够指定相关函数族的设计。

- 第 9 章 内存模型和名称空间：本章讨论如何创建多文件程序，介绍分配内存的各种方式、管理内存的各种方式以及作用域、链接、名称空间，这些内容决定了变量在程序的哪些部分是可见的。

- 第 10 章 对象和类：类是用户定义的类型，对象（如变量）是类的实例。本章介绍面向对象编程和类设计。对象声明描述的是存储在对象中的信息以及可对对象执行的操作（类方法）。对象的某些组成部分对于外界来说是可见的（公有部分），而某些部分却是隐藏的（私有部分）。特殊的类方法（构造函数和析构函数）在对象创建和释放时发挥作用。在本章中，您将学习所有这些内容以及其他类知识，了解如何使用类来实现 ADT，如栈。

- 第 11 章 使用类：在本章中，您将深入了解类。首先了解运算符重载，它使程序员能够定义与类

对象一起使用的运算符，如+。还将学习友元函数，这些函数可以访问外部世界不可访问的类数据。同时还将了解一些构造函数和重载运算符成员函数是如何被用来管理类类型转换的。

- 第 12 章 类和动态内存分配：一般来说，让类成员指向动态分配的内存很有用。如果程序员在类构造函数中使用 `new` 来分配动态内存，就有责任提供适当的析构函数，定义显式拷贝构造函数和显式赋值运算符。本章介绍了在程序员没有提供显式定义时，将如何隐式地生成成员函数以及这些成员函数的行为。您还将通过使用对象指针，了解队列模拟问题，扩充类方面的知识。

- 第 13 章 类继承：在面向对象编程中，继承是功能最强大的特性之一，通过继承，派生类可以继承基类的特性，可重用基类代码。本章讨论公有继承，这种继承模拟了 `is-a` 关系，即派生对象是基对象的特例。例如，物理学家是科学家的特例。有些继承关系是多态的，这意味着相同的方法名称可能导致依赖于对象类型的行为。要实现这种行为，需要使用一种新的成员函数——虚函数。有时，使用抽象基类是实现继承关系的最佳方式。本章讨论了这些问题，说明了公有继承在什么情况下合适，在什么情况下不合适。

- 第 14 章 C++中的代码重用：公有继承只是代码重用的方式之一。本章将介绍其他几种方式。如果一个类包含了另一个类的对象，则称为包含。包含可以用来模拟 `has-a` 关系，其中一个类包含另一个类的对象。例如，汽车有马达。也可以使用私有继承和保护继承来模拟这种关系。本章说明了各种方法之间的区别。同时，您还将学习类模板，它让程序员能够使用泛型定义类，然后使用模板根据具体类型创建特定的类。例如，栈模板使程序员能够创建整数栈或字符串栈。最后，本章还将介绍多重公有继承，使用这种方式，一个类可以从多个类派生而来。

- 第 15 章 友元、异常和其他：本章扩展了对友元的讨论，讨论了友元类和友元成员函数。然后从异常开始介绍了 C++的几项新特性。异常为处理程序异常提供了一种机制，如函数参数值不正确或内存耗尽等。您还将学习 `RTTI`，这种机制用来确定对象类型。最后，本章还将介绍一种更安全的方法来替代不受限制的强制类型转换。

- 第 16 章 `string` 类和标准模板库：本章讨论 C++语言中新增的一些类库。对于传统的 C-风格字符串来说，`string` 类是一种方便且功能强大的替代方式。`auto_ptr` 类帮助管理动态分配的内存。STL 提供了几种类容器（包括数组、队列、链表、集合和映射）的模板表示。它还提供了高效的泛型算法库，这些算法可用于 STL 容器，也可用于常规数组。模板类 `valarray` 为数值数组提供了支持。

- 第 17 章 输入、输出和文件：本章复习 C++ I/O，并讨论如何格式化输出。您将要学习如何使用类方法来确定输入或输出流的状态，了解输入类型是否匹配或是否检测到了文件尾。C++使用继承来派生用于管理文件输入和输出的类。您将学习如何打开文件，以进行输入和输出，如何在文件中追加数据，如何使用二进制文件，如何获得对文件的随机访问权。最后，还将学习如何使用标准的 I/O 方法来读取和写入字符串。

- 第 18 章 探讨 C++新标准：本章首先复习之前介绍过的几项 C++11 新功能，包括新类型、统一的初始化语法、自动类型推断、新的智能指针以及作用域内枚举。然后，讨论新增的右值引用类型以及如何使用它来实现移动语义。接下来，介绍了新增的类功能、`lambda` 表达式和可变参数模板。最后，概述了众多其他的新功能。

- 附录 A 计数系统：本附录讨论八进制数、十六进制数和二进制数。
- 附录 B C++保留字：本附录列出了 C++关键字。
- 附录 C ASCII 字符集：本附录列出了 ASCII 字符集及其十进制、八进制、十六进制和二进制表示。
- 附录 D 运算符优先级：本附录按优先级从高到低的顺序列出了 C++的运算符。
- 附录 E 其他运算符：本附录总结了正文中没有介绍的其他 C++运算符，如按位运算符等。
- 附录 F 模板类 `string`：本附录总结了 `string` 类方法和函数。
- 附录 G 标准模板库方法和函数：本附录总结了 STL 容器方法和通用的 STL 算法函数。
- 附录 H 精选读物和网上资源：本附录列出一些参考书，帮助您深入了解 C++。
- 附录 I 转换为 ISO 标准 C++：本附录提供了从 C 和老式 C++实现到标准 C++的转换指南。
- 附录 J 复习题答案：本附录提供各章结尾的复习题的答案。

