SAMS

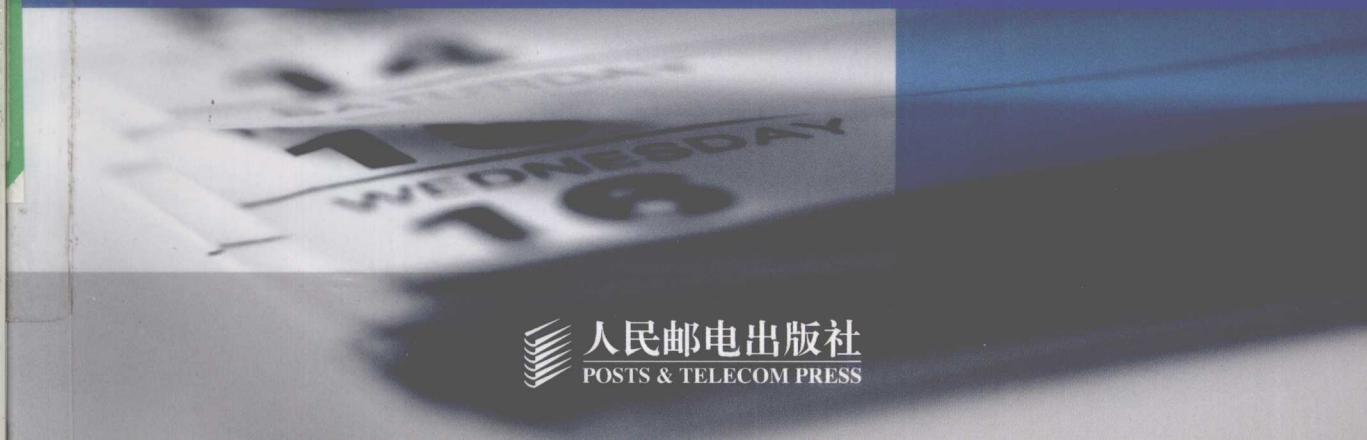
畅销全球的
经典C++教程

# 21天学通C++（第6版）

[美] Jesse Liberty
Siddhartha Rao
Bradley Jones 著

袁国忠 陈秋萍 译

全美销量超过25万册




人民邮电出版社
POSTS & TELECOM PRESS

# 21天学通C++（第6版）

Jesse Liberty
[美] Siddhartha Rao 著
Bradley Jones
袁国忠 陈秋萍 译

**图书在版编目（CIP）数据**

21天学通C++：第6版 /（美）利伯蒂（Liberty, J.），（美）拉奥（Rao, S.），（美）琼斯（Jones, B.）著；袁国忠，陈秋萍译. —北京：人民邮电出版社，2009.8
ISBN 978-7-115-20793-7

Ⅰ. 2… Ⅱ. ①利…②拉…③琼…④袁…⑤陈… Ⅲ. C语言—程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2009）第087275号

版 权 声 明

Jesse Liberty, Siddhartha Rao, Bradley Jones: Sams Teach Yourself C++ in One Hour a Day
ISBN: 978-0-672-32941-8
Copyright © 2009 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。

**21 天学通 C++（第 6 版）**

◆ 著　［美］Jesse Liberty　Siddhartha Rao
　　Bradley Jones
　译　袁国忠　陈秋萍
　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
　北京鸿佳印刷厂印刷

◆ 开本：787×1092　1/16
　印张：31.5
　字数：931 千字　2009 年 8 月第 1 版
　印数：1－4 000 册　2009 年 8 月北京第 1 次印刷

著作权合同登记号　图字：01-2008-3320 号
ISBN 978-7-115-20793-7/TP

定价：55.00 元

**读者服务热线：(010)67132705　印装质量热线：(010)67129223**
**反盗版热线：(010)67171154**

# 内容提要

本书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，包括管理输入/输出、循环和数组、面向对象编程、模板、使用标准模板库以及创建C++应用程序等。这些内容被组织成结构合理、联系紧密的章节，每章都可在1小时内阅读完毕；每章都提供了示例程序清单，并辅以示例输出和代码分析，以阐述该章介绍的主题。为加深读者对所学内容的理解，每章末尾都提供了常见问题及其答案以及练习和测验。读者可对照附录D提供的测验和练习答案，了解自己对所学内容的掌握程度。

本书是针对C++初学者编写的，不要求读者有C语言方面的背景知识，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

