

C++11

程序设计

(英文版)

(第2版)

[美] Paul Deitel 著
Harvey Deitel



Deitel (TM)

C++11
for Programmers, 2E



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

· 原味精品书系 ·

C++11程序设计

(第2版)

(英文版)

C++11 for Programmers, 2E

[美] Paul Deitel
Harvey Deitel 著



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书利用活代码深入研究 C++11 和 C++ 标准库。内容包括 C++11 的新特性, 基于模板的标准库容器、迭代器和算法, C++ 标准库字符串, C++ 标准库数组, 构造重要类, 面向对象编程的案例研究, 异常处理, 等等。本书示例丰富, 包含了从计算机科学、商业、模拟、游戏和其他主题中挑选出来的各种示例程序, 并在三个具有工业强度的 C++11 编译器上对代码进行了测试; 书中配有大量的插图, 包括图表、线条图、UML 图、程序和程序输出。

本书适合具有一定高级语言编程背景的程序员阅读。

Original edition, entitled C++11 for Programmers, 2E, 0133439852, by Paul Deitel, Harvey Deitel, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright©2014 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.

China edition published by Pearson Education Asia Ltd. and Publishing House of Electronics Industry Copyright © 2016. The edition is manufactured in the People's Republic of China, and is authorized for sale and distribution only in the mainland of China exclusively (except Hong Kong SAR, Macau SAR, and Taiwan).

本书英文影印版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书仅限中国大陆境内 (不包括中国香港、澳门特别行政区和中国台湾地区) 销售发行。

本书英文影印版贴有 Pearson Education 培生教育出版集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号 图字: 01-2015-6095

图书在版编目 (CIP) 数据

C++11 程序设计: 第 2 版 = C++11 for Programmers, 2E: 英文 / (美) 戴特尔 (Deitel, P.), (美) 戴特尔 (Deitel, H.) 著. — 北京: 电子工业出版社, 2016.4
(原味精品书系)

ISBN 978-7-121-27264-6

I. ① C… II. ① 戴… ② 戴… III. ① C 语言—程序设计—英文 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2015) 第 227464 号

策划编辑: 张春雨

责任编辑: 徐津平

印刷: 北京中新伟业印刷有限公司

装订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开本: 787×980 1/16 印张: 52.75 字数: 1266 千字

版次: 2016 年 4 月第 1 版

印次: 2016 年 4 月第 1 次印刷

定价: 148.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

“语言的首要价值是清晰……”

——Galen

欢迎阅读本书。本书会向软件开发人员介绍最先进的计算技术。

我们专注于软件工程的最佳实践。本书沿用了 Deitel 特色的“程序实况解说”风格——用完整的可运行程序（而不是代码片段）来解释相关概念。每个完整的代码示例都配有生动的示例。所有的源代码都可以从下面的网址下载：

www.deitel.com/books/cpp11fp

如果你在阅读本书时遇到任何困难，请发送邮件到：

deitel@deitel.com

我们会及时回复你。要了解本书的最新信息，可以访问 www.deitel.com/books/cpp11fp。加入我们的 Facebook 社区（www.deitel.com/DeitelFan）、Twitter 社区（@deitel）、Google+ 社区（[gplus.to/deitel](https://plus.google.com/deitel)）和 LinkedIn 社区（bit.ly/DeitelLinkedIn），你还可以订阅 *Deitel® Buzz Online* 电子邮件新闻简报（www.deitel.com/newsletter/subscribe.html）。

本书特色

本书的主要特色如下所述。

C++11 标准

2011 年发布的 C++11 新标准是我们写这本书的动机。我们使用标有“11”的图标来表示在本书中提到的每个 C++11 新特性，如本页的空白处所示。这一版包含的一些 C++11 特性如下。