对教师的提示

本书宗旨之一是，提供一本既可用于自学又可用于教学的书籍。下面是本书在支持教学方面的一些特征。

- 本书介绍的是通用 C++，不依赖于特定实现。
- 本书内容跟踪了 ISO/ANSI C++ 标准委员会的工作，并讨论了模板、STL、string 类、异常、RTTI 和名称空间。
- 本书不要求学生了解 C 语言，但如果有一定的编程经验则更好。
- 本书内容经过了精心安排，前几章可以作为对 C 预备知识的复习一带而过。
- 各章都有复习题和编程练习。附录 J 提供了复习题的答案。
- 本书介绍的一些主题很适于计算机科学课程，包括抽象数据类型 (ADT)、栈、队列、简单链表、模拟、泛型编程以及使用递归来实现分而治之的策略。
- 各章都非常简短，用一周甚至更短的时间就可以学完。
- 本书讨论了何时使用具体的特性以及如何使用它们。例如，把 is-a 关系的公有继承同组合、has-a 关系的私有继承联系起来，讨论了何时应使用虚函数以及何时不应使用。

本书约定

为区别不同类型的文本，我们使用了一些印刷上的约定。

- 代码行、命令、语句、变量、文件名和程序输出使用 **courier new** 字体：

```
#include <iostream>
int main()
{
    using namespace std;
    cout << "What's up, Doc!\n";
    return 0;
}
```

- 用户需要输入的程序输入用**粗体**表示：

```
Please enter your name:
Plato
```

- 语法描述中的占位符用**斜体**表示。您应使用实际的文件名、参数等替换占位符。
- 新术语用**斜体**表示。

旁注：提供更深入的讨论和额外的背景知识，帮助阐明主题。

提示：提供特定编程情形下很有帮助简单指南。

警告：指出潜在的陷阱。

注意：提供不属于其他类别的各种说明。

开发本书编程示例时使用的系统

本书的 C++11 示例是使用 Microsoft Visual C++ 2010 和带 Gnu g++ 4.5.0 的 Cygwin 开发的，它们都运行在 64 位的 Windows 7 系统上。其他示例在这些系统上进行了测试，还在 OS X 10.6.8 系统和 Ubuntu Linux 系统上分别使用 g++ 4.21 和 g++ 4.4.1 进行了测试。大多数非 C++11 示例最初都是在 Windows XP Professional 系统上使用 Microsoft Visual C++ 2003 和 Metrowerks CodeWarrior Development Studio 9 开发的，并在该系统上使用 Borland C++ 5.5 命令行编译器和 GNU gpp 3.3.3 进行了测试；其次，在运行 SuSE 9.0 Linux 的系统上使用 Comeau 4.3.3 和 GNU g++3.3.1 进行了测试；最后，在运行 OS 10.3 的 Macintosh G4 上使用 Metrowerks Development Studio 9 进行了测试。

C++ 为程序员提供了丰富多彩的内容。祝您学习愉快！

目 录

| | |
|--------------------------------------|---|
| 第 1 章 预备知识1 | 2.7 编程练习.....35 |
| 1.1 C++简介.....1 | 第 3 章 处理数据37 |
| 1.2 C++简史.....2 | 3.1 简单变量.....37 |
| 1.2.1 C 语言.....2 | 3.1.1 变量名.....38 |
| 1.2.2 C 语言编程原理.....2 | 3.1.2 整型.....39 |
| 1.2.3 面向对象编程.....3 | 3.1.3 整型 short、int、long 和 long long.....39 |
| 1.2.4 C++和泛型编程.....4 | 3.1.4 无符号类型.....43 |
| 1.2.5 C++的起源.....4 | 3.1.5 选择整型类型.....45 |
| 1.3 可移植性和标准.....5 | 3.1.6 整型字面值.....45 |
| 1.3.1 C++的发展.....6 | 3.1.7 C++如何确定常量的类型.....47 |
| 1.3.2 本书遵循的 C++标准.....6 | 3.1.8 char 类型：字符和小整数.....47 |
| 1.4 程序创建的技巧.....6 | 3.1.9 bool 类型.....53 |
| 1.4.1 创建源代码文件.....7 | 3.2 const 限定符.....54 |
| 1.4.2 编译和链接.....8 | 3.3 浮点数.....55 |
| 1.5 总结.....10 | 3.3.1 书写浮点数.....55 |
| 第 2 章 开始学习 C++12 | 3.3.2 浮点类型.....56 |
| 2.1 进入 C++.....12 | 3.3.3 浮点常量.....58 |
| 2.1.1 main()函数.....13 | 3.3.4 浮点数的优缺点.....58 |
| 2.1.2 C++注释.....15 | 3.4 C++算术运算符.....59 |
| 2.1.3 C++预处理器和 iostream 文件.....16 | 3.4.1 运算符优先级和结合性.....60 |
| 2.1.4 头文件名.....16 | 3.4.2 除法分支.....60 |
| 2.1.5 名称空间.....17 | 3.4.3 求模运算符.....61 |
| 2.1.6 使用 cout 进行 C++输出.....18 | 3.4.4 类型转换.....62 |
| 2.1.7 C++源代码的格式化.....19 | 3.4.5 C++11 中的 auto 声明.....66 |
| 2.2 C++语句.....21 | 3.5 总结.....67 |
| 2.2.1 声明语句和变量.....21 | 3.6 复习题.....67 |
| 2.2.2 赋值语句.....22 | 3.7 编程练习.....68 |
| 2.2.3 cout 的新花样.....23 | 第 4 章 复合类型70 |
| 2.3 其他 C++语句.....23 | 4.1 数组.....70 |
| 2.3.1 使用 cin.....24 | 4.1.1 程序说明.....72 |
| 2.3.2 使用 cout 进行拼接.....24 | 4.1.2 数组的初始化规则.....73 |
| 2.3.3 类简介.....25 | 4.1.3 C++11 数组初始化方法.....73 |
| 2.4 函数.....26 | 4.2 字符串.....74 |
| 2.4.1 使用有返回值的函数.....26 | 4.2.1 拼接字符串常量.....75 |
| 2.4.2 函数变体.....29 | 4.2.2 在数组中使用字符串.....76 |
| 2.4.3 用户定义的函数.....29 | 4.2.3 字符串输入.....77 |
| 2.4.4 用户定义的有返回值的函数.....32 | 4.2.4 每次读取一行字符串输入.....78 |
| 2.4.5 在多函数程序中使用 using 编译指令.....33 | 4.2.5 混合输入字符串和数字.....81 |
| 2.5 总结.....34 | 4.3 string 类简介.....82 |
| 2.6 复习题.....35 | 4.3.1 C++11 字符串初始化.....83 |

