

畅销全球的经典 C++ 教程

SAMS

# C++ Primer Plus (第五版) 中文版

[美] Stephen Prata 著  
孙建春 韦强 译

 人民邮电出版社  
POSTS & TELECOM PRESS

# C++ Primer Plus (第五版) 中文版

[美] Stephen Prata 著

孙建春 韦 强 译

人民邮电出版社

## 图书在版编目 (CIP) 数据

C++ Primer Plus: 第5版/ (美) 普拉塔 (Prata, S.) 著; 孙建春, 韦强译.  
—北京: 人民邮电出版社, 2005.5

ISBN 7-115-13416-2

I. C... II. ①普...②孙...③韦... III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2005) 第 042689 号

## 版权声明

Stephen Prata: C++ Primer Plus, Fifth Edition (ISBN: 0672326973)

Copyright © 2005 by Sams Publishing.

Authorized translation from the English language edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

## C++Primer Plus (第五版) 中文版

- 
- ◆ 著 [美] Stephen Prata
  - 译 孙建春 韦强
  - 责任编辑 李际
  - ◆ 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>
  - 读者热线 010-67132705
  - 北京顺义振华印刷厂印刷
  - 新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16
  - 印张: 50.25
  - 字数: 1 639 千字 2005年5月第1版
  - 印数: 1-5 000 册 2005年5月北京第1次印刷

著作权合同登记号 图字: 01-2004-4418 号

ISBN 7-115-13416-2/TP · 4665

定价: 72.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

## 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、通用编程和传统的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。

全书分 17 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、分支语句和逻辑操作符、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/输出等内容。

本书针对 C++初学者，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，因此不要求读者有较多 C 语言方面的背景知识。本书可作为高等院校 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，在加州大学伯克利分校获得博士学位。Stephen 本人或与他人合作编写的图书有十多本。他撰写的 *The Waite Group's New C Primer Plus* 一书获得了计算机出版联合会 1990 年度最佳 “How-to” 计算机图书奖；他撰写的 *The Waite Group's C++ Primer Plus* 一书获得了计算机出版联合会 1991 年度最佳 “How-to” 计算机图书奖的提名。

# 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模式，其中包括面向对象编程、通用编程和传统的过程化编程。随着新特性的不断添加，C++ 一度成为一个活动目标，不过现在有了 2003 年的 ISO/ANSI C++ 标准第二版后，已经稳定下来了。现代编译器支持该标准要求的多数或全部特性，程序员要花费时间来习惯这些特性的应用。本书第五版是按 ISO/ANSI 标准编写的，将介绍这种成熟的 C++ 版本。

本书在介绍 C++ 特性的同时，讨论了基本 C 语言，使二者成为有机的整体。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。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，还讲述了 C++ 在 C 语言的基础上新增的许多特性，其中包括：

- 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。
- 通用（或独立于类型的）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。

本书在传授知识方面有几个优点。大约 20 年前，《C Primer Plus》一书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。
- 初级教程不要求读者已经熟悉相关的编程概念。
- 初级教程强调的是“实践性”学习，通过简短、容易输入的范例同时阐述一个或多个概念。
- 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读者对知识的理解，从而适于自学或课堂教学。

基于上述理念，本书帮助读者理解这种用途广泛的语言，并学习如何使用它。例如：

- 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，读者阅读本书后，能够编写出可靠、高效的程序，并且觉得这是一种享受。

## 本书使用的范例代码

本书包含大量的范例代码，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。书中的范例在 Windows XP 系统、Macintosh OS X 系统和 Linux 系统上进行了测试。只有为数不多的几个程序会受编译器不兼容问题的影响。本书前一版面世后，编译器在遵循 C++ 标准方面更严格。

对于本书中完整的程序，其源代码可从 Sams 网站（[www.sampublishing.com](http://www.sampublishing.com)）下载得到。为此，可输入本书的英文书名，然后单击 Search，再单击该书名，切换到能够下载源代码的网页。在该网站，也可找到有些编程练习的解决方案。

