

TURING

图灵原版计算机科学系列

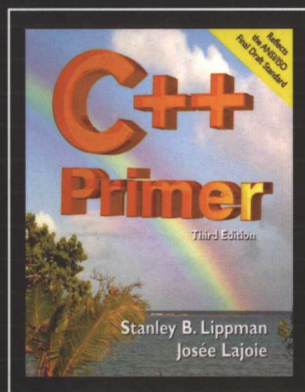
PEARSON
Addison
Wesley

本书英文原版
全球销量超过 450 000 册!

C++ Primer

(英文版·第3版)

[美] Stanley B. Lippman 著
[加] Josée Lajoie



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵原版计算机科学系列

C++ Primer

(英文版·第3版)

[美]Stanley B. Lippman

[加]Josée Lajoie

著



人民邮电出版社

POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

C++ Primer: 第3版 / (美) 李普曼, (加) 拉茹瓦著.

—北京: 人民邮电出版社, 2005.9

(图灵原版计算机科学系列)

ISBN 7-115-14056-1

I. C... II. ①李...②拉... III. C语言—程序设计—英文
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 109131 号

图灵原版计算机科学系列

C++ Primer (英文版·第3版)

◆ 著 [美] Stanley B. Lippman [加] Josée Lajoie

责任编辑 杨海玲

◆ 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 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京密云春雷印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 850×1168 1/32

印张: 39.5

字数: 1 050 千字

2005 年 9 月第 1 版

印数: 4 001—5 000 册

2006 年 6 月北京第 3 次印刷

著作权合同登记号 图字: 01-2005-3578 号

ISBN 7-115-14056-1/TP·5004

定价: 69.00 元

读者服务热线: (010) 88593802 印装质量热线: (010) 67129223

内 容 提 要

本书全面讲解了 C++ 程序设计语言的特性和用法。全书分五个部分对 C++ 进行阐述。第一部分是 C++ 概述，第二部分在此基础上介绍 C++ 语言，主要涉及数据类型、表达式、语句及抽象容器类型，第三部分是基于过程的程序设计，主要介绍函数、域和生命期、重载函数、函数模板、异常处理及泛型算法，第四部分是基于对象的程序设计，主要介绍类、重载操作符及类模板，第五部分是面向对象的程序设计，主要讲解类继承和子类型、多继承和虚拟继承及 `iostream` 库等。本书结合了 C++ 大师 Stanley B. Lippman 的实践经验和 C++ 标准委员会原负责人 Josée Lajoie 对于 ANSI/ISO 标准的深入理解，各个层次的程序设计人员都会从本书中获益匪浅。

前 言

本书的第 3 版和第 2 版之间变化显著。其中最引人瞩目的地方是，C++ 已经完成了国际标准化过程，这不仅为语言增加了许多新的特性，比如异常处理、运行时类型识别（RTTI）、名字空间、内置布尔数据类型、新的强制转换表示方式，而且还大幅度修改并扩展了现有的特性，比如模板（template）、同时支持面向对象（object-oriented）程序设计和基于对象（object-based）程序设计所需的类机制、嵌套类型以及重载函数的解析机制。也许尤其重要的是，标准 C++ 中现在涵盖了一个非常丰富的库，其中包含了以前称为标准模板库（STL）的内容。新的 string 类型，一组顺序和关联容器类型（比如 vector、list、map 和 set）以及对这些类型进行操作的一组可扩展的泛型算法都是这个新标准库的组成部分。本书不仅增加了许多新的材料，而且还谈论了如何用 C++ 进行程序设计的新的思考方式。简而言之，事实上不仅 C++ 本身已经脱胎换骨，本书第 3 版亦然。

在第 3 版中，不但对语言的阐述方式有了根本性的变化，连作者也发生了变动：首先，人数已经翻了一番；其次，写作过程已经国际化，虽然我们也都来自北美大陆——Stanley Lippman 是美国人，Josée Lajoie 是加拿大人。最后，这种两位一体的作者关系也反映了 C++ 社区的两种主要活动：Stanley 现在任职于迪士尼动画公司¹，将 C++ 应用于三维计算机图形和动画制作以提高工作效率；Josée 正在从事 C++ 的定义与实现，而且还担任着 C++ 标准语言核心小组的主席²，同时还是 IBM 加拿大研究中心 C++ 编译器组的成员。