- 符合 **C++11 新标准**。广泛地覆盖了 C++11 的新特性（图 1）。
- 在 3 个具有工业强度的 **C++11 编译器** 上对代码进行了测试。我们在 GNU™ C++ 4.7、Microsoft® Visual C++® 2012 和 Apple® Xcode® 4.5 的 LLVM 上对代码示例进行了测试。
- **智能指针**。智能指针可以帮助你避免动态由于提供了内置指针无法实现的功能而导致的内存管理错误。我们将在第 17 章讨论 `unique_ptr`，在第 24 章讨论 `shared_ptr` 和 `weak_ptr`。

all_of 算法	基类构造函数的继承	非确定性随机数的生成
any_of 算法	返回迭代器的 insert 容器成员函数	none_of 算法
array 容器	is_heap 算法	数字转换函数
用于类型推断的 auto	is_heap_until 算法	nullptr
begin/end 函数	C++11 中的新关键字	override 关键字
cbegin/cend 容器成员函数	Lambda 表达式 (匿名函数)	基于范围的 for 语句
用于模板类型中 >> 的编译器修复	键-值 pair 的 List 初始化	正则表达式
copy_if 算法	pair 对象的 List 初始化	rvalue 引用
copy_n 算法	返回值的 List 初始化	作用域内枚举 (enum)
crbegin/crend 容器成员函数	List 初始化一个动态分配数组	shared_ptr 智能指针
decltype	List 初始化一个 vector	shrink_to_fit vector/deque 成员函数
函数模板中的默认类型参数	构造函数调用中的 List 初始化	指定一个枚举常量的类型
default 成员函数	long long int 类型	文件名的 static_assert 对象
委托构造函数	具有 initializer_list 参数的 min 和 max 算法	文件名的 string 对象
delete 成员函数	minmax 算法	swap 非成员函数
explicit 转换运算符	minmax_element 算法	函数的尾随返回类型
final 类	move 算法	tuple 可变参数模板
final 成员函数	移动赋值操作符	unique_ptr 智能指针
find_if_not 算法	move_backward 算法	无符号 long long int weak_ptr 智能指针
forward_list 容器	移动构造函数	
关联容器中的不可变键	noexcept	
类的初始化		

图 1 本书包含的一些 C++11 特性

- 更早地介绍基于模板的标准库容器、迭代器和算法，并用 C++11 新特性对这些内容进行了扩充。我们将第 1 版中位于第 20 章的标准库容器、迭代器和算法的处理移到了第 15 章和第 16 章，并且使用 C++11 新特性对这些内容进行了扩充。你需要的绝大多数数据结构都可以通过重用这些标准库来实现。
- 在线阅读第 24 章。我们将在这一章介绍 C++11 的其他主题。C++11 新标准是在 2011 年发布的，但并非所有的 C++ 编译器都完全支持这些特性。在我们编写本书时，如果某个 C++ 特性已经在这 3 个主要的编译器中实现了，我们就用文本和代码实例来介绍该特性。如果这 3 个编译器都没有实现该特性，我们就用粗斜体标题和对这个特性的简短说明来介绍它。当这些特性被编译器实现时，我们就会在第 24 章对这些简短说明进行扩充。之所以将本章放在网上，是因为这样做我们就可以动态更新其内容。本章包含对正则表达式、shared_ptr 和 weak_ptr 智能指针、移动语义等的讨论。你可以通过以下网址访问本章：