Siddhartha Rao 是一位微软 Visual C++ MVP，还是最活跃的 Internet 开发社区之一 CodeGuru 的主持人。他是 Windows 编程领域的专家，在架构设计以及使用 C++和其他现代编程语言开发驱动程序和应用程序方面拥有丰富的经验。当前，他为德国的一家软件巨人工作，致力于软件管理和软件开发最佳实践。鉴于在 3 个国家居住和生活过，他认为自己和家人得了旅行狂热症。Siddhartha 能说多种语言，闲暇期间他喜欢在全球各地旅行和摄影。

Jesse Liberty 编著了大量有关软件开发的图书，其中包括 C++和.NET 方面的畅销书。他是 Liberty Associates 公司的总裁，该公司致力于为客户提供编程、咨询和培训方面的服务。

Bradley Jones 是一位微软 Visual C++ MVP，他身兼网站管理员、经理、编码大师、执行编辑等职，其主要精力放在众多软件开发网站和频道上，其中包括 Developer.com、CodeGuru.com、DevX、VBForums、Gamelan 以及 Jupitermedia 的其他网站。

试读结束：需要全本请在线购买： www.ertongbook.com

# 前　言

本书旨在帮助读者学习如何使用 C++进行编程。就像人需要慢慢学会走路一样，学习 C++编程也需要循序渐进，因此本书每章包含的内容都可以在 1 小时内阅读完毕。本书通过实际使用 C++，帮助读者快速掌握编写实用的 C++应用程序涉及的最重要的概念。

通过每天学习 1 小时，读者将逐步掌握管理输入/输出、循环和数组、面向对象编程、模板、使用标准模板库以及创建 C++应用程序等基本知识，所有这些内容都组织成结构合理、易于理解的章节。每章都提供示例程序清单，并辅以示例输出和代码分析以演示该章介绍的主题。

为加深读者对所学内容的理解，每章末尾都提供了常见问题及其答案以及练习和测验。读者可对照附录 D 提供的测验和练习答案，了解自己对所学内容的掌握程度。

## 针对的读者

通过阅读本书来学习 C++时，读者不需要有任何编程经验。本书从入门开始，既介绍 C++语言，又讨论使用 C++进行编程涉及的概念。本书提供了大量语法实例和详细的代码分析，它们是引导读者完成 C++编程之旅的优秀向导。无论读者是刚开始学习编程还是已经有一些编程经验，书中精心安排的内容都将让您的 C++学习过程变得既快速又轻松。

## 本书内容

本书适合初学者阅读，也可供有一定经验的 C++程序员从实用角度更深入了解 C++时参考。本书包含 5 部分：

第一部分简要地介绍了 C++及其语法，这对于需要学习 C++编程基本知识的读者极具参考价值。

第二部分简要地介绍了 C++的面向对象编程功能，这些功能让 C++不同于其前身 C 语言。这部分将为实际使用 C++及其标准模板库打下坚实的基础。

第三部分深入探讨了如何使用 C++编写实用的应用程序，通过使用符合标准的现成结构，可极大地改善应用程序的质量。

第四部分简要地介绍了诸如排序等 STL 算法以及其他 STL 结构，它们有助于改善应用程序的效率和可靠性。

第五部分详细讨论了 C++的一些高级功能。虽然并非编写每个应用程序都涉及这些概念，但了解它们有助于分析错误以及编写出质量更高的代码。

## 本书约定

在本书中，遍布如下提供更多信息的元素：

**提示**　提供使读者进行 C++编程时更高效、更有效的信息。

**注意** 提供与读者阅读的内容相关的信息。

**FAQ**

对 C++语言的用法进行了深入剖析，澄清一些容易混淆的问题。

**警告** 提醒读者注意在特定情况下可能出现的问题或副作用。

| 应该 | 不应该 |
| --- | --- |
| 提供当前章介绍的基本原理的摘要。 | 提供一些有用的信息。 |

## 示例代码

本书正文及附录 D 中的示例代码可从人民邮电出版社网站下载，网址为 http://www.ptpress.com.cn。

# 目　录

## 第一部分　基 础 知 识


第 1 章　绪论 …… 2

1.1　C++简史 …… 2
1.1.1　解释器和编译器 …… 3
1.1.2　不断变化的需求和平台 …… 3
1.1.3　过程化编程、结构化编程和面向对象编程 …… 4
1.1.4　面向对象编程（OOP） …… 4
1.1.5　C++和面向对象编程 …… 4
1.2　C++的发展历程 …… 5
1.3　应该先学习 C 语言吗 …… 5
1.4　微软的 C++托管扩展 …… 6
1.5　ANSI 标准 …… 6
1.6　编程准备 …… 6
1.7　开发环境 …… 7
1.8　创建程序的步骤 …… 7
1.8.1　用编译器生成对象文件 …… 7
1.8.2　用链接器生成可执行文件 …… 7
1.9　程序开发周期 …… 8
1.10　HELLO.cpp：第一个 C++程序 …… 9
1.11　编译器初步 …… 10
1.12　编译错误 …… 11
1.13　总结 …… 11
1.14　问与答 …… 11
1.15　作业 …… 12
1.15.1　测验 …… 12
1.15.2　练习 …… 12