| | | | | | |
|--------------|-----------------------------|------------|--------------|-----------------------|------------|
| 4.3.2 | 赋值、拼接和附加 | 83 | 5.1.6 | 副作用和顺序点 | 134 |
| 4.3.3 | string 类的其他操作 | 84 | 5.1.7 | 前缀格式和后缀格式 | 135 |
| 4.3.4 | string 类 I/O | 86 | 5.1.8 | 递增/递减运算符和指针 | 135 |
| 4.3.5 | 其他形式的字符串字面值 | 87 | 5.1.9 | 组合赋值运算符 | 136 |
| 4.4 | 结构简介 | 88 | 5.1.10 | 复合语句(语句块) | 136 |
| 4.4.1 | 在程序中使用结构 | 89 | 5.1.11 | 其他语法技巧—— 逗号运算符 | 138 |
| 4.4.2 | C++11 结构初始化 | 91 | 5.1.12 | 关系表达式 | 140 |
| 4.4.3 | 结构可以将 string 类 作为成员吗 | 91 | 5.1.13 | 赋值、比较和可能犯的错误 | 141 |
| 4.4.4 | 其他结构属性 | 91 | 5.1.14 | C-风格字符串的比较 | 142 |
| 4.4.5 | 结构数组 | 93 | 5.1.15 | 比较 string 类字符串 | 144 |
| 4.4.6 | 结构中的位字段 | 94 | 5.2 | while 循环 | 145 |
| 4.5 | 共用体 | 94 | 5.2.1 | for 与 while | 147 |
| 4.6 | 枚举 | 95 | 5.2.2 | 等待一段时间: 编写延时循环 | 148 |
| 4.6.1 | 设置枚举量的值 | 97 | 5.3 | do while 循环 | 150 |
| 4.6.2 | 枚举的取值范围 | 97 | 5.4 | 基于范围的 for 循环(C++11) | 152 |
| 4.7 | 指针和自由存储空间 | 97 | 5.5 | 循环和文本输入 | 152 |
| 4.7.1 | 声明和初始化指针 | 100 | 5.5.1 | 使用原始的 cin 进行输入 | 152 |
| 4.7.2 | 指针的危险 | 101 | 5.5.2 | 使用 cin.get(char) 进行补救 | 153 |
| 4.7.3 | 指针和数字 | 102 | 5.5.3 | 使用哪一个 cin.get() | 154 |
| 4.7.4 | 使用 new 来分配内存 | 102 | 5.5.4 | 文件尾条件 | 155 |
| 4.7.5 | 使用 delete 释放内存 | 104 | 5.5.5 | 另一个 cin.get() 版本 | 157 |
| 4.7.6 | 使用 new 来创建动态数组 | 104 | 5.6 | 嵌套循环和二维数组 | 159 |
| 4.8 | 指针、数组和指针算术 | 106 | 5.6.1 | 初始化二维数组 | 160 |
| 4.8.1 | 程序说明 | 107 | 5.6.2 | 使用二维数组 | 160 |
| 4.8.2 | 指针小结 | 109 | 5.7 | 总结 | 162 |
| 4.8.3 | 指针和字符串 | 111 | 5.8 | 复习题 | 163 |
| 4.8.4 | 使用 new 创建动态结构 | 114 | 5.9 | 编程练习 | 163 |
| 4.8.5 | 自动存储、静态存储和 动态存储 | 117 | 第 6 章 | 分支语句和逻辑运算符 | 165 |
| 4.9 | 类型组合 | 118 | 6.1 | if 语句 | 165 |
| 4.10 | 数组的替代品 | 120 | 6.1.1 | if else 语句 | 167 |
| 4.10.1 | 模板类 vector | 120 | 6.1.2 | 格式化 if else 语句 | 168 |
| 4.10.2 | 模板类 array(C++11) | 120 | 6.1.3 | if else if else 结构 | 169 |
| 4.10.3 | 比较数组、vector 对象和 array 对象 | 120 | 6.2 | 逻辑表达式 | 170 |
| 4.11 | 总结 | 122 | 6.2.1 | 逻辑 OR 运算符: | 171 |
| 4.12 | 复习题 | 123 | 6.2.2 | 逻辑 AND 运算符: && | 172 |
| 4.13 | 编程练习 | 123 | 6.2.3 | 用 && 来设置取值范围 | 174 |
| 第 5 章 | 循环和关系表达式 | 125 | 6.2.4 | 逻辑 NOT 运算符: ! | 175 |
| 5.1 | for 循环 | 125 | 6.2.5 | 逻辑运算符细节 | 176 |
| 5.1.1 | for 循环的组成部分 | 126 | 6.2.6 | 其他表示方式 | 177 |
| 5.1.2 | 回到 for 循环 | 131 | 6.3 | 字符函数库 ctype | 177 |
| 5.1.3 | 修改步长 | 132 | 6.4 | ? : 运算符 | 179 |
| 5.1.4 | 使用 for 循环访问字符串 | 133 | 6.5 | switch 语句 | 180 |
| 5.1.5 | 递增运算符(++) 和递减运算符(--) | 133 | 6.5.1 | 将枚举量用作标签 | 183 |
| | | | 6.5.2 | switch 和 if else | 184 |
| | | | 6.6 | break 和 continue 语句 | 185 |
| | | | 6.7 | 读取数字的循环 | 187 |