## 本书的组织结构

本书分为 17 章和 10 个附录。

### 第 1 章：预备知识

本章介绍了 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++ 编程语言的。讨论了过程化语言（如 C 语言）与面向对象语言（如 C++）之间的区别。读者将了解 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。读者可以学习到 `main()` 函数扮演的角色以及 C++ 程序使用的一些语句。读者将使用预定义的 `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种类型。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++ 允许程序员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 9~13 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）。读者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` 操作符。

### 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++ 的关系操作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，读者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：分支语句和逻辑操作符

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，读者将了解到如何使用 `if`、`if else` 和 `switch` 语句及条件操作符来控制程序流程，学习如何使用逻辑操作符来表达决策测试。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，读者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。读者将学习内联函数，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，读者还将学习函数模板，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，读者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，如堆栈。

## 第 11 章：使用类

在本章中，读者将深入了解类。首先了解操作符重载，它使程序员能够定义与类对象一起使用的操作符，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操作符成员函数是如何被用来管理类的类型转换的。

## 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，定义显式复制构造函数和显式赋值操作符。本章介绍了在程序员没有提供显式定义时，如何将如何隐式地生成成员函数以及这些成员函数的行为。读者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，读者还将学习类模板，它使程序员能够使用通用类型来定义类，然后使用模板来根据具体类型创建特定的类。例如，堆栈模板使程序员能够创建整数堆栈或字符串堆栈。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。读者还将学习 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

本章将讨论 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。它还提供了高效的通用算法库，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。读者将要学习如何使用类方法来确定输入或输出流的状态，了解输入类型是否不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。读者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。