www.informit.com/title/9780133439854

- R 随机数生成、模拟和游戏。为了让程序更安全（参见下一页的“C++ 安全编程”一节），我们会在本书中讨论 C++11 的非确定性随机数生成机制。

面向对象程序设计

- **提早介绍对象。**我们在本书的第1章就介绍了对象技术的基本概念和术语，在第3章将会带你开发一个自定义的C++类和对象。
- **C++ 标准库字符串。**C++ 提供了两种类型的字符串——字符串类对象（从第3章开始使用）和C字符串（来自C语言）。我们用C++字符串类的实例替换了大多数C字符串，以便让程序变得更健壮，并消除C字符串带来的很多安全问题。我们稍后会在本书中讨论C字符串，以便让你能够使用工业上的遗留代码。在新的开发项目中，你应该支持字符串对象。
- **C++ 标准库数组。**我们在本书中使用的主要是标准库的数组类模板，而不是内置的、C风格的、基于指针的数组。但我们还是会介绍内置数组，因为它们仍然会被用在C++中，而且了解它才能够阅读遗留代码。C++ 提供了三种类型的数组——类模板数组、`vector`（从第7章开始使用），以及C风格的、基于指针的数组（在第8章介绍）。我们在本书中会酌情使用模板数组和类模板 `vector`（偶尔），而不是C数组。在新的开发项目中，你应该支持类模板数组和 `vector`。
- **构造重要类。**本书的一个主要目标是让你构造重要的、可重用的C++类。在第10章的案例研究中，你将创建自己的自定义 `Array` 类。第10章以类模板字符串的测试驱动开始，这样在你使用重载操作符实现自己的自定义类前，可以先看到如何使用操作符重载。
- **面向对象编程的案例研究。**我们提供了跨越多个章节并涵盖了软件开发生命周期的案例研究。这些案例研究包括第3~7章的 `GradeBook` 类，第9章的 `Time` 类和第11章、第12章的 `Employee` 类。第12章用详细的图表说明了C++如何“在幕后”实现多态、虚函数和动态绑定。
- **可选案例研究：使用 UML 创建一个 ATM 的面向对象设计和 C++ 实现。**UML™（统一建模语言，Unified Modeling Language）是为面向对象系统建模的行业标准图形化语言。我们在本书的前几章就介绍了UML。第22章和第23章包含使用UML来进行面向对象设计的可选案例研究。我们设计并实现了一个简单的自动取款机（ATM）的软件。我们分析了一个指定如何构建系统的典型需求文档。我们确定了实现该系统所需要的类、这些类需要拥有的属性和这些类需要表现出的行为，我们还指定了这些类必须如何交互才能满足系统需求。我们根据设计产生了一个完整的C++实现。读者经常反馈说这个案例研究“将所有内容都联系起来”，并让他们对面向对象有了更深入的了解。
- **异常处理。**在本书的前几章就包含了基本的异常处理。你可以从第17章来了解异常处理的更多内容。
- **主要的编程范式。**我们在本书中讨论了面向对象编程和泛型编程。

教学特色

- **示例。**本书包含了从计算机科学、商业、模拟、游戏和其他主题中挑选出来的各种示例程序。

- **插图和图。**本书包含了大量的图表、线条图、UML 图、程序和程序输出。

其他特色

- **指针。**本书详细地介绍了内置指针，以及内置指针、C 字符串和内置数组之间的密切关系。
- **调试器附录。**本书包含了 3 个调试器附录——附录 H，讲述 Visual Studio 调试器的使用；附录 I，讲述 GNU C++ 调试器的使用；附录 J，讲述 Xcode 调试器的使用。

C++ 安全编程

很难创建出可以抵抗病毒、蠕虫等其他“恶意软件”攻击的、具有工业强度的系统。如今，通过互联网，这类攻击瞬间就可以对全球范围造成影响。从开发周期的一开始就在软件中加入安全性，可以大大减少安全风险。

人们创建了 CERT[®] Coordination Center (www.cert.org) 来分析和及时应对攻击。CERT(计算机安全应急响应小组, Computer Emergency Response Team) 是一个政府资助的组织, 位于卡内基梅隆大学软件工程研究所。CERT 会发布和推广各种流行编程语言的安全编码标准, 帮助软件开发人员实现具有工业强度的系统, 从而避免会造成开放式系统受到攻击的编程实践。

我们要感谢 Robert C. Seacord, 他是 CERT 的安全编程管理员和卡内基梅隆大学计算机科学系的副教授。Seacord 先生是 *C How to Program, 7e* 一书的技术评审, 他从安全角度审查了这本书中的 C 语言程序, 并建议我们遵守 CERT C 语言安全编程标准。

本书也遵守 CERT C++ 安全编码标准指南 (符合书籍水平), 你可以通过以下网址找到该指南:

www.securecoding.cert.org

我们很高兴地发现, 从 20 世纪 90 年代早期开始我们就在书籍中推荐了这些编程实践。如果你要创建具有工业强度的 C++ 系统, 就一定不要错过 *Secure Coding in C and C++, 2e* (Robert Seacord, Addison-Wesley Professional 出版)。

教学方法

本书强调的是程序的清晰性, 并专注于构建良好的软件工程。

程序实况解说。本书包含上百个“程序实况解说”示例——每个新概念都是在完整的可运行 C++ 程序中提出的, 紧接着是显示该程序输入和输出的一个或者多个实际执行。

语法着色。为了增加可读性, 我们为代码添加了语法阴影, 这与大多数 IDE 和代码编辑器给代码的语法添加颜色是类似的。我们采用的代码的着色约定是:

注释用常规字体表示

关键字用粗黑字体表示

常量和字面值用比关键字稍浅一点的粗灰字来表示

所有的其他代码以黑体表示

代码高亮显示。本书在每个源代码程序的关键代码段都放置了一个灰色的矩形框。

使用字体突出显示。为了便于参考，我们用粗体表示每个定义性出现的关键概念和索引的页面引用，用加粗的 Helvetica 字体来强调屏幕组件（例如，**File** 菜单），用 Lucida 字体来强调 C 语言程序文本（例如，`int x = 5;`）。