| | | | |
|---------------------------------|------------|------------------------------|------------|
| 6.8 简单文件输入/输出 | 190 | 8.2.2 将引用用作函数参数 | 258 |
| 6.8.1 文本 I/O 和文本文件 | 190 | 8.2.3 引用的属性和特别之处 | 260 |
| 6.8.2 写入到文本文件中 | 191 | 8.2.4 将引用用于结构 | 263 |
| 6.8.3 读取文本文件 | 194 | 8.2.5 将引用用于类对象 | 268 |
| 6.9 总结 | 197 | 8.2.6 对象、继承和引用 | 271 |
| 6.10 复习题 | 198 | 8.2.7 何时使用引用参数 | 274 |
| 6.11 编程练习 | 199 | 8.3 默认参数 | 274 |
| 第 7 章 函数——C++的编程模块 | 202 | 8.4 函数重载 | 276 |
| 7.1 复习函数的基本知识 | 202 | 8.4.1 重载示例 | 278 |
| 7.1.1 定义函数 | 203 | 8.4.2 何时使用函数重载 | 280 |
| 7.1.2 函数原型和函数调用 | 205 | 8.5 函数模板 | 281 |
| 7.2 函数参数和按值传递 | 207 | 8.5.1 重载的模板 | 283 |
| 7.2.1 多个参数 | 208 | 8.5.2 模板的局限性 | 285 |
| 7.2.2 另外一个接受两个参数的函数 | 210 | 8.5.3 显式具体化 | 285 |
| 7.3 函数和数组 | 212 | 8.5.4 实例化和具体化 | 288 |
| 7.3.1 函数如何使用指针来处理数组 | 213 | 8.5.5 编译器选择使用哪个函数版本 | 289 |
| 7.3.2 将数组作为参数意味着什么 | 213 | 8.5.6 模板函数的发展 | 295 |
| 7.3.3 更多数组函数示例 | 215 | 8.6 总结 | 297 |
| 7.3.4 使用数组区间的函数 | 220 | 8.7 复习题 | 297 |
| 7.3.5 指针和 const | 221 | 8.8 编程练习 | 298 |
| 7.4 函数和二维数组 | 224 | 第 9 章 内存模型和名称空间 | 300 |
| 7.5 函数和 C-风格字符串 | 225 | 9.1 单独编译 | 300 |
| 7.5.1 将 C-风格字符串作为参数的函数 | 225 | 9.2 存储持续性、作用域和链接性 | 304 |
| 7.5.2 返回 C-风格字符串的函数 | 227 | 9.2.1 作用域和链接 | 305 |
| 7.6 函数和结构 | 228 | 9.2.2 自动存储持续性 | 305 |
| 7.6.1 传递和返回结构 | 228 | 9.2.3 静态持续变量 | 309 |
| 7.6.2 另一个处理结构的函数示例 | 230 | 9.2.4 静态持续性、外部链接性 | 310 |
| 7.6.3 传递结构的地址 | 234 | 9.2.5 静态持续性、内部链接性 | 313 |
| 7.7 函数和 string 对象 | 235 | 9.2.6 静态存储持续性、无链接性 | 315 |
| 7.8 函数与 array 对象 | 236 | 9.2.7 说明符和限定符 | 317 |
| 7.9 递归 | 238 | 9.2.8 函数和链接性 | 318 |
| 7.9.1 包含一个递归调用的递归 | 239 | 9.2.9 语言链接性 | 319 |
| 7.9.2 包含多个递归调用的递归 | 240 | 9.2.10 存储方案和动态分配 | 319 |
| 7.10 函数指针 | 241 | 9.3 名称空间 | 324 |
| 7.10.1 函数指针的基础知识 | 241 | 9.3.1 传统的 C++名称空间 | 324 |
| 7.10.2 函数指针示例 | 243 | 9.3.2 新的名称空间特性 | 325 |
| 7.10.3 深入探讨函数指针 | 244 | 9.3.3 名称空间示例 | 331 |
| 7.10.4 使用 typedef 进行简化 | 248 | 9.3.4 名称空间及其前途 | 334 |
| 7.11 总结 | 248 | 9.4 总结 | 335 |
| 7.12 复习题 | 249 | 9.5 复习题 | 335 |
| 7.13 编程练习 | 250 | 9.6 编程练习 | 338 |
| 第 8 章 函数探幽 | 253 | 第 10 章 对象和类 | 340 |
| 8.1 C++内联函数 | 253 | 10.1 过程性编程和面向对象编程 | 340 |
| 8.2 引用变量 | 255 | 10.2 抽象和类 | 341 |
| 8.2.1 创建引用变量 | 256 | 10.2.1 类型是什么 | 341 |
| | | 10.2.2 C++中的类 | 342 |