Stanley 是贝尔实验室中曾与 C++ 之父 Bjarne Stroustrup 并肩工作的早期成员之一，自 1984 年起就一直从事 C++ 方面的工作。Stanley 曾经致力于最早的 C++

1. Stanley Lippman 现已成为 Microsoft C++/CLI 的架构设计师。

2. Josée Lajoie 现已离开 IBM，进入滑铁卢大学计算机科学系攻读硕士学位，并已不再在 C++ 标准化委员会中任职。

编译器 cfront 各个版本的开发，从 1986 年的 1.1 版到 3.0 版，并领导了 2.1 和 3.0 版的开发小组。之后他参与了 Stroustrup 领导的 Foundation 研究项目中关于程序设计环境的对象模型部分。

Josée 担任 IBM 加拿大研究中心 C++ 编译器组的成员已达 8 年时间。从 1990 年起她成为 C++ 标准委员会成员。曾经担任该委员会的副主席 3 年，迄今担任语言核心小组子委员会的主席达 4 年之久。

本书第 3 版进行了大幅修订，不仅反映了语言的变化和扩展，也反映了作者洞察力和经验的变化。

本书的结构

本书为 C++ 国际标准提供了全面的介绍。它提供了一种有意为之的讲述 C++ 语言的指导性方法，正是从这个意义上来说，它被称为 primer（入门书）。（但它并不是要为 C++ 语言进行简单化的或者说“温和”的叙述，从这个意义上来说它又不是一本入门书。）C++ 语言的各个程序设计方面，如异常处理、容器类型、面向对象程序设计等，都置于解决特定问题或者程序设计任务的上下文中阐述。C++ 语言的各种规则，如重载函数调用的解析和面向对象程序设计中 supported 的类型转换，本书都进行了广泛的论述，乍看来这似乎并不适合出现在一本入门书中。我们相信这样的选材对于实际理解 C++ 语言而言是必要的，在我们看来，这些材料应该不时回头重读，而不是一次就能消化的。如果感觉这些内容一开始太难理解或者太过枯燥，那就先将它们放在一边，留待来日——我们用特殊的符号 ◆ 来标记这样的小节。

阅读本书不需要 C 语言的知识，虽然熟悉某种现代的结构化语言能够加快学习的进度。本书的目的是作为学习 C++ 的第一本书，而不是学习程序设计的第一本书！当然，我们最开始还是会使用通用的词汇；然而，最初的几章讨论了一些基本概念，比如循环语句和变量等，有些读者可能会觉得过于简单。不必担心，有深度的内容很快就会出现。

C++ 的强大功能很多都源自它支持各种程序设计的新方式和对各种程序设计问题新的思考方式。所以，要想高效地学习如何使用 C++，决不能只是简单地学

习一组新的语法和语义。为了使这种宽泛的学习更加容易，本书将围绕一系列经过扩展的示例来组织。这些示例既用于介绍各种语言特性的细节，也用来引出这些语言特性。在一个完整示例的上下文中学习语言特性时，这些特性的用处就会其义自现，因为我们能够对何时和怎样在解决实际问题过程中使用它们有感性认识。此外，通过这样关注于示例的方式，读者能够较早地使用一些概念，而随着读者的知识基础逐渐建立，这些概念将得到更加完整地阐释。书中较早出现的示例含有对 C++ 基本概念的简单运用，读者可以借此初步体会一下 C++ 程序设计，而无需完全理解设计和实现的细节。

第 1 章 (Chapter 1) 和第 2 章 (Chapter 2) 组成了一个自成一体的对 C++ 语言的介绍和概述。第一部分 (Part I) 的目的是使我们更快地理解 C++ 支持的概念和语言设施，以及编写和执行一个程序所需要的基础知识。读完这一部分之后，应该对 C++ 语言提供的支持有所认识，但是又感觉还没有真正理解 C++。这样很好，这正是本书其余部分的目的所在！