## 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: 操作符优先级**

本附录按优先级从高到低的顺序列出了 C++ 的操作符。

**附录 E: 其他操作符**

本附录总结了正文中没有介绍的其他 C++ 操作符, 如按位操作符等。

**附录 F: string 模板类**

本附录总结了 string 类方法和函数。

**附录 G: STL 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: 精选读物和网资源**

本附录列出一些参考书, 帮助读者深入了解 C++。

**附录 I: 转换为 ANSI/ISO 标准 C++**

本附录提供了从 C 和老式 C++ 实现到 ANSI/ISO C++ 的转换指南。

**附录 J: 复习题答案**

本附录提供每章结尾的复习题的答案。

**对教师的提示**

本书第五版的宗旨之一是: 提供一本既可以用作自学书也可以用作教材的书籍。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++, 不依赖于特定的实现。
- 本书内容记录了 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 预备知识的复习章节一带而过。
- 各章都有复习题和编程练习。附录 J 提供了复习题的答案; 有些编程练习的解决方案可在 Sams 网站 ([www.sampublishing.com](http://www.sampublishing.com)) 上找到。
- 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)、堆栈、队列、简单链表、模拟、通用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 <<
    return 0;
}
```

- 用户需要输入的程序输入用粗体表示:

Please enter your name:

**Plato**

- 句法描述中的占位符用斜体表示。读者应使用实际的文件名、参数等替换占位符。

## 开发书中编程范例时使用的系统

本书的范例都是使用 Microsoft Visual C++ 7.1 (Microsoft Visual Studio .NET 2003 中的版本) 和 Metrowerks CodeWarrior Development Studio 9 在一台带硬盘的运行 Windows XP Professional 的 Pentium PC 上开发的。对于其中的多数程序, 都在上述系统上使用 Borland C++ 5.5 命令行编译器和 GNU gpp 3.3.3, 在运行 SuSE 9.0 Linux 的 IBM 兼容 Pentium 机上使用 Comeau 4.3.3 和 GNU g++3.3.1, 在运行 OS 10.3 的 Macintosh G4 上使用 Metrowerks Development Studio 9 进行了测试。本书指出了由于滞后于标准而导致的差异, 如“老式实现使用 `ios::fixed`, 而不是 `ios_base::fixed`”。本书指出了老式编译器中的错误和令人厌烦或容易混淆的特性, 不过, 它们都将在最新版本中得到了解决。

C++ 为程序员提供了丰富多彩的内容。祝您学习愉快!

# 目 录

<b>第 1 章 预备知识</b> .....1	<b>第 3 章 处理数据</b> .....36
1.1 C++简介.....1	3.1 简单变量.....36
1.2 C++简史.....2	3.1.1 变量名.....37
1.2.1 C 语言.....2	3.1.2 整型.....38
1.2.2 C 语言编程原理.....2	3.1.3 short、int 和 long.....38
1.2.3 面向对象编程.....3	3.1.4 无符号类型.....41
1.2.4 C++和通用编程.....4	3.1.5 选择整型类型.....43
1.2.5 C++的起源.....4	3.1.6 整型常量.....44
1.3 可移植性和标准.....5	3.1.7 C++如何确定常量的类型.....45
1.4 程序创建的技巧.....6	3.1.8 char 类型：字符和小整数.....45
1.4.1 创建源代码文件.....7	3.1.9 新的 bool 类型.....50
1.4.2 编译和链接.....8	3.2 const 限定符.....51
1.5 总结.....11	3.3 浮点数.....51
<b>第 2 章 开始学习 C++</b> .....12	3.3.1 书写浮点数.....52
2.1 进入 C++.....12	3.3.2 浮点类型.....53
2.1.1 main() 函数.....14	3.3.3 浮点常量.....54
2.1.2 C++注释.....15	3.3.4 浮点数的优缺点.....55
2.1.3 C++预处理器和 iostream 文件.....16	3.4 C++算术操作符.....55