网页存取。所有的源代码示例都可以从以下网址下载：

www.deitel.com/books/cpp11fp

目标。每章都包含一个本章目标的列表。

编程技巧。我们提供了编程技巧来帮助你关注程序开发的重要方面。这些技巧和实践是我们在 8 年编程和企业培训经验中积累的精华。



最佳编程实践

最佳编程实践让读者关注那些有助于产生更清晰、更易理解和维护程序的技术。



常见编程错误

指出这些常见的编程错误可以降低读者犯类似错误的可能性。



错误预防技巧

这些技巧包括揭露和删除程序中 bug 的建议；大多数技巧描述的是如何预防将 bug 带入 C++ 程序中。



性能技巧

这些技巧强调的是如何让程序运行得更快，或者尽量减少它们占用的内存。



可移植性技巧

可移植性技巧可以帮助你编写能够在各种平台上运行的代码。



软件工程意见

软件工程意见强调的是影响软件系统（尤其是大型系统）构建的架构和设计问题。

在线章节和附录

下面的章节和附录可以在线获取。

- 第 24 章，C++11 的其他主题（C++11: Additional Features）。

- 附录 F, C 遗留代码主题 (C Legacy Code Topics)。
- 附录 G, UML 2: 其他图表类型 (UML 2: Additional Diagram Types)。
- 附录 H, Visual Studio 调试器的使用 (Using the Visual Studio Debugger)。
- 附录 I, GNU C++ 调试器的使用 (Using the GNU C++ Debugger)。
- 附录 J, Xcode 调试器的使用 (Using the Xcode Debugger)。
- 附录 K, 对 Mac OS X 上 C++ 程序的测试驱动 (Test Driving a C++ Program on Mac OS X)。

可以通过以下网址访问在线章节和附录:

www.informit.com/register

你需要注册一个 InformIT 账号,并登录。登录后,你就能看到有一个“Register a Product”框。输入本书的 ISBN (9780133439854) 就可以访问本书的在线章节和附录了。

获取本书使用的软件

本书的代码示例是通过下面的 C++ 开发工具来编写的:

- 用于 Windows 桌面的免费 Visual Studio Express 2012, 其中包含了 Visual C++ 和其他 Microsoft 开发工具。它可以在 Windows 7 和 Windows 8 中运行,并可以从如下网址下载。

www.microsoft.com/express

- GNU 的免费 GNU C++ (gcc.gnu.org/install/binaries.html), 其中大多数 Linux 系统都已经预先安装了该编译器,它还可以被安装到 Mac OS X 和 Windows 系统中。
- Apple 的免费 Xcode, OS X 用户可以从 Mac App Store 中下载它。

C++11 基础: LiveLessons 视频培训产品的第 I 部分 ~ 第 III 部分

“C++ 11 基础: LiveLessons 视频培训产品的第 I 部分 ~ 第 III 部分”介绍了开始使用 C++ 构建健壮的、强大的软件时所需要知道的内容。它包含了二十多个小时与本书同步的专家培训。要了解关于 Deitel LiveLessons 视频产品的更多信息,请访问

www.deitel.com/livelessons

或者可以通过 deitel@deitel.com 联系我们。如果你已经订阅了 Safari Books Online (www.safaribooksonline.com), 你还可以获取我们的 LiveLessons 视频。这些 LiveLessons 已于 2013 年夏季上线。

致谢

我们要感谢 Deitel & Associates 有限公司的 Abbey Deitel 和 Barbara Deitel。他们为这个项目投入了大量时间。Abbey 参与撰写了本书的第 1 章和前言，她和 Barbara 仔细研究了 C++11 的新功能。

我们荣幸与 Prentice Hall/Pearson 的出版专家团队共同完成该项目。我们要感谢培生科技出版集团的主编 Mark L. Taub 给我们的指导和为此付出的精力。我们要感谢 Carole Snyder 出色地完成了招募优秀的 C++ 社区成员来审稿的工作；感谢 Chuti Prasertsith 为本书设计了具有创造力和精确度的封面——我们把自己对封面的构想告诉他，而他准确地实现了它；感谢 John Fuller 出色地完成了我们的 Deitel 开发人员系统图书和 LiveLessons 视频产品的出版工作。