第 1 章介绍了语言的基本要素：内置数据类型、变量、表达式、语句以及函数。这一章将观察一个极小的合法 C++ 程序，简要讨论编译程序的过程，介绍所谓的预处理器 (preprocessor)，初步了解语言对输入和输出的支持。这一章还给出了许多简单而完整的 C++ 程序，读者可以自己编译并执行它们。第 2 章介绍了 C++ 使用类机制对基于对象程序设计和面向对象程序设计所提供的支持，通过数组抽象的演化对此进行说明。此外，还简要介绍了模板、名字空间、异常处理和标准库为通用的容器类型和泛型程序设计提供的支持。这一章的进展速度较快，有些读者可能会感觉无法接受。如果这样的话，我们建议跳过这一章，以后再回头重读。

对于 C++ 至关重要的，是用户用来定义新的数据类型以扩展语言本身的各种设施。这些新的类型可以像内置类型一样灵活和简单地使用。但是精通语言的第一步是理解基本语言本身。第 3 章 (Chapter 3) 到第 6 章 (Chapter 6) (即第二部分, Part 2) 在这个层次上对 C++ 语言进行了介绍。

第 3 章介绍 C++ 语言预定义的内置数据类型和复合数据类型，还有 C++ 标准库提供的 string、complex 和 vector 类数据类型。这些类型构成了所有程序的基本要素。第 4 章 (Chapter 4) 详细讨论了 C++ 语言支持的表达式，如算术表达式、

关系表达式和赋值表达式。第 5 章 (Chapter 5) 的主题——语句，语句构成了 C++ 程序中最小的独立单元。C++ 标准库提供的容器类型是第 6 章的焦点。我们并没有简单地罗列所有操作，而是通过一个文本查询系统的实现来说明这些容器类型的设计和使用。

第 7 章 (Chapter 7) 到第 12 章 (Chapter 12) (即第三部分, Part III) 集中讨论 C++ 对基于过程的程序设计所提供的支持。第 7 章介绍了 C++ 的函数机制。函数将通常组成一项任务的一组操作封装起来, 比如 `print()`。(名字后面的空括号表明它代表一个函数。) 对程序作用域和变量生命期的概念以及名字空间设施的讨论是第 8 章 (Chapter 8) 的主题。第 9 章 (Chapter 9) 在第 7 章函数介绍的基础上进一步介绍了函数重载。函数重载能够使提供一个公共操作的多个函数实例共享一个相同的名字 (但是要求实现代码不同)。例如, 我们可以定义一组 `print()` 函数来输出不同类型的数据。第 10 章 (Chapter 10) 介绍和说明了函数模板的用法。函数模板为自动生成多个 (可能无限多个) 函数实例 (这些函数实例类型不同但实现方式保持不变) 提供了一种规范。

C++ 支持异常处理设施。所谓异常表示的是一个预料之外的程序行为, 比如所有可用的程序内存耗尽。出现异常情况的程序片段会抛出异常, 也就是说, 它能够被程序的其他部分访问到。然后程序中的某个函数必须捕获这个异常, 并进行必要处理。对异常处理的讨论分为两章。第 11 章 (Chapter 11) 中通过一个简单的捕获和抛出类类型异常的例子, 介绍了异常处理的基本语法和使用。因为在程序中处理的实际异常通常是一个面向对象类层次结构中的类对象, 所以第 19 章 (Chapter 19) 将继续讨论如何抛出和处理异常, 也就是在介绍了面向对象程序设计之后。

第 12 章介绍标准库所提供的丰富的泛型算法集合, 并考察它们如何与第 6 章中介绍的容器类型以及与内置数组类型互相作用。这一章的开始概述了如何使用泛型算法进行程序设计。第 6 章介绍的迭代器 (iterator) 将在第 12 章中进一步讨论, 因为它们提供了将泛型算法与实际容器绑定起来的胶合剂。这一章还介绍并阐述了函数对象的概念。通过函数对象, 我们能够为泛型算法中用到的操作符 (比如等于操作符或小于操作符) 提供另一种语义。附录中对泛型算法进行了详细说明, 并包括用法示例。