2.1.4 头文件名.....16	3.4.1 操作符优先级和结合性.....56
2.1.5 名称空间.....17	3.4.2 除法分支.....57
2.1.6 使用 cout 进行 C++输出.....18	3.4.3 求模操作符.....58
2.1.7 C++源代码的格式化.....19	3.4.4 类型转换.....59
2.2 C++语句.....21	3.5 总结.....62
2.2.1 声明语句和变量.....21	3.6 复习题.....63
2.2.2 赋值语句.....22	3.7 编程练习.....63
2.2.3 cout 的新花样.....23	<b>第 4 章 复合类型</b> .....65
2.3 其他 C++语句.....23	4.1 数组.....65
2.3.1 使用 cin.....24	4.2 字符串.....68
2.3.2 使用 cout 进行拼接.....24	4.2.1 拼接字符串常量.....70
2.3.3 类简介.....25	4.2.2 在数组中使用字符串.....70
2.4 函数.....26	4.2.3 字符串输入.....71
2.4.1 使用有返回值的函数.....26	4.2.4 每次读取一行字符串输入.....72
2.4.2 函数变体.....28	4.2.5 混合输入字符串和数字.....75
2.4.3 用户定义的函数.....29	4.3 string 类简介.....76
2.4.4 用户定义的有返回值的函数.....31	4.3.1 赋值、拼接和附加.....77
2.4.5 在多函数程序中使用 using 编译指令.....32	4.3.2 string 类的其他操作.....78
2.5 总结.....33	4.3.3 string 类 I/O.....79
2.6 复习题.....34	4.4 结构简介.....81
2.7 编程练习.....34	4.4.1 在程序中使用结构.....82
	4.4.2 结构可以将 string 类作为成员吗.....84
	4.4.3 其他结构属性.....84

4.4.4	结构数组	85	5.5.2	使用 <code>cin.get(char)</code> 进行补救	137
4.4.5	结构中的位字段	86	5.5.3	使用哪一个 <code>cin.get()</code>	138
4.5	共用体	86	5.5.4	文件尾条件	138
4.6	枚举	88	5.5.5	另一个 <code>cin.get()</code> 版本	140
4.6.1	设置枚举量的值	89	5.6	嵌套循环和二维数组	142
4.6.2	枚举的取值范围	89	5.7	总结	145
4.7	指针和自由存储空间	89	5.8	复习题	145
4.7.1	声明和初始化指针	92	5.9	编程练习	146
4.7.2	指针的危险	93	<b>第 6 章 分支语句和逻辑操作符</b>		148
4.7.3	指针和数字	93	6.1	if 语句	148
4.7.4	使用 <code>new</code> 来分配内存	94	6.1.1	if else 语句	149
4.7.5	使用 <code>delete</code> 来释放内存	95	6.1.2	格式化 if else 语句	151
4.7.6	使用 <code>new</code> 来创建动态数组	96	6.1.3	if else if else 结构	152
4.8	指针、数组和指针算术	98	6.2	逻辑表达式	153
4.8.1	程序说明	99	6.2.1	逻辑 OR 操作符: <code>  </code>	153
4.8.2	指针和字符串	101	6.2.2	逻辑 AND 操作符: <code>&amp;&amp;</code>	154
4.8.3	使用 <code>new</code> 创建动态结构	104	6.2.3	逻辑 NOT 操作符: <code>!</code>	157
4.8.4	自动存储、静态存储和动态存储	107	6.2.4	逻辑操作符细节	158
4.9	总结	108	6.2.5	其他表示方式	159
4.10	复习题	109	6.3	字符函数库 <code>cctype</code>	159
4.11	编程练习	109	6.4	? 操作符	161
<b>第 5 章 循环和关系表达式</b>		111	6.5	switch 语句	162
5.1	for 循环	111	6.5.1	将枚举量用作标签	165
5.1.1	for 循环的组成部分	112	6.5.2	switch 和 if else	166
5.1.2	回到 for 循环	116	6.6	break 和 continue 语句	166
5.1.3	修改步长	118	6.7	读取数字的循环	168
5.1.4	使用 for 循环访问字符串	118	6.8	简单文件输入/输出	171
5.1.5	递增操作符 ( <code>++</code> ) 和递减操作符 ( <code>--</code> )	119	6.8.1	文本 I/O 和文本文件	171
5.1.6	副作用和顺序点	119	6.8.2	写入到文本文件中	172
5.1.7	前缀格式和后缀格式	120	6.8.3	读取文本文件	174