审稿人员

还要感谢本书审稿人员的付出，他们提出的建设性的批评意见帮助我们完成了本书的新版本。他们完成了对本书文本和程序的审查，并为提高本书的表述方式提供了无数建议，他们是：Dean Michael Berris（谷歌，ISO C++ 委员会成员）、Danny Kaley（C++ 专家，经过认证的系统分析师和 C++ 标准委员会的前成员）、Linda M. Krause（艾姆赫斯特学院）、James P. McNellis（微软公司）、Robert C. Seacord（SEI/CERT 的安全编码管理员、*Secure Coding in C and C++* 一书的作者）、José Antonio González Seco（西班牙安达卢西亚会议议员）、Virginia Bailey（杰克逊州立大学）、Thomas J. Borrelli（罗彻斯特理工学院）、Ed Brey（科勒有限公司）、Chris Cox（Adobe 系统有限公司）、Gregory Dai（eBay）、Peter J. DePasquale（美国新泽西学院）、John Dibling（SpryWare）、Susan Gauch（阿肯色大学）、Doug Gregor（苹果公司）、Jack Hagemester（华盛顿州立大学）、Williams M. Higdon（美国印第安纳大学）、Anne B. Horton（Lockheed Martin 公司）、Terrell Hull（Logicalis 集成解决方案）、Ed James-Beckham（Borland 公司）、Wing-Ning Li（阿肯色大学）、Dean Mathias（犹他州立大学）、Robert A. McLain（潮水社区学院）、Robert Myers（佛罗里达州立大学）、Gavin Osborne（萨斯喀彻温省应用科技学院）、Amar Raheja（加利福尼亚大学波莫纳分校）、April Reagan（微软）、Raymond Stephenson（微软）、Dave Topham（美国奥龙尼学院）、Anthony Williams（作家和 C++ 标准委员会成员）和 Chad Willwerth（华盛顿大学塔科马分校）。

真诚地欢迎读者提出可以提高本书质量的意见、批评、更正和建议，请将它们发送到：

deitel@deitel.com

我们会及时回复你。我们在编写本书时享受到了很多乐趣。希望你也能在本书的阅读过程中得到乐趣。

Paul Deitel

Harvey Deitel

作者简介

Paul Deitel, Deitel & Associates 有限公司的 CEO 兼 CTO, 毕业于麻省理工学院, 主修信息技术。通过 Deitel & Associate 有限公司, 他已经为行业、政府机关和军队客户提供了数百节编程课程, 这些客户包括思科、IBM、西门子、Sun Microsystems、戴尔、Fidelity、肯尼迪航天中心、美国国家强风暴实验室、白沙导弹试验场、Rogue Wave Software、波音公司、SunGard Higher Education、北电网络公司、彪马、iRobot、Invensys 等。他和本书的合著者 Harvey Deitel 博士是畅销全球的编程语言教材、专业书籍和视频的作者。

Harvey Deitel 博士, Deitel & Associates 有限公司的董事长和首席战略官, 在计算机领域中拥有五十多年的经验。Deitel 博士获得了麻省理工学院电子工程(学习计算)的学士和硕士学位, 并获得了波士顿大学的数学博士学位(学习计算机科学)。在 20 世纪 60 年代, 通过先进的计算机技术和计算机应用公司, 他成为了创建各种 IBM 操作系统的团队中的一员。在 20 世纪 70 年代, 他创建了商业操作系统。他拥有丰富的大学教学经验, 在 1991 年与儿子 Paul Deitel 创办 Deitel & Associates 有限公司之前, 他是波士顿大学计算机科学系的主任, 并获得了终身任职权。Deitel 的出版物获得了国际上的认可, 并被翻译为繁体中文、简体中文、韩语、日语、德语、俄语、西班牙语、法语、波兰语、意大利语、葡萄牙语、希腊语、乌尔都语和土耳其语。Deitel 博士为很多大公司、学术研究机构、政府机关和军方提供了数百场专业编程讲座。