第 13 章 (Chapter 13) 到第 16 章 (Chapter 16) (即第四部分, Part IV) 集中讨论基于对象程序设计——也就是用于创建独立的抽象数据类型的类设施的定义和用法。C++ 允许程序员通过创建新的类型来描述问题领域, 从而在编写应用程序时无需再操心各种琐碎的底层工作。应用程序的基本类型可以只实现一次但是多次重用, 这样程序员就能将注意力放在问题本身, 而非实现细节上。这些将数据封装起来的设施能够大大地简化以后对应用程序的维护和改善工作。

第 13 章集中讨论通用的类机制: 如何定义一个类、信息隐藏 (即将类的公有接口与私有实现分离) 的概念以及如何定义和封装一个类的对象实例, 同时还讨论了类的作用域、嵌套类和作为名字空间成员的类。

第 14 章 (Chapter 14) 详细讨论 C++ 如何使用一些特殊的成员函数 (称为构造函数、析构函数和复制赋值操作符) 分别为类对象的初始化、析构和赋值提供特殊的支持。这一章我们还将考察按成员初始化和按成员复制 (其中一个类对象用类的另一个对象初始化或者赋值) 的问题, 以及用来有效地支持按成员初始化和按成员复制的命名返回值优化技术。

第 15 章 (Chapter 15) 将介绍特定于类的操作符重载, 首先给出一般概念和设计方面的考虑因素。然后介绍一些特殊的操作符, 如赋值、下标、调用和特定于类的 `new` 和 `delete` 操作符。这一章还讲述了具有特殊访问权限的类友元的概念, 以及它的必要性。然后讨论了用户定义的转换, 包括底层概念和一个演示其用法的扩展示例。这一章还详细讨论了函数重载解析的规则, 用许多代码示例进行了阐释。

第 16 章的主题是类模板。类模板是用来创建类的规范, 这里的类有一个或多个参数化的类型或值。例如, 一个 `vector` 类可以对所包含的元素类型进行参数化。一个 `buffer` 类可以对所包含的元素类型以及缓冲区的大小进行参数化。在更复杂的使用情形 (比如分布式计算) 中, `IPC` (进程间通信) 接口、寻址接口、同步化接口等都可以参数化。这一章还包括对如何定义类模板, 如何创建类模板的特定类型实例, 如何定义类模板的成员 (成员函数、静态成员和嵌套类型), 以及如何使用类模板组织程序等内容的讨论。最后是一个扩展的类模板示例。

第 17 章 (Chapter 17)、第 18 章 (Chapter 18)、第 19 章 (Chapter 19) 和第 20 章 (Chapter 20) (即第五部分, Part V) 的主题是面向对象程序设计和 C++ 对此的支持机制。第 17 章介绍了 C++ 支持面向对象程序设计主要要素的设施: 继承和动态绑定。面向对象程序设计中, 具有共同行为的各个类之间定义了父/子关系 (也称为类型/子类型关系)。类无需重新实现那些共同的特性, 可以从父类中继承数据和操作。在子类或者说子类型中只需编写不同于父类的地方。例如, 我们可以定义一个父类类型 `Employee` 及两个子类型 `TemporaryEmpl` 和 `Manager`。这些子类型将继承了 `Employee` 的所有行为, 因此只需要实现各自类型独有的行为即可。

继承的另一个方面称为多态性, 是指父类型引用其派生的任何子类型的能力。例如 `Employee` 的指针或者引用可以指向自己的类型也可以指向 `TemporaryEmpl` 或者 `Manager`。而动态绑定则是指在运行时根据多态对象的实际类型确定应该执行哪个操作的能力。这在 C++ 中是通过虚拟函数机制处理的。

第 17 章介绍了面向对象程序设计的基本特性。这一章演示了如何设计和实现一个 `Query` 类层次结构, 以支持第 6 章中开始实现的文本查询系统。

第 18 章介绍了更为复杂的继承层次结构, 这种层次结构是通过多重继承和虚拟继承机制实现的。这一章使用多重继承和虚拟继承将第 16 章中的模板类实例扩展为一个三层类模板层次结构。