5.1.8	递增/递减操作符和指针	120	6.9	总结	178
5.1.9	组合赋值操作符	121	6.10	复习题	178
5.1.10	复合语句 (语句块)	122	6.11	编程练习	180
5.1.11	逗号操作符 (或其他句法技巧)	123	<b>第 7 章 函数——C++的编程模块</b>		182
5.2	关系表达式	125	7.1	复习函数的基本知识	182
5.2.1	可能犯的错误	126	7.1.1	定义函数	183
5.2.2	C-风格字符串的比较	127	7.1.2	函数原型和函数调用	184
5.2.3	比较 <code>string</code> 类字符串	129	7.2	函数参数和按值传递	187
5.3	while 循环	130	7.2.1	多个参数	188
5.3.1	for 与 while	131	7.2.2	另外一个接受两个参数的函数	189
5.3.2	等待一段时间: 编写延时循环	132	7.3	函数和数组	191
5.4	do while 循环	134	7.3.1	函数如何使用指针来处理数组	192
5.5	循环和文本输入	136	7.3.2	将数组作为参数意味着什么	192
5.5.1	使用原始的 <code>cin</code> 进行输入	136	7.3.3	更多数组函数范例	194
			7.3.4	使用数组区间的函数	198
			7.3.5	指针和 <code>const</code>	200

7.4	函数和二维数组	202	9.2.3	静态持续变量	270
7.5	函数和 C-风格字符串	203	9.2.4	说明符和限定符	277
7.5.1	将 C-风格字符串作为参数的函数	203	9.2.5	函数和链接性	278
7.5.2	返回 C-风格字符串的函数	204	9.2.6	语言链接性	279
7.6	函数和结构	206	9.2.7	存储方案和动态分配	279
7.6.1	传递和返回结构	206	9.3	布局 new 操作符	280
7.6.2	另一个处理结构的函数范例	207	9.4	名称空间	282
7.6.3	传递结构的地址	211	9.4.1	传统的 C++ 名称空间	282
7.7	函数和 string 对象	212	9.4.2	新的名称空间特性	284
7.8	递归	214	9.4.3	名称空间范例	289
7.8.1	包含一个递归调用的递归	214	9.4.4	名称空间及其前途	292
7.8.2	包含多个递归调用的递归	215	9.5	总结	292
7.9	函数指针	216	9.6	复习题	293
7.9.1	函数指针的基础知识	216	9.7	编程练习	295
7.9.2	函数指针示例	218	<b>第 10 章 对象和类</b>		297
7.10	总结	219	10.1	过程性编程和面向对象编程	297
7.11	复习题	220	10.2	抽象和类	298
7.12	编程练习	220	10.2.1	类型是什么	298
<b>第 8 章 函数探幽</b>		223	10.2.2	C++ 中的类	299
8.1	C++ 内联函数	223	10.2.3	实现类成员函数	302
8.2	引用变量	225	10.2.4	使用类	306
8.2.1	创建引用变量	225	10.2.5	小结	308
8.2.2	将引用用作函数参数	227	10.3	类的构造函数和析构函数	309
8.2.3	引用的属性和特别之处	230	10.3.1	声明和定义构造函数	309
8.2.4	将引用用于结构	232	10.3.2	使用构造函数	310
8.2.5	将引用用于类对象	235	10.3.3	默认构造函数	311
8.2.6	对象、继承和引用	238	10.3.4	析构函数	311
8.2.7	何时使用引用参数	240	10.3.5	改进 Stock 类	312
8.3	默认参数	240	10.3.6	构造函数和析构函数小结	317
8.4	函数重载	242	10.4	this 指针	318
8.4.1	重载范例	243	10.5	对象数组	322
8.4.2	何时使用函数重载	245	10.6	接口和实现小结	325
8.5	函数模板	246	10.7	类作用域	326
8.5.1	重载的模板	248	10.8	抽象数据类型	327
8.5.2	显式具体化	250	10.9	总结	331
8.5.3	实例化和具体化	253	10.10	复习题	332
8.5.4	编译器选择使用哪个函数版本	254	10.11	编程练习	332
8.6	总结	258	<b>第 11 章 使用类</b>		335
8.7	复习题	258	11.1	操作符重载	335
8.8	编程练习	259	11.2	计算时间：一个操作符重载范例	336
<b>第 9 章 内存模型和名称空间</b>		261	11.2.1	添加加法操作符	339
9.1	单独编译	261	11.2.2	重载限制	341
9.2	存储持续性、作用域和链接性	265	11.2.3	其他重载操作符	342
9.2.1	作用域和链接	265	11.3	友元简介	345
9.2.2	自动存储持续性	266	11.3.1	创建友元	346
			11.3.2	常用的友元：重载 << 操作符	347