Deitel® Dive-Into® 企业培训

由 Paul Deitel 和 Harvey Deitel 创立的 Deitel & Associates 有限公司是一家国际知名的写作、企业培训和软件开发公司, 该公司专注于计算机编程语言、对象技术、手机应用程序开发, 以及 Internet 和 Web 软件技术方面的培训和写作。公司的客户包括一些全球最大的公司、政府机关、军队和学术研究机构。该公司向全球客户提供了由讲师主导的主要编程语言和平台课程, 包括 C++、Visual C++®、C、Java™、Visual C#®、Visual Basic®、XML®、Python®、对象技术、互联网和 Web 编程、Android™ 应用程序开发、Objective-C 和 iOS® 应用程序开发, 并且还在不断提供其他编程语言和软件开发课程。

通过与 Prentice Hall/Pearson 37 年的合作, Deitel & Associates 有限公司出版了一系列国际一流的专业编程书籍、大学教科书和 LiveLessons 视频课程。读者可以通过以下邮件地址联系 Deitel & Associates 有限公司和作者:

deitel@deitel.com

要了解 Deitel Dive-Into® 系列企业培训课程的更多信息, 请访问:

www.deitel.com/training

如果贵机构希望获得关于由讲师主导的现场培训的相关建议, 可发送邮件到 deitel@deitel.com。

想购买 Deitel 书籍和 LiveLessons 视频培训的个人可以访问 **www.deitel.com**。公司、政府机关、军队和学术研究机构的大宗团购可以直接与 Pearson 联系。要了解更多信息, 请访问:

www.informit.com/store/sales.aspx

第 24 章及附录 F~K 为 PDF 文档，请访问 www.informit.com/title/9780133439854 或博文官网（<http://www.broadview.com.cn/27264>）。

前言

xvii

I	Introduction	I
1.1	Introduction	2
1.2	C++	2
1.3	Object Technology	3
1.4	Typical C++ Development Environment	6
1.5	Test-Driving a C++ Application	9
1.6	Operating Systems	15
1.6.1	Windows—A Proprietary Operating System	15
1.6.2	Linux—An Open-Source Operating System	15
1.6.3	Apple's OS X; Apple's iOS for iPhone®, iPad® and iPod Touch® Devices	16
1.6.4	Google's Android	16
1.7	C++11 and the Open Source Boost Libraries	17
1.8	Web Resources	18
2	Introduction to C++ Programming, Input/Output and Operators	19
2.1	Introduction	20
2.2	First Program in C++: Printing a Line of Text	20
2.3	Modifying Our First C++ Program	23
2.4	Another C++ Program: Adding Integers	24
2.5	Arithmetic	28
2.6	Decision Making: Equality and Relational Operators	32
2.7	Wrap-Up	35
3	Introduction to Classes, Objects and Strings	37
3.1	Introduction	38
3.2	Defining a Class with a Member Function	38
3.3	Defining a Member Function with a Parameter	41
3.4	Data Members, <i>set</i> Member Functions and <i>get</i> Member Functions	44
3.5	Initializing Objects with Constructors	50

3.6	Placing a Class in a Separate File for Reusability	54
3.7	Separating Interface from Implementation	58
3.8	Validating Data with <i>set</i> Functions	63
3.9	Wrap-Up	68

4 Control Statements: Part 1; Assignment, ++ and -- Operators 69

4.1	Introduction	70
4.2	Control Structures	70
4.3	if Selection Statement	73
4.4	if...else Double-Selection Statement	74
4.5	while Repetition Statement	78
4.6	Counter-Controlled Repetition	79
4.7	Sentinel-Controlled Repetition	85
4.8	Nested Control Statements	92
4.9	Assignment Operators	95
4.10	Increment and Decrement Operators	96
4.11	Wrap-Up	99

5 Control Statements: Part 2; Logical Operators 100

5.1	Introduction	101
5.2	Essentials of Counter-Controlled Repetition	101
5.3	for Repetition Statement	102
5.4	Examples Using the for Statement	106
5.5	do...while Repetition Statement	110
5.6	switch Multiple-Selection Statement	112
5.7	break and continue Statements	121
5.8	Logical Operators	122
5.9	Confusing the Equality (==) and Assignment (=) Operators	127
5.10	Wrap-Up	128

6 Functions and an Introduction to Recursion 129

6.1	Introduction	130
6.2	Math Library Functions	130
6.3	Function Definitions with Multiple Parameters	132
6.4	Function Prototypes and Argument Coercion	137
6.5	C++ Standard Library Headers	139
6.6	Case Study: Random Number Generation	141
6.7	Case Study: Game of Chance; Introducing enum	146
6.8	C++11 Random Numbers	151
6.9	Storage Classes and Storage Duration	152
6.10	Scope Rules	155
6.11	Function Call Stack and Activation Records	158