第 19 章介绍了 RTTI (运行时类型识别) 设施。通过 RTTI, 在执行过程中程序能够查询多态类对象获取其类型。例如, 我们可以查询一个 `Employee` 对象, 看它是否真指向一个 `Manager` 类型。此外, 第 19 章再次涉及异常处理机制, 讨论了标准库的异常类层次结构, 并说明了如何定义和处理自己的异常类层次结构。这一章还深入讨论了在存在继承的情况下重载函数的解析过程。

第 20 章详细说明了如何使用 C++ 的 `iostream` 输入/输出库。它提供了对一般数据输入和输出的解释和示例, 定义特定于类的输入输出操作符实例的解释和示例, 以及如何识别和设置条件状态和如何格式化数据的解释和示例。`iostream` 库是用虚拟继承和多重继承实现的一个类层次结构。

本书的最后是一个附录, 它按数字和字母的顺序简短地讨论了所有泛型算法, 每个算法还给出了程序示例, 便于查阅。

最后要说明的是, 在写一本书的时候, 未包括的内容常常与讲到的内容同样

重要。C++语言的某些方面（比如对构造函数工作机理的详细探讨，在哪些情况下编译器会创建内部临时对象，或者一般性的对效率的考虑）虽然对于编写实际的应用程序非常重要，但是不适合一本介绍语言的教程。在着手本书第3版的写作之前，Stanley Lippman 写了 *Inside the C++ Object Model* 一书（参见前言所参考文献中的[LIPPMAN96a]），其中讲述了大量配套材料。在读者可能希望有更详细的解释的地方，特别是讨论基于对象程序设计和面向对象程序设计时，本书常常会引用该书的讨论。

本书有意省略了 C++标准库中的某些部分，比如本地化库和数值计算库的支持。C++标准库博大精深，想涵盖方方面面将远远超出本书的范围。在前言所列参考文献中有些书更详细地讨论了标准库（参见 [MUSSER96] 和 [STROUSTRUP97]）。我们相信本书出版之后，一定还会有更多讨论 C++标准库各方面的书出版。

第3版的变化

本书第3版主要有以下四个方面的变化：

1) 涵盖了语言中增加的新特性：异常处理、运行时类型识别、名字空间、内置布尔类型、新的类型强制转换方式。

2) 涵盖了新的 C++标准库，包括 `complex` 和 `string` 类型、`auto_ptr` 和 `pair` 类型、顺序容器和关联容器类型（主要是 `list`、`vector`、`map`、`set` 容器）以及泛型算法。

3) 调整了原有的文字，以反映标准 C++中对原有语言特性的提炼、修改和扩展。提炼方面的例子是，现在能够前向声明一个嵌套类型了，而这在以前是不允许的。修改的例子是，虚拟函数的派生类实例能够返回一个从基类实例的返回类型派生的类。这种修改支持了称为克隆或工厂的方法（`clone()`虚拟函数在 17.4.7 节中进行了解释）。对原有语言特性扩展的例子是，现在可以显式地指定函数模板的一个或多个模板实参了。（实际上，模板已经得到了很大的扩展，差不多已经成为一个新特性。）

4) 改善了对许多 C++高级特性（具体而言就是模板和类）的讨论和组织方

式以及面向对象程序设计的讨论方式。Stanley 从一个相对较小的 C++ 开发者社区，转到了更普遍的 C++ 使用者社区，这种转换产生的影响在于，他开始相信，越能够深入洞察问题，程序员就越能够灵活地运用 C++ 语言。因此，第 3 版中许多情况下重点已经转移到更好地解释特性背后的概念和最好地使用这些特性的方法，在适当的时候指明要避免的潜在陷阱。