11.4 重载操作符: 作为成员函数还是非成员函数	351	13.6 抽象基类	454
11.5 再谈重载: 矢量类	352	13.6.1 应用 ABC 概念	456
11.5.1 使用状态成员	358	13.6.2 ABC 理念	459
11.5.2 为 Vector 类重载算术操作符	359	13.7 继承和动态内存分配	459
11.5.3 对实现的说明	361	13.7.1 第一种情况: 派生类不使用 new	460
11.5.4 使用 Vector 类来模拟随机行走	361	13.7.2 第二种情况: 派生类使用 new	460
11.6 类的自动转换和强制类型转换	363	13.7.3 使用动态内存分配和友元的继承范例	462
11.6.1 程序说明	367	13.8 类设计回顾	466
11.6.2 转换函数	367	13.8.1 编译器生成的成员函数	466
11.6.3 转换函数和友元函数	371	13.8.2 其他的类方法	467
11.7 总结	373	13.8.3 公有继承的考虑因素	469
11.8 复习题	374	13.8.4 类函数小结	472
11.9 编程练习	375	13.9 总结	472
<b>第 12 章 类和动态内存分配</b>	<b>377</b>	13.10 复习题	473
12.1 动态内存和类	377	13.11 编程练习	474
12.1.1 复习范例和静态类成员	378	<b>第 14 章 C++ 中的代码重用</b>	<b>477</b>
12.1.2 隐式成员函数	383	14.1 包含对象成员类	477
12.1.3 改进后的新 String 类	389	14.1.1 valarray 类简介	478
12.1.4 在构造函数中使用 new 时应注意的事项	397	14.1.2 Student 类的设计	478
12.1.5 有关返回对象的说明	399	14.1.3 Student 类范例	479
12.1.6 使用指向对象的指针	401	14.2 私有继承	485
12.1.7 复习各种技术	408	14.3 多重继承	492
12.2 队列模拟	409	14.3.1 有多少 Worker	496
12.2.1 队列类	410	14.3.2 哪个方法	499
12.2.2 Customer 类	417	14.3.3 MI 小结	507
12.2.3 模拟	420	14.4 类模板	508
12.3 总结	423	14.4.1 定义类模板	508
12.4 复习题	424	14.4.2 使用模板类	510
12.5 编程练习	425	14.4.3 深入探讨模板类	512
<b>第 13 章 类继承</b>	<b>428</b>	14.4.4 数组模板范例和非类型参数	517
13.1 一个简单的基类	429	14.4.5 模板多功能性	518
13.1.1 派生一个类	430	14.4.6 模板的具体化	521
13.1.2 构造函数: 访问权限的考虑	431	14.4.7 成员模板	523
13.1.3 使用派生类	433	14.4.8 将模板用作参数	525
13.2 派生类和基类之间的特殊关系	435	14.4.9 模板类和友元	526
13.3 继承——is-a 关系	436	14.5 总结	532
13.4 多态公有继承	438	14.6 复习题	533
13.4.1 开发 Brass 类和 BrassPlus 类	438	14.7 编程练习	534
13.4.2 静态联编和动态联编	447	<b>第 15 章 友元、异常和其他</b>	<b>539</b>
13.4.3 指针和引用类型兼容性	448	15.1 友元	539
13.4.4 虚拟成员函数和动态联编	449	15.1.1 友元类	539
13.4.5 有关虚函数注意事项	450	15.1.2 友元成员函数	543
13.5 访问控制: protected	452	15.1.3 其他友好关系	546

15.2 嵌套类	547	16.6.2 算法的通用特征	638
15.2.1 嵌套类和访问权限	548	16.6.3 STL 和 string 类	639
15.2.2 模板中的嵌套	549	16.6.4 函数和容器方法	640
15.3 异常	552	16.6.5 使用 STL	641
15.3.1 调用 abort()	552	16.7 其他库	644
15.3.2 返回错误码	553	16.8 总结	648
15.3.3 异常机制	554	16.9 复习题	649
15.3.4 将对象用作异常类型	557	16.10 编程练习	650
15.3.5 堆栈解退	560	<b>第 17 章 输入、输出和文件</b>	<b>651</b>
15.3.6 其他异常特性	564	17.1 C++ 输入和输出概述	651
15.3.7 exception 类	566	17.1.1 流和缓冲区	652
15.3.8 异常、类和继承	569	17.1.2 流、缓冲区和 iostream 文件	653
15.3.9 异常何时会迷失方向	573	17.1.3 重定向	655
15.3.10 有关异常的注意事项	575	17.2 使用 cout 进行输出	656