6.12	Functions with Empty Parameter Lists	162
6.13	Inline Functions	163
6.14	References and Reference Parameters	164
6.15	Default Arguments	167
6.16	Unary Scope Resolution Operator	169
6.17	Function Overloading	170
6.18	Function Templates	173
6.19	Recursion	175
6.20	Example Using Recursion: Fibonacci Series	179
6.21	Recursion vs. Iteration	182
6.22	Wrap-Up	184

7 Class Templates array and vector; Catching Exceptions 185

7.1	Introduction	186
7.2	arrays	186
7.3	Declaring arrays	188
7.4	Examples Using arrays	188
7.4.1	Declaring an array and Using a Loop to Initialize the array's Elements	188
7.4.2	Initializing an array in a Declaration with an Initializer List	189
7.4.3	Specifying an array's Size with a Constant Variable and Setting array Elements with Calculations	190
7.4.4	Summing the Elements of an array	192
7.4.5	Using Bar Charts to Display array Data Graphically	193
7.4.6	Using the Elements of an array as Counters	195
7.4.7	Using arrays to Summarize Survey Results	196
7.4.8	Static Local arrays and Automatic Local arrays	198
7.5	Range-Based for Statement	200
7.6	Case Study: Class GradeBook Using an array to Store Grades	202
7.7	Sorting and Searching arrays	209
7.8	Multidimensional arrays	211
7.9	Case Study: Class GradeBook Using a Two-Dimensional array	214
7.10	Introduction to C++ Standard Library Class Template vector	221
7.11	Wrap-Up	227

8 Pointers 228

8.1	Introduction	229
8.2	Pointer Variable Declarations and Initialization	229
8.3	Pointer Operators	231
8.4	Pass-by-Reference with Pointers	233
8.5	Built-In Arrays	238
8.6	Using const with Pointers	240
8.6.1	Nonconstant Pointer to Nonconstant Data	241
8.6.2	Nonconstant Pointer to Constant Data	241

8.6.3	Constant Pointer to Nonconstant Data	243
8.6.4	Constant Pointer to Constant Data	243
8.7	sizeof Operator	244
8.8	Pointer Expressions and Pointer Arithmetic	247
8.9	Relationship Between Pointers and Built-In Arrays	249
8.10	Pointer-Based Strings	252
8.11	Wrap-Up	255

9 Classes: A Deeper Look; Throwing Exceptions 256

9.1	Introduction	257
9.2	Time Class Case Study	258
9.3	Class Scope and Accessing Class Members	264
9.4	Access Functions and Utility Functions	265
9.5	Time Class Case Study: Constructors with Default Arguments	266
9.6	Destructors	272
9.7	When Constructors and Destructors Are Called	272
9.8	Time Class Case Study: A Subtle Trap—Returning a Reference or a Pointer to a <code>private</code> Data Member	276
9.9	Default Memberwise Assignment	279
9.10	<code>const</code> Objects and <code>const</code> Member Functions	281
9.11	Composition: Objects as Members of Classes	283
9.12	<code>friend</code> Functions and <code>friend</code> Classes	289
9.13	Using the <code>this</code> Pointer	291
9.14	<code>static</code> Class Members	297
9.15	Wrap-Up	302

10 Operator Overloading; Class `string` 303

10.1	Introduction	304
10.2	Using the Overloaded Operators of Standard Library Class <code>string</code>	305
10.3	Fundamentals of Operator Overloading	308
10.4	Overloading Binary Operators	309
10.5	Overloading the Binary Stream Insertion and Stream Extraction Operators	310
10.6	Overloading Unary Operators	314
10.7	Overloading the Unary Prefix and Postfix <code>++</code> and <code>--</code> Operators	315
10.8	Case Study: A Date Class	316
10.9	Dynamic Memory Management	321
10.10	Case Study: Array Class	323
	10.10.1 Using the Array Class	324
	10.10.2 Array Class Definition	328
10.11	Operators as Member vs. Non-Member Functions	336
10.12	Converting Between Types	337
10.13	<code>explicit</code> Constructors and Conversion Operators	338
10.14	Overloading the Function Call Operator <code>()</code>	340
10.15	Wrap-Up	341