| | | | | | |
|---------------|----------------------|------------|---------------|-----------------------------|------------|
| 10.2.3 | 实现类成员函数 | 345 | 12.1.2 | 特殊成员函数 | 432 |
| 10.2.4 | 使用类 | 349 | 12.1.3 | 回到 Stringbad: 复制构造函数的哪里出了问题 | 434 |
| 10.2.5 | 修改实现 | 350 | 12.1.4 | Stringbad 的其他问题: 赋值运算符 | 436 |
| 10.2.6 | 小结 | 351 | 12.2 | 改进后的新 String 类 | 437 |
| 10.3 | 类的构造函数和析构函数 | 352 | 12.2.1 | 修订后的默认构造函数 | 438 |
| 10.3.1 | 声明和定义构造函数 | 353 | 12.2.2 | 比较成员函数 | 439 |
| 10.3.2 | 使用构造函数 | 354 | 12.2.3 | 使用中括号表示法访问字符 | 439 |
| 10.3.3 | 默认构造函数 | 354 | 12.2.4 | 静态类成员函数 | 441 |
| 10.3.4 | 析构函数 | 355 | 12.2.5 | 进一步重载赋值运算符 | 441 |
| 10.3.5 | 改进 Stock 类 | 356 | 12.3 | 在构造函数中使用 new 时应注意的事项 | 446 |
| 10.3.6 | 构造函数和析构函数小结 | 362 | 12.3.1 | 应该和不应该 | 447 |
| 10.4 | this 指针 | 363 | 12.3.2 | 包含类成员的类的逐成员复制 | 448 |
| 10.5 | 对象数组 | 368 | 12.4 | 有关返回对象的说明 | 449 |
| 10.6 | 类作用域 | 370 | 12.4.1 | 返回指向 const 对象的引用 | 449 |
| 10.6.1 | 作用域为类的常量 | 371 | 12.4.2 | 返回指向非 const 对象的引用 | 449 |
| 10.6.2 | 作用域内枚举 (C++11) | 372 | 12.4.3 | 返回对象 | 450 |
| 10.7 | 抽象数据类型 | 373 | 12.4.4 | 返回 const 对象 | 450 |
| 10.8 | 总结 | 376 | 12.5 | 使用指向对象的指针 | 451 |
| 10.9 | 复习题 | 377 | 12.5.1 | 再谈 new 和 delete | 453 |
| 10.10 | 编程练习 | 377 | 12.5.2 | 指针和对象小结 | 454 |
| | | | 12.5.3 | 再谈定位 new 运算符 | 456 |
| 第 11 章 | 使用类 | 380 | 12.6 | 复习各种技术 | 459 |
| 11.1 | 运算符重载 | 380 | 12.6.1 | 重载 << 运算符 | 459 |
| 11.2 | 计算时间: 一个运算符重载示例 | 381 | 12.6.2 | 转换函数 | 460 |
| 11.2.1 | 添加加法运算符 | 384 | 12.6.3 | 其构造函数使用 new 的类 | 460 |
| 11.2.2 | 重载限制 | 387 | 12.7 | 队列模拟 | 460 |
| 11.2.3 | 其他重载运算符 | 388 | 12.7.1 | 队列类 | 461 |
| 11.3 | 友元 | 390 | 12.7.2 | Customer 类 | 468 |
| 11.3.1 | 创建友元 | 391 | 12.7.3 | ATM 模拟 | 471 |
| 11.3.2 | 常用的友元: 重载 << 运算符 | 392 | 12.8 | 总结 | 475 |
| 11.4 | 重载运算符: 作为成员函数还是非成员函数 | 397 | 12.9 | 复习题 | 476 |
| 11.5 | 再谈重载: 一个矢量类 | 398 | 12.10 | 编程练习 | 477 |
| 11.5.1 | 使用状态成员 | 404 | 第 13 章 | 类继承 | 480 |
| 11.5.2 | 为 Vector 类重载算术运算符 | 406 | 13.1 | 一个简单的基类 | 481 |
| 11.5.3 | 对实现的说明 | 408 | 13.1.1 | 派生一个类 | 482 |
| 11.5.4 | 使用 Vector 类来模拟随机漫步 | 408 | 13.1.2 | 构造函数: 访问权限的考虑 | 484 |
| 11.6 | 类的自动转换和强制类型转换 | 411 | 13.1.3 | 使用派生类 | 486 |
| 11.6.1 | 转换函数 | 415 | 13.1.4 | 派生类和基类之间的特殊关系 | 488 |
| 11.6.2 | 转换函数和友元函数 | 419 | 13.2 | 继承: is-a 关系 | 489 |
| 11.7 | 总结 | 421 | 13.3 | 多态公有继承 | 490 |
| 11.8 | 复习题 | 422 | | | |
| 11.9 | 编程练习 | 422 | | | |
| 第 12 章 | 类和动态内存分配 | 425 | | | |
| 12.1 | 动态内存和类 | 425 | | | |
| 12.1.1 | 复习示例和静态类成员 | 425 | | | |