C++ 的未来

在本书出版的时候，ISO/ANSI C++ 标准委员会已经完成了第一个 C++ 国际标准的技术工作。该标准已于 1998 年夏天由 ISO 出版。

在标准出版之后，支持标准 C++ 的编译器将很快出现。随着标准的出版，C++ 语言的进化将趋于稳定。这样用标准 C++ 编写复杂的程序库解决各行业特有的问题将成为可能。因此，C++ 世界里的程序库有望得到蓬勃发展。

标准出版后，标准委员会仍将继续工作（尽管进度会慢一些），回应 C++ 标准使用者提出的进行解释的请求。这会导致对 C++ 标准进行一些小的澄清和修订。如果需要，国际标准将每 5 年修订一次，以考虑技术的变化和业界需要。

C++ 标准出版 5 年后会发生什么，当然是无法知道的。可能 C++ 标准库中会加入业界广泛使用的新库组件。但是到目前为止，C++ 标准委员会的工作已经完成，C++ 的命运完全掌握在使用者的手中。

致谢

和往常一样，我们要特别感谢 Bjarne Stroustrup，感谢他给予我们如此美妙的编程语言，感谢他多年来对我们的关心。还要特别感谢 C++ 标准委员会成员的奉献和艰苦工作（常常是自愿的）以及对 C++ 所作的重大贡献。

以下各位为本书各版本的草稿提供了许多有益的建议：Paul Abrahams、Michael Ball、Stephen Edwards、Cay Horstmann、Brian Kernighan、Tom Lyons、Robert Murray、Ed Scheibel、Roy Turner 和 Jon Wada，尤其要感谢 Michael Ball 富于思想的建议和鼓励。特别感谢 Clovis Tondo 和 Bruce Leung 为本书所做的深入审

阅。

Stanley 想特别感谢 Shyh-Chyuan Huang 和 Jinko Gotoh, 感谢他们对 Firebird 特效制作给予的帮助和支持。当然还有 Jon Wada 和 Josée。

Josée 要感谢 Gabby Silberman、Karen Bennet 以及 IBM 高级研究中心其他小组成员, 感谢他们为本书写作的支持。最大的感谢要给予 Stanley, 感谢他带着她经历了这项伟大的冒险活动。

最后我们两人都要感谢编辑们的辛苦工作以及巨大的耐心: Debbie Lafferty (从一开始就为本书操劳)、Mike Hendrickson 以及 John Fuller。Big Purple 公司的排版工作做得非常漂亮。6.1 节的插图是 Elena Driskill 制作的。非常感谢 Elena 允许我们使用它。

第 2 版的致谢

这本书是无数看不见的手帮助作者的结果。最由衷地感谢 Barbara Moo, 她的鼓励和建议以及对原稿无数遍的仔细审读是无价的。特别感谢 Bjarne Stroustrup 持续不断的帮助和鼓励, 以及他给予我们如此美妙的编程语言。还有 Stephen Dewhurst, 他在我第一次学习 C++ 时给了许多支持。还有 Nancy Wilkinson, 另一位 cfront 的编写者和 Gummi Bears 的提供者。

Dag Brück、Martin Carroll、William Hopkins、Brian Kernighan、Andrew Koenig、Alexis Layton 以及 Barbara Moo 提供了特别详细和敏锐的建议, 他们的评论大大完善了这本书。Andy Baily、Phil Brown、James Coplien、Elizabeth Flanagan、David Jordan、Don Kretsch、Craig Rubin、Jonathan Shopiro、Judy Ward、Nancy Wilkinson 以及 Clay Wilson 审阅了本书各版本的草稿, 提出了许多有益的建议。David Prosser 澄清了 ANSI C 的许多问题。Jerry Schwarz 实现了 iostream 包, 他提供了附录 A 所依据的原始文档。(在第 3 版中变成了第 20 章!) 非常感谢他对附录的细致审阅, 感谢 3.0 版本开发小组的其他成员: Laura Eaves、George Logothetis、Judy Ward 和 Nancy Wilkinson。

以下各位为 Addison-Wesley 审阅了书稿: James Adcock、Steven Bellovin、Jon Forrest、Maurice Herlihy、Norman Kerth、Darrell Long、Victor Milenkovic 以及 Justin