15.4 RTTI	576	17.2.1 重载的 << 操作符	656
15.4.1 RTTI 的用途	576	17.2.2 其他 ostream 方法	658
15.4.2 RTTI 的工作原理	576	17.2.3 刷新输出缓冲区	660
15.5 类型转换操作符	583	17.2.4 用 cout 进行格式化	660
15.6 总结	585	17.3 使用 cin 进行输入	672
15.7 复习题	586	17.3.1 cin>> 如何检查输入	673
15.8 编程练习	587	17.3.2 流状态	675
<b>第 16 章 string 类和标准模板库</b>	<b>588</b>	17.3.3 其他 istream 类方法	678
16.1 string 类	588	17.3.4 其他 istream 方法	683
16.1.1 构造字符串	588	17.4 文件输入和输出	686
16.1.2 string 类输入	591	17.4.1 简单的文件 I/O	686
16.1.3 使用字符串	593	17.4.2 流状态检查和 is_open()	688
16.1.4 string 还提供了哪些功能	597	17.4.3 打开多个文件	689
16.2 auto_ptr 类	599	17.4.4 命令行处理技术	689
16.2.1 使用 auto_ptr	599	17.4.5 文件模式	691
16.2.2 有关 auto_ptr 的注意事项	601	17.4.6 随机存取	698
16.3 STL	602	17.5 内核格式化	704
16.3.1 vector 模板类	602	17.6 接下来的任务	706
16.3.2 可对矢量执行的操作	604	17.7 总结	707
16.3.3 对矢量可执行的其他操作	608	17.8 复习题	707
16.4 通用编程技术	611	17.9 编程练习	708
16.4.1 为何使用迭代器	611	<b>附录 A 计数系统</b>	<b>712</b>
16.4.2 迭代器类型	614	A.1 十进制数	712
16.4.3 迭代器层次结构	615	A.2 八进制整数	712
16.4.4 概念、改进和模型	616	A.3 十六进制数	712
16.4.5 容器种类	620	A.4 二进制数	713
16.4.6 联合容器	627	A.5 二进制和十六进制	714
16.5 函数对象	631	<b>附录 B C++ 保留字</b>	<b>715</b>
16.5.1 函数符概念	632	B.1 C++ 关键字	715
16.5.2 预定义的函数符	634	B.2 替代标记	715
16.5.3 自适应函数符和函数适配器	635		
16.6 算法	637		
16.6.1 算法组	637		

B.3 C++库保留名称.....	716	G.1 所有容器都有的成员.....	744
<b>附录 C ASCII 字符集</b> .....	717	G.2 矢量、链表和双端队列的其他成员.....	746
<b>附录 D 操作符优先级</b> .....	721	G.3 set 和 map 的其他成员.....	747
<b>附录 E 其他操作符</b> .....	724	G.4 STL 函数.....	748
E.1 按位操作符.....	724	G.4.1 非修改式序列操作.....	749
E.1.1 移位操作符.....	724	G.4.2 修改式序列操作.....	751
E.1.2 逻辑按位操作符.....	725	G.4.3 排序和相关操作.....	756
E.1.3 按位操作符的另一种表示.....	727	G.4.4 数字操作.....	763
E.1.4 几种常用的按位操作符技术.....	727	<b>附录 H 精选读物和网上资源</b> .....	765
E.2 成员解除引用操作符.....	728	H.1 精选读物.....	765
<b>附录 F string 模板类</b> .....	732	H.2 网上资源.....	766
F.1 13 种类型和一个常量.....	732	<b>附录 I 转换为 ANSI/ISO 标准 C++</b> .....	767
F.2 数据信息、构造函数及其他.....	733	I.1 使用一些预处理器编译指令的替代品.....	767
F.2.1 默认构造函数.....	734	I.1.1 使用 const 而不是#define 来	
F.2.2 使用数组的构造函数.....	735	定义常量.....	767
F.2.3 使用部分数组的构造函数.....	735	I.1.2 使用 inline 而不是# define 来	
F.2.4 复制构造函数.....	735	定义小型函数.....	768
F.2.5 使用一个字符的 <i>n</i> 个副本的		I.2 使用函数原型.....	769
构造函数.....	736	I.3 使用类型转换.....	769
F.2.6 使用区间的构造函数.....	736	I.4 逐渐熟悉 C++ 特性.....	769
F.2.7 内存杂记.....	736	I.5 使用新的头文件.....	769
F.3 字符串存取.....	737	I.6 使用名称空间.....	770