第 2 章　C++程序的组成部分 …… 13

2.1　一个简单程序 …… 13
2.2　cout 简介 …… 14
2.3　使用标准名称空间 …… 16
2.4　对程序进行注释 …… 17
2.4.1　注释的类型 …… 17
2.4.2　使用注释 …… 18
2.4.3　有关注释的警告 …… 18
2.5　函数 …… 18
2.5.1　使用函数 …… 19
2.5.2　方法和函数 …… 21
2.6　总结 …… 21
2.7　问与答 …… 21
2.8　作业 …… 21
2.8.1　测验 …… 21
2.8.2　练习 …… 21

第 3 章　使用变量和常量 …… 23

3.1　什么是变量 …… 23
3.1.1　将数据存储在内存中 …… 23
3.1.2　预留内存 …… 24
3.1.3　整型变量的大小 …… 24
3.1.4　基本变量类型 …… 24
3.2　定义变量 …… 25
3.2.1　区分大小写 …… 26
3.2.2　命名规则 …… 26
3.2.3　关键字 …… 27
3.3　确定变量类型占用的内存量 …… 27
3.4　一次创建多个变量 …… 28
3.5　给变量赋值 …… 28
3.6　使用 typedef 创建别名 …… 29
3.7　何时使用 short 和 long …… 30
3.7.1　unsigned 整型变量的回绕 …… 31
3.7.2　signed 整型变量的回绕 …… 31
3.8　使用字符 …… 32
3.8.1　字符和数字 …… 32
3.8.2　特殊打印字符 …… 33
3.9　常量 …… 34
3.9.1　字面常量 …… 34
3.9.2　符号常量 …… 34
3.10　枚举常量 …… 35
3.11　总结 …… 36
3.12　问与答 …… 37
3.13　作业 …… 37
3.13.1　测验 …… 37
3.13.2　练习 …… 38

第 4 章 管理数组和字符串……39
4.1 什么是数组……39
4.1.1 访问数组元素……39
4.1.2 在数组末尾后写入数据……40
4.1.3 护栏柱错误……42
4.1.4 初始化数组……42
4.1.5 声明数组……43
4.2 使用多维数组……44
4.2.1 声明多维数组……44
4.2.2 初始化多维数组……44
4.3 字符数组和字符串……46
4.4 使用方法 strcpy( )和 strncpy( )……48
4.5 string 类……49
4.6 总结……50
4.7 问与答……51
4.8 作业……51
4.8.1 测验……51
4.8.2 练习……51
第 5 章 使用表达式、语句和运算符……53
5.1 语句简介……53
5.1.1 使用空白……53
5.1.2 语句块和复合语句……54
5.2 表达式……54
5.3 使用运算符……55
5.3.1 赋值运算符……55
5.3.2 数学运算符……55
5.3.3 整数除法和求模……56
5.4 结合使用赋值运算符与数学运算符……57
5.5 递增和递减……57
5.6 理解运算符优先级……59
5.7 括号的嵌套……59
5.8 真值的本质……60
5.9 if 语句……61
5.9.1 缩进风格……63
5.9.2 else 语句……63
5.9.3 高级 if 语句……65
5.10 在嵌套 if 语句中使用大括号……66
5.11 使用逻辑运算符……68
5.11.1 逻辑 AND 运算符……68
5.11.2 逻辑 OR 运算符……68
5.11.3 逻辑 NOT 运算符……68
5.12 简化求值……68
5.13 关系运算符的优先级……69
5.14 再谈真和假……69
5.15 条件运算符（三目运算符）……70
5.16 总结……71
5.17 问与答……71
5.18 作业……71
5.18.1 测验……71
5.18.2 练习……72
第 6 章 使用函数组织代码……73
6.1 什么是函数……73
6.2 返回值、参数和实参……74
6.3 声明和定义函数……74
6.3.1 函数原型……74
6.3.2 定义函数……75
6.4 函数的执行……76
6.5 确定变量的作用域……77
6.5.1 局部变量……77
6.5.2 作用域为语句块的局部变量……78
6.6 参数是局部变量……79
6.6.1 全局变量……80
6.6.2 有关全局变量的注意事项……81
6.7 创建函数语句时的考虑因素……81
6.8 再谈函数实参……81
6.9 再谈返回值……82
6.10 默认参数……83
6.11 重载函数……85
6.12 函数特有的主题……87
6.12.1 内联函数……87
6.12.2 递归……89
6.13 函数的工作原理……92
6.13.1 抽象层次……92
6.13.2 划分 RAM……92
6.13.3 堆栈和函数……93
6.14 总结……94
6.15 问与答……94
6.16 作业……95
6.16.1 测验……95
6.16.2 练习……95
第 7 章 控制程序流程……97
7.1 循环……97
7.1.1 循环的鼻祖：goto……97
7.1.2 为何避免使用 goto 语句……98
7.2 使用 while 循环……98
7.2.1 更复杂的 while 语句……99
7.2.2 continue 和 break 简介……100
7.2.3 while(true)循环……102