Smith。

以下各位指出了第 1 版的各种印刷错误: David Beckedorff、Dag Brück、John Eldridge、Jim Humelsine、Dave Jordan、Ami Kleinman、Andrew Koenig、Tim O'Konski、Clovis Tondo 和 Steve Vinoski。

非常感谢 Brian Kernighan 和 Andrew Koenig 提供了大量的排版工具。

参考文献

以下书籍分为两部分,有些是影响本书写作的文献,有些是值得推荐的 C++ 重要文献。

- [BOOCH94] Booch, Grady, *Object-Oriented Analysis and Design*, Benjamin/Cummings, Redwood City, CA (1994) ISBN 0-8053-5340-2.
- [GAMMA95] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns*, Addison Wesley Longman, Inc., Reading, MA (1995) ISBN 0-201-63361-2.
- [GHEZZI97] Ghezzi, Carlo, and Mehdi Jazayeri, *Programming Language Concepts, 3rd Edition*, John Wiley and Sons, New York, NY (1997) ISBN 0-471-10426-4.
- [HARBISON88] Samuel P. Harbison and Guy L. Steele, Jr, *C: A Reference Manual, 3rd Edition*, Prentice-Hall, Englewood Cliffs, NJ (1988) ISBN 0-13-110933-2.
- [ISO-C++97] Draft Proposed International Standard for Information Systems — Programming Language C++ - Final Draft (FDIS) 14882.
- [KERNIGHAN88] Kernighan, Brian W., and Dennis M. Ritchie, *The C Programming Language*, Prentice-Hall, Englewood Cliffs, NJ (1988) ISBN 0-13-110362-8.
- [KOENIG97] Koenig, Andrew, and Barbara Moo, *Ruminations on C++*, Addison Wesley Longman, Inc., Reading, MA (1997) ISBN 0-201-42339-1.
- [LIPPMAN91] Lippman, Stanley, *C++ Primer, 2nd Edition*, Addison Wesley Longman, Inc., Reading, MA (1991) ISBN 0-201-54848-8.
- [LIPPMAN96a] Lippman, Stanley, *Inside the C++ Object Model*, Addison Wesley Longman, Inc., Reading, MA (1996) ISBN 0-201-83454-5.
- [LIPPMAN96b] Lippman, Stanley, Editor, *C++ Gems*, a SIGS Books imprint, Cambridge University Press, Cambridge, England (1996) ISBN 0-13570581-9.

- [MEYERS98] Meyers, Scott, *Effective C++, 2nd Edition*, Addison Wesley Longman, Inc., Reading, MA (1998) ISBN 0-201-92488-9.
- [MEYERS96] Meyers, Scott, *More Effective C++*, Addison Wesley Longman, Inc., Reading, MA (1996) ISBN 0-201-63371-X.
- [MURRAY93] Murray, Robert B., *C++ Strategies and Tactics*, Addison Wesley Longman, Inc., Reading, MA (1993) ISBN 0-201-56382-7.
- [MUSSER96] Musser, David R., and Atul Saini, *STL Tutorial and Reference Guide*, Addison Wesley Longman, Inc., Reading, MA (1996) ISBN 0-201-63398-1.
- [NACKMAN94] Barton, John J., and Lee R. Nackman, *Scientific and Engineering C++, An Introduction with Advanced Techniques and Examples*, Addison Wesley Longman, Inc., Reading, MA (1994) ISBN 0-201-53393-6.
- [NEIDER93] Neider, Jackie, Tom Davis, and Mason Woo, *OpenGL Programming Guide*, Addison Wesley, Inc., Reading, MA (1993) ISBN 0-201-63274-8.
- [PERSON68] Person, Russell V., *Essentials of Mathematics, 2nd Edition*, John Wiley & Sons, Inc., New York, NY (1968) ISBN 0-132-84191-6.
- [PLAUGER92] Plauger, P.J., *The Standard C Library*, Prentice-Hall, Englewood Cliffs, NJ (1992) ISBN 0-13-131509-9.
- [SEDEGEWICK88] Sedgewick, Robert, *Algorithms, 2nd Edition*, Addison Wesley Longman, Inc., Reading, MA (1988) ISBN 0-201-06673-4.
- [SHAMPINE97] Shampine, L.F., R.C. Allen, Jr., and S. Pruess, *Fundamentals of Numerical Computing*, John Wiley & Sons, Inc., New York, NY (1997) ISBN 0-471-16363-5.
- [STROUSTRUP94] Stroustrup, Bjarne, *The Design and Evolution of C++*, Addison Wesley Longman, Inc., Reading, MA (1994) ISBN 0-201-54330-3.
- [STROUSTRUP97] Stroustrup, Bjarne, *The C++ Programming Language, 3rd Edition*, Addison Wesley Longman, Inc., Reading, MA (1997) ISBN 0-201-88954-4.
- [UPSTILL90] Upstill, Steve, *The RenderMan Companion*, Addison Wesley Longman, Inc., Reading, MA (1990) ISBN 0-201-50868-0.
- [WERNECKE94] Wernecke, Josie, *The Inventor Mentor*, Addison Wesley Longman, Inc., Reading, MA (1994) ISBN 0-201-62495-8.
- [YOUNG95] Young, Douglas A., *Object-Oriented Programming with C++ and OSF/Motif, 2nd Edition*, Prentice-Hall, Englewood Cliffs, NJ (1995) ISBN 0-132-09255-7.