| | | | |
|------------------------------|------------|---------------------------------|------------|
| 13.4 静态联编和动态联编 | 501 | 14.4.10 模板别名 (C++11) | 593 |
| 13.4.1 指针和引用类型的兼容性 | 502 | 14.5 总结 | 594 |
| 13.4.2 虚成员函数和动态联编 | 503 | 14.6 复习题 | 595 |
| 13.4.3 有关虚函数注意事项 | 505 | 14.7 编程练习 | 597 |
| 13.5 访问控制: protected | 507 | 第 15 章 友元、异常和其他 | 602 |
| 13.6 抽象基类 | 508 | 15.1 友元 | 602 |
| 13.6.1 应用 ABC 概念 | 510 | 15.1.1 友元类 | 602 |
| 13.6.2 ABC 理念 | 516 | 15.1.2 友元成员函数 | 606 |
| 13.7 继承和动态内存分配 | 516 | 15.1.3 其他友元关系 | 609 |
| 13.7.1 第一种情况: 派生类 不使用 new | 516 | 15.1.4 共同的友元 | 610 |
| 13.7.2 第二种情况: 派生类 使用 new | 517 | 15.2 嵌套类 | 611 |
| 13.7.3 使用动态内存分配和友元的 继承示例 | 519 | 15.2.1 嵌套类和访问权限 | 612 |
| 13.8 类设计回顾 | 523 | 15.2.2 模板中的嵌套 | 613 |
| 13.8.1 编译器生成的成员函数 | 523 | 15.3 异常 | 616 |
| 13.8.2 其他的类方法 | 524 | 15.3.1 调用 abort() | 616 |
| 13.8.3 公有继承的考虑因素 | 527 | 15.3.2 返回错误码 | 617 |
| 13.8.4 类函数小结 | 530 | 15.3.3 异常机制 | 619 |
| 13.9 总结 | 530 | 15.3.4 将对象用作异常类型 | 621 |
| 13.10 复习题 | 531 | 15.3.5 异常规范和 C++11 | 624 |
| 13.11 编程练习 | 531 | 15.3.6 栈解退 | 625 |
| 第 14 章 C++ 中的代码重用 | 534 | 15.3.7 其他异常特性 | 629 |
| 14.1 包含对象成员的类 | 534 | 15.3.8 exception 类 | 631 |
| 14.1.1 valarray 类简介 | 535 | 15.3.9 异常、类和继承 | 634 |
| 14.1.2 Student 类的设计 | 535 | 15.3.10 异常何时会迷失方向 | 639 |
| 14.1.3 Student 类示例 | 537 | 15.3.11 有关异常的注意事项 | 641 |
| 14.2 私有继承 | 543 | 15.4 RTTI | 642 |
| 14.2.1 Student 类示例 (新版本) | 543 | 15.4.1 RTTI 的用途 | 642 |
| 14.2.2 使用包含还是私有继承 | 549 | 15.4.2 RTTI 的工作原理 | 642 |
| 14.2.3 保护继承 | 549 | 15.5 类型转换运算符 | 649 |
| 14.2.4 使用 using 重新定义访问权限 | 550 | 15.6 总结 | 652 |
| 14.3 多重继承 | 551 | 15.7 复习题 | 653 |
| 14.3.1 有多少 Worker | 555 | 15.8 编程练习 | 654 |
| 14.3.2 哪个方法 | 558 | 第 16 章 string 类和标准模板库 | 655 |
| 14.3.3 MI 小结 | 567 | 16.1 string 类 | 655 |
| 14.4 类模板 | 567 | 16.1.1 构造字符串 | 655 |
| 14.4.1 定义类模板 | 568 | 16.1.2 string 类输入 | 659 |
| 14.4.2 使用模板类 | 570 | 16.1.3 使用字符串 | 661 |
| 14.4.3 深入探讨模板类 | 572 | 16.1.4 string 还提供了哪些功能 | 665 |
| 14.4.4 数组模板示例和非类型参数 | 577 | 16.1.5 字符串种类 | 666 |
| 14.4.5 模板多功能性 | 578 | 16.2 智能指针模板类 | 667 |
| 14.4.6 模板的具体化 | 582 | 16.2.1 使用智能指针 | 668 |
| 14.4.7 成员模板 | 584 | 16.2.2 有关智能指针的注意事项 | 670 |
| 14.4.8 将模板用作参数 | 586 | 16.2.3 unique_ptr 为何优于 auto_ptr | 672 |
| 14.4.9 模板类和友元 | 588 | 16.2.4 选择智能指针 | 673 |
| | | 16.3 标准模板库 | 674 |
| | | 16.3.1 模板类 vector | 675 |