7.3 实现 do...while 循环……103
7.4 使用 do...while……103
7.5 for 循环……105
7.5.1 高级 for 循环……106
7.5.2 空 for 循环……108
7.5.3 循环嵌套……109
7.5.4 for 循环中声明的变量的作用域……110
7.6 循环小结……111
7.7 使用 switch 语句控制程序流程……112
7.8 总结……116
7.9 问与答……117
7.10 作业……117
7.10.1 测验……117
7.10.2 练习……117

**第 8 章 阐述指针**……119

8.1 什么是指针……119
8.1.1 内存简介……119
8.1.2 获取变量的内存地址……120
8.1.3 将变量的地址存储到指针中……120
8.1.4 指针名……121
8.1.5 获取指针指向的变量的值……121
8.1.6 使用间接运算符解除引用……122
8.1.7 指针、地址和变量……122
8.1.8 使用指针来操纵数据……123
8.1.9 查看地址……124
8.1.10 指针和数组名……125
8.1.11 数组指针和指针数组……126
8.2 为什么使用指针……127
8.3 栈和自由存储区（堆）……127
8.3.1 使用关键字 new 分配内存……128
8.3.2 使用关键字 delete 归还内存……128
8.4 再谈内存泄漏……130
8.5 在自由存储区上创建对象……130
8.6 删除自由存储区中的对象……130
8.7 迷途指针……131
8.8 使用 const 指针……133
8.9 总结……134
8.10 问与答……134
8.11 作业……134
8.11.1 测验……134
8.11.2 练习……135

**第 9 章 使用引用**……136

9.1 什么是引用……136
9.2 将地址运算符用于引用……137
9.3 空指针和空引用……139
9.4 按引用传递函数参数……139
9.4.1 使用指针让 swap( )管用……140
9.4.2 使用引用来实现 swap( )……141
9.5 返回多个值……142
9.6 按引用传递以提高效率……145
9.6.1 传递 const 指针……147
9.6.2 用引用代替指针……148
9.7 何时使用引用和指针……150
9.8 混合使用引用和指针……150
9.9 返回指向不在作用域中的对象的引用……151
9.10 总结……153
9.11 问与答……153
9.12 作业……153
9.12.1 测验……154
9.12.2 练习……154

## 第二部分 面向对象编程和 C++基础

**第 10 章 类和对象**……156

10.1 C++是面向对象的吗……156
10.2 创建新类型……157
10.3 类和成员简介……157
10.3.1 声明类……158
10.3.2 有关命名规则的说明……158
10.3.3 定义对象……159
10.3.4 类与对象……159
10.4 访问类成员……159
10.4.1 给对象而不是类赋值……159
10.4.2 类不能有没有声明的功能……159
10.5 私有和公有……160
10.6 实现类方法……163
10.7 添加构造函数和析构函数……165
10.7.1 默认构造函数和析构函数……166
10.7.2 使用默认构造函数……166
10.8 const 成员函数……168
10.9 将类声明和方法定义放在什么地方……169
10.10 内联实现……169
10.11 将其他类用作成员数据的类……171
10.12 探索结构……174
10.13 总结……174
10.14 问与答……174
10.15 作业……175