Contents

Part I: C++, An Overview	1
<i>Chapter 1: Getting Started</i>	5
1.1: Problem Solving	5
1.2: The C++ Program	6
1.3: Preprocessor Directives	13
1.4: A Word About Comments	17
1.5: A First Look at Input/Output	19
<i>Chapter 2: A Tour of C++</i>	23
2.1: The Built-In Array Data Type	23
2.2: Dynamic Memory Allocation and Pointers	26
2.3: An Object-Based Design	30
2.4: An Object-Oriented Design	41
2.5: A Generic Design	51
2.6: An Exception-Based Design	58
2.7: An Array by Any Other Name	62
2.8: The Standard Array Is a Vector	68
Part II: The Basic Language	73
<i>Chapter 3: The C++ Data Types</i>	75
3.1: Literal Constant	75
3.2: Variables	79
3.3: Pointer Types	88
3.4: String Types	93
3.5: const Qualifier	102
3.6: Reference Types	105
3.7: The bool Type	110
3.8: Enumeration Types	111
3.9: Array Types	114
3.10: The vector Container Type	121
3.11: complex Number Types	125
3.12: Typedef Names	126
3.13: volatile Qualifier	127
3.14: The pair Type	128
3.15: Class Types	129

<i>Chapter 4: Expressions</i>	141
4.1: What Is an Expression?	141
4.2: Arithmetic Operators	143
4.3: Equality, Relational, and Logical Operators	146
4.4: Assignment Operators	149
4.5: Increment and Decrement Operators	154
4.6: Complex Number Operations	155
4.7: The Conditional Operator	159
4.8: The sizeof Operator	160
4.9: The new and delete Expressions	162
4.10: Comma Operator	164
4.11: The Bitwise Operators	164
4.12: bitset Operations	168
4.13: Precedence	172
4.14: Type Conversions	175
4.15: A Stack Class Example	185
<i>Chapter 5: Statements</i>	189
5.1: Simple and Compound Statements	189
5.2: Declaration Statement	191
5.3: The if Statement	194
5.4: The switch Statement	202
5.5: The for Loop Statement	210
5.6: The while Statement	214
5.7: The do while Statement	216
5.8: The break Statement	218
5.9: The continue Statement	220
5.10: The goto Statement	220
5.11: A Linked List Example	222
<i>Chapter 6: Abstract Container Types</i>	249
6.1: Our Text Query System	250
6.2: A vector or a list?	254
6.3: How a vector Grows Itself	256
6.4: Defining a Sequence Container	260
6.5: Iterators	265
6.6: Sequence Container Operations	269
6.7: Storing Lines of Text	273
6.8: Finding a Substring	276
6.9: Handling Punctuation	282
6.10: A String by Any Other Format	285
6.11: Additional String Operations	288