11	Object-Oriented Programming: Inheritance	342
11.1	Introduction	343
11.2	Base Classes and Derived Classes	343
11.3	Relationship between Base and Derived Classes	346
11.3.1	Creating and Using a <code>CommissionEmployee</code> Class	346
11.3.2	Creating a <code>BasePlusCommissionEmployee</code> Class Without Using Inheritance	351
11.3.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	357
11.3.4	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using protected Data	361
11.3.5	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using private Data	364
11.4	Constructors and Destructors in Derived Classes	369
11.5	<code>public</code> , <code>protected</code> and <code>private</code> Inheritance	371
11.6	Software Engineering with Inheritance	372
11.7	Wrap-Up	372
12	Object-Oriented Programming: Polymorphism	374
12.1	Introduction	375
12.2	Introduction to Polymorphism: Polymorphic Video Game	376
12.3	Relationships Among Objects in an Inheritance Hierarchy	376
12.3.1	Invoking Base-Class Functions from Derived-Class Objects	377
12.3.2	Aiming Derived-Class Pointers at Base-Class Objects	380
12.3.3	Derived-Class Member-Function Calls via Base-Class Pointers	381
12.3.4	Virtual Functions and Virtual Destructors	383
12.4	Type Fields and <code>switch</code> Statements	390
12.5	Abstract Classes and Pure virtual Functions	390
12.6	Case Study: Payroll System Using Polymorphism	392
12.6.1	Creating Abstract Base Class <code>Employee</code>	393
12.6.2	Creating Concrete Derived Class <code>SalariedEmployee</code>	397
12.6.3	Creating Concrete Derived Class <code>CommissionEmployee</code>	399
12.6.4	Creating Indirect Concrete Derived Class <code>BasePlusCommissionEmployee</code>	401
12.6.5	Demonstrating Polymorphic Processing	403
12.7	(Optional) Polymorphism, Virtual Functions and Dynamic Binding “Under the Hood”	407
12.8	Case Study: Payroll System Using Polymorphism and Runtime Type Information with Downcasting, <code>dynamic_cast</code> , <code>typeid</code> and <code>typeid</code>	410
12.9	Wrap-Up	414
13	Stream Input/Output: A Deeper Look	415
13.1	Introduction	416

13.2	Streams	417
13.2.1	Classic Streams vs. Standard Streams	417
13.2.2	iostream Library Headers	418
13.2.3	Stream Input/Output Classes and Objects	418
13.3	Stream Output	420
13.3.1	Output of char * Variables	421
13.3.2	Character Output Using Member Function put	421
13.4	Stream Input	422
13.4.1	get and getline Member Functions	422
13.4.2	istream Member Functions peek, putback and ignore	425
13.4.3	Type-Safe I/O	425
13.5	Unformatted I/O Using read, write and gcount	425
13.6	Introduction to Stream Manipulators	426
13.6.1	Integral Stream Base: dec, oct, hex and setbase	427
13.6.2	Floating-Point Precision (precision, setprecision)	427
13.6.3	Field Width (width, setw)	429
13.6.4	User-Defined Output Stream Manipulators	430
13.7	Stream Format States and Stream Manipulators	431
13.7.1	Trailing Zeros and Decimal Points (showpoint)	432
13.7.2	Justification (left, right and internal)	433
13.7.3	Padding (fill, setfill)	435
13.7.4	Integral Stream Base (dec, oct, hex, showbase)	436
13.7.5	Floating-Point Numbers; Scientific and Fixed Notation (scientific, fixed)	437
13.7.6	Uppercase/Lowercase Control (uppercase)	438
13.7.7	Specifying Boolean Format (boolalpha)	438
13.7.8	Setting and Resetting the Format State via Member Function flags	439
13.8	Stream Error States	440
13.9	Tying an Output Stream to an Input Stream	443
13.10	Wrap-Up	443

14 File Processing 444

14.1	Introduction	445
14.2	Files and Streams	445
14.3	Creating a Sequential File	446
14.4	Reading Data from a Sequential File	450
14.5	Updating Sequential Files	456
14.6	Random-Access Files	456
14.7	Creating a Random-Access File	457
14.8	Writing Data Randomly to a Random-Access File	462
14.9	Reading from a Random-Access File Sequentially	464
14.10	Case Study: A Transaction-Processing Program	466
14.11	Object Serialization	473
14.12	Wrap-Up	473