| | | | | | |
|------------------------|-----------------------------|-----|--------------------------|--------------------------|-----|
| 16.3.2 | 可对矢量执行的操作 | 676 | 17.4.3 | 打开多个文件 | 771 |
| 16.3.3 | 对矢量可执行的其他操作 | 680 | 17.4.4 | 命令行处理技术 | 772 |
| 16.3.4 | 基于范围的 for 循环 (C++11) | 684 | 17.4.5 | 文件模式 | 773 |
| 16.4 | 泛型编程 | 684 | 17.4.6 | 随机存取 | 781 |
| 16.4.1 | 为何使用迭代器 | 685 | 17.5 | 内核格式化 | 788 |
| 16.4.2 | 迭代器类型 | 688 | 17.6 | 总结 | 790 |
| 16.4.3 | 迭代器层次结构 | 689 | 17.7 | 复习题 | 791 |
| 16.4.4 | 概念、改进和模型 | 690 | 17.8 | 编程练习 | 792 |
| 16.4.5 | 容器种类 | 695 | | | |
| 16.4.4 | 关联容器 | 702 | 第 18 章 探讨 C++ 新标准 | 795 | |
| 16.4.5 | 无序关联容器 (C++11) | 707 | 18.1 | 复习前面介绍过的 C++11 功能 | 795 |
| 16.5 | 函数对象 | 707 | 18.1.1 | 新类型 | 795 |
| 16.5.1 | 函数符概念 | 707 | 18.1.2 | 统一的初始化 | 795 |
| 16.5.2 | 预定义的函数符 | 710 | 18.1.3 | 声明 | 796 |
| 16.5.3 | 自适应函数符和函数适配器 | 711 | 18.1.4 | 智能指针 | 798 |
| 16.6 | 算法 | 713 | 18.1.5 | 异常规范方面的修改 | 798 |
| 16.6.1 | 算法组 | 713 | 18.1.6 | 作用域内枚举 | 799 |
| 16.6.2 | 算法的通用特征 | 714 | 18.1.7 | 对类的修改 | 799 |
| 16.6.3 | STL 和 string 类 | 715 | 18.1.8 | 模板和 STL 方面的修改 | 800 |
| 16.6.4 | 函数和容器方法 | 716 | 18.1.9 | 右值引用 | 801 |
| 16.6.5 | 使用 STL | 717 | 18.2 | 移动语义和右值引用 | 802 |
| 16.7 | 其他库 | 720 | 18.2.1 | 为何需要移动语义 | 802 |
| 16.7.1 | vector、valarray 和 array | 720 | 18.2.2 | 一个移动示例 | 803 |
| 16.7.2 | 模板 initializer_list (C++11) | 724 | 18.2.3 | 移动构造函数解析 | 808 |
| 16.7.3 | 使用 initializer_list | 725 | 18.2.4 | 赋值 | 809 |
| 16.8 | 总结 | 727 | 18.2.5 | 强制移动 | 809 |
| 16.9 | 复习题 | 728 | 18.3 | 新的类功能 | 813 |
| 16.10 | 编程练习 | 728 | 18.3.1 | 特殊的成员函数 | 813 |
| 第 17 章 输入、输出和文件 | 731 | | 18.3.2 | 默认的方法和禁用的方法 | 814 |
| 17.1 | C++ 输入和输出概述 | 731 | 18.3.3 | 委托构造函数 | 815 |
| 17.1.1 | 流和缓冲区 | 732 | 18.3.4 | 继承构造函数 | 815 |
| 17.1.2 | 流、缓冲区和 iostream 文件 | 733 | 18.3.5 | 管理虚方法: override 和 final | 817 |
| 17.1.3 | 重定向 | 735 | 18.4 | Lambda 函数 | 817 |
| 17.2 | 使用 cout 进行输出 | 736 | 18.4.1 | 比较函数指针、函数符和 Lambda 函数 | 818 |
| 17.2.1 | 重载的 << 运算符 | 736 | 18.4.2 | 为何使用 lambda | 820 |
| 17.2.2 | 其他 ostream 方法 | 738 | 18.5 | 包装器 | 822 |
| 17.2.3 | 刷新输出缓冲区 | 740 | 18.5.1 | 包装器 function 及模板的 低效性 | 823 |
| 17.2.4 | 用 cout 进行格式化 | 741 | 18.5.2 | 修复问题 | 825 |
| 17.3 | 使用 cin 进行输入 | 753 | 18.5.3 | 其他方式 | 826 |
| 17.3.1 | cin>> 如何检查输入 | 754 | 18.6 | 可变参数模板 | 827 |
| 17.3.2 | 流状态 | 756 | 18.6.1 | 模板和函数参数包 | 827 |
| 17.3.3 | 其他 istream 类方法 | 759 | 18.6.2 | 展开参数包 | 828 |
| 17.3.4 | 其他 istream 方法 | 764 | 18.6.3 | 在可变参数模板函数中使用 递归 | 828 |
| 17.4 | 文件输入和输出 | 768 | 18.7 | C++11 新增的其他功能 | 831 |
| 17.4.1 | 简单的文件 I/O | 768 | 18.7.1 | 并行编程 | 831 |
| 17.4.2 | 流状态检查和 is_open() | 770 | | | |