F.4 基本赋值.....	737	I.7 使用 auto_ptr 模板.....	771
F.5 字符串搜索.....	738	I.8 使用 string 类.....	771
F.5.1 find() 系列.....	738	I.9 使用 STL.....	771
F.5.2 rfind() 系列.....	738	<b>附录 J 复习题答案</b> .....	772
F.5.3 find_first_of() 系列.....	738	第 2 章复习题答案.....	772
F.5.4 find_last_of() 系列.....	739	第 3 章复习题答案.....	772
F.5.5 find_first_not_of() 系列.....	739	第 4 章复习题答案.....	773
F.5.6 find_last_not_of() 系列.....	739	第 5 章复习题答案.....	774
F.6 比较方法和函数.....	740	第 6 章复习题答案.....	774
F.7 字符串修改方法.....	741	第 7 章复习题答案.....	775
F.7.1 用于追加和相加的方法.....	741	第 8 章复习题答案.....	777
F.7.2 其他赋值方法.....	741	第 9 章复习题答案.....	778
F.7.3 插入方法.....	742	第 10 章复习题答案.....	779
F.7.4 清除方法.....	742	第 11 章复习题答案.....	780
F.7.5 替换方法.....	742	第 12 章复习题答案.....	781
F.7.6 其他修改方法: copy() 和		第 13 章复习题答案.....	783
swap().....	743	第 14 章复习题答案.....	784
F.8 输出和输入.....	743	第 15 章复习题答案.....	785
<b>附录 G STL 方法和函数</b> .....	744	第 16 章复习题答案.....	786
		第 17 章复习题答案.....	786



# 第 1 章 预备知识

## 本章内容包括:

- C 语言和 C++的发展历史和基本原理。
- 过程性编程技术和面向对象编程技术。
- C++是如何在 C 语言的基础上添加面向对象概念的。
- C++是如何在 C 语言的基础上添加通用编程概念的。
- 编程语言标准。
- 创建程序的技巧。

欢迎进入 C++ 世界! 这是一种令人兴奋的语言, 它在 C 语言的基础上添加了对面向对象编程的支持, 在 20 世纪 90 年代便是最重要的编程语言之一, 其应用在 21 世纪仍保持强劲势头。C++ 继承了 C 语言高效、简洁、快速和可移植性的传统。C++ 面向对象的特性带来了全新的编程方法, 这种方法是应付复杂程度不断提高的现代编程任务而设计的。C++ 的模板特性提供了另一种全新的编程方法——通用编程。这 3 件法宝既是福也是祸, 一方面让 C++ 语言功能强大, 一方面则意味着有更多的东西需要学习。

本章先介绍 C++ 的背景, 然后进一步介绍创建 C++ 程序的一些基本原则。本书其他章节将讲述如何使用 C++ 语言, 从语言最浅显的基本知识开始, 到面向对象的编程 (OOP) 及其支持的新术语——对象、类、封装、数据隐藏、多态和继承等, 然后介绍它对通用编程的支持 (当然, 随着您对 C++ 的学习, 这些词汇将从花里胡哨的词语变为论述中必不可少的术语)。

## 1.1 C++ 简介

C++ 融合了 3 种不同的编程传统——C 语言代表的过程性语言传统、C++ 在 C 语言基础上添加的类代表的面向对象语言的 tradition 以及 C++ 模板支持的通用编程传统。本章将简要介绍这些传统。不过首先, 我们来看看这种传统对于学习 C++ 来说意味着什么。使用 C++ 的原因之一是为了利用其面向对象的特性。要利用这种特性, 必须对标准 C 语言知识有较深入的了解, 因为它提供了基本类型、操作符、控制结构和句法规则。所以, 如果已经对 C 有所了解, 便可以学习 C++ 了, 但这并不仅仅是学习更多的关键字和结构, 从 C 过渡到 C++ 的学习量就像从头学习 C 语言一样大。另外, 如果先掌握了 C 语言, 则在过渡到 C++ 时, 必须摒弃一些编程习惯。如果不了解 C 语言, 则学习 C++ 时需要掌握 C 语言的知识、OOP 知识以及通用编程知识, 但无须摒弃任何编程习惯。如果您认为学习 C++ 可能需要扩展思维, 这就对了。本书将以清晰的、帮助的方式, 引导读者一步一个脚印地学习, 因此扩展思维的过程是温和的, 不至于让您的大脑受不了。

本书通过传授 C 语言基础知识和 C++ 新增的内容, 带您步入 C++ 的世界, 因此不要求读者具备 C 语言知识。首先学习 C++ 与 C 语言共有的一些特性。即使已经了解 C 语言, 也会发现阅读本书的这一部分是一次很好的复习。另外, 本章还介绍了一些对后面的学习十分重要的概念, 指出了 C++ 和 C 之间的区别。在牢固地掌握了 C 语言的基础知识后, 就可以在此基础上学习 C++ 方面的知识了。那时将学习对象和类以及 C++ 是如何实现它们的, 另外还将学习模板。