| | | | |
|-----------------------------------|-----|--|-----|
| 18.7.2 新增的库 | 831 | F.2.5 使用右值引用的构造函数 (C++11) | 866 |
| 18.7.3 低级编程 | 832 | F.2.6 使用一个字符的 n 个副本的构造 函数 | 867 |
| 18.7.4 杂项 | 832 | F.2.7 使用区间的构造函数 | 867 |
| 18.8 语言变化 | 832 | F.2.8 使用初始化列表的构造函数 (C++11) | 868 |
| 18.8.1 Boost 项目 | 833 | F.2.9 内存杂记 | 868 |
| 18.8.2 TR1 | 833 | F.3 字符串存取 | 868 |
| 18.8.3 使用 Boost | 833 | F.4 基本赋值 | 869 |
| 18.9 接下来的任务 | 834 | F.5 字符串搜索 | 869 |
| 18.10 总结 | 834 | F.5.1 find() 系列 | 870 |
| 18.11 复习题 | 835 | F.5.2 rfind() 系列 | 870 |
| 18.12 编程练习 | 838 | F.5.3 find_first_of() 系列 | 870 |
| 附录 A 计数系统 | 839 | F.5.4 find_last_of() 系列 | 871 |
| A.1 十进制数 | 839 | F.5.5 find_first_not_of() 系列 | 871 |
| A.2 八进制整数 | 839 | F.5.6 find_last_not_of() 系列 | 871 |
| A.3 十六进制数 | 839 | F.6 比较方法和函数 | 872 |
| A.4 二进制数 | 840 | F.7 字符串修改方法 | 873 |
| A.5 二进制和十六进制 | 841 | F.7.1 用于追加和相加的方法 | 873 |
| 附录 B C++保留字 | 842 | F.7.2 其他赋值方法 | 874 |
| B.1 C++关键字 | 842 | F.7.3 插入方法 | 874 |
| B.2 替代标记 | 842 | F.7.4 清除方法 | 875 |
| B.3 C++库保留名称 | 843 | F.7.5 替换方法 | 875 |
| B.4 有特殊含义的标识符 | 843 | F.7.6 其他修改方法: copy() 和 swap() | 876 |
| 附录 C ASCII 字符集 | 845 | F.8 输出和输入 | 876 |
| 附录 D 运算符优先级 | 849 | 附录 G 标准模板库方法和函数 | 877 |
| 附录 E 其他运算符 | 852 | G.1 STL 和 C++11 | 877 |
| E.1 按位运算符 | 852 | G.1.1 新增的容器 | 877 |
| E.1.1 移位运算符 | 852 | G.1.2 对 C++98 容器所做的修改 | 877 |
| E.1.2 逻辑按位运算符 | 853 | G.2 大部分容器都有的成员 | 878 |
| E.1.3 按位运算符的替代表示 | 855 | G.3 序列容器的其他成员 | 881 |
| E.1.4 几种常用的按位运算符技术 | 856 | G.4 set 和 map 的其他操作 | 883 |
| E.2 成员解除引用运算符 | 857 | G.4 无序关联容器 (C++11) | 884 |
| E.3 alignof (C++11) | 860 | G.5 STL 函数 | 886 |
| E.4 noexcept (C++11) | 861 | G.5.1 非修改式序列操作 | 886 |
| 附录 F 模板类 string | 862 | G.5.2 修改式序列操作 | 890 |
| F.1 13 种类型和一个常量 | 862 | G.5.3 排序和相关操作 | 897 |
| F.2 数据信息、构造函数及其他 | 863 | G.5.4 数值运算 | 907 |
| F.2.1 默认构造函数 | 865 | 附录 H 精选读物和网上资源 | 909 |
| F.2.2 使用 C-风格字符串的构造函数 | 865 | H.1 精选读物 | 909 |
| F.2.3 使用部分 C-风格字符串的构造 函数 | 865 | H.2 网上资源 | 910 |
| F.2.4 使用左值引用的构造函数 | 866 | 附录 I 转换为 ISO 标准 C++ | 911 |
| | | I.1 使用一些预处理器编译指令的替代品 | 911 |

| | |
|---|-------------------|
| I.1.1 使用 <code>const</code> 而不是 <code>#define</code> 来定义 常量.....911 | 第4章复习题答案.....917 |
| I.1.2 使用 <code>inline</code> 而不是 <code>#define</code> 来定义 小型函数.....912 | 第5章复习题答案.....919 |
| I.2 使用函数原型.....913 | 第6章复习题答案.....919 |
| I.3 使用类型转换.....913 | 第7章复习题答案.....920 |
| I.4 熟悉 C++特性.....913 | 第8章复习题答案.....922 |
| I.5 使用新的头文件.....914 | 第9章复习题答案.....924 |
| I.6 使用名称空间.....914 | 第10章复习题答案.....925 |
| I.7 使用智能指针.....915 | 第11章复习题答案.....927 |
| I.8 使用 <code>string</code> 类.....915 | 第12章复习题答案.....927 |
| I.9 使用 STL.....915 | 第13章复习题答案.....929 |
| 附录 J 复习题答案916.. | 第14章复习题答案.....930 |
| 第2章复习题答案.....916 | 第15章复习题答案.....931 |
| 第3章复习题答案.....916 | 第16章复习题答案.....932 |
| | 第17章复习题答案.....933 |
| | 第18章复习题答案.....935 |