10.15.1 测验……175
10.15.2 练习……176

第 11 章 实现继承……177

11.1 什么是继承……177
11.1.1 继承和派生……177
11.1.2 动物世界……178
11.1.3 派生的语法……178
11.2 私有和保护……180
11.3 构造函数和析构函数的继承性……181
11.4 覆盖基类函数……186
11.4.1 隐藏基类的方法……187
11.4.2 调用基类方法……189
11.5 虚方法……190
11.5.1 虚函数的工作原理……193
11.5.2 通过基类指针访问派生类的方法……193
11.5.3 切除……194
11.5.4 创建虚析构函数……195
11.5.5 虚复制构造函数……195
11.5.6 使用虚方法的代价……198
11.6 私有继承……198
11.6.1 使用私有继承……198
11.6.2 私有继承和聚合（组合）……199
11.7 总结……200
11.8 问与答……201
11.9 作业……201
11.9.1 测验……201
11.9.2 练习……202

第 12 章 多态……203

12.1 单继承存在的问题……203
12.1.1 提升……205
12.1.2 向下转换……205
12.1.3 将对象添加到链表中……207
12.2 多重继承……207
12.2.1 多重继承对象的组成部分……209
12.2.2 多重继承对象中的构造函数……210
12.2.3 避免歧义……212
12.2.4 从共同基类继承……212
12.2.5 虚继承……215
12.2.6 多重继承存在的问题……217
12.2.7 混合（功能）类……217
12.3 抽象数据类型……218
12.3.1 纯虚函数……220
12.3.2 实现纯虚函数……221
12.3.3 复杂的抽象层次结构……224
12.3.4 哪些类是抽象的……226
12.4 总结……226
12.5 问与答……227
12.6 作业……227
12.6.1 测验……227
12.6.2 练习……228

第 13 章 运算符类型与运算符重载……229

13.1 C++中的运算符……229
13.2 单目运算符……229
13.2.1 单目运算符的类型……230
13.2.2 单目递增与单目递减运算符……230
13.2.3 解除引用运算符*与成员选择运算符->的编程……232
13.2.4 转换运算符的编程……234
13.3 双目运算符……235
13.3.1 双目运算符的类型……235
13.3.2 双目加与双目减运算符的编程……236
13.3.3 运算符+=与-=的编程……237
13.3.4 重载比较运算符……238
13.3.5 重载运算符<、>、<=和>=……241
13.3.6 下标运算符……243
13.4 operator()函数……244
13.5 不能重新定义的运算符……245
13.6 总结……245
13.7 问与答……245
13.8 作业……246
13.8.1 测验……246
13.8.2 练习……246

第 14 章 类型转换运算符……247

14.1 什么是类型转换……247
14.2 为何需要类型转换……247
14.3 为何有些 C++程序员不喜欢 C 风格类型转换……248
14.4 C++类型转换运算符……248
14.4.1 使用 static_cast……248
14.4.2 使用 dynamic_cast 和运行阶段类型识别……249
14.4.3 使用 reinterpret_cast……250
14.4.4 使用 const_cast……251
14.5 C++类型转换运算符存在的问题……252
14.6 总结……252
14.7 问与答……252
14.8 作业……253

试读结束：需要全本请在线购买： www.ertongbook.com

第 15 章 宏和模板简介 ……254
15.1 预处理器与编译器 ……254
15.2 预处理器指令#define ……254
15.3 宏函数 ……255
15.3.1 为什么要使用括号 ……255
15.3.2 宏与类型安全问题 ……256
15.3.3 宏与函数及模板之比较 ……256
15.3.4 内联函数 ……256
15.4 模板简介 ……258
15.4.1 模板声明语法 ……258
15.4.2 各种类型的模板声明 ……258
15.4.3 模板类 ……259
15.4.4 模板的实例化和具体化 ……259
15.4.5 模板与类型安全 ……259
15.4.6 使用多个参数声明模板 ……259
15.4.7 使用默认参数来声明模板 ……260
15.4.8 一个模板示例 ……260
15.4.9 在实际 C++编程中使用模板 ……261
15.5 总结 ……262
15.6 问与答 ……262
15.7 作业 ……263
15.7.1 测验 ……263
15.7.2 练习 ……263

## 第三部分 学习标准模板库（STL）

第 16 章 标准模板库简介 ……266
16.1 STL 容器 ……266
16.1.1 顺序容器 ……266
16.1.2 关联容器 ……266
16.1.3 选择正确的容器 ……267
16.2 STL 迭代器 ……267
16.3 STL 算法 ……268
16.4 使用迭代器在容器和算法之间交互 ……268
16.5 总结 ……270
16.6 问与答 ……270
16.7 作业 ……270
第 17 章 STL string 类 ……271
17.1 为何需要字符串操作类 ……271
17.2 使用 STL string 类 ……272
17.2.1 实例化 STL string 及复制 ……272
17.2.2 访问 string 及其内容 ……273
17.2.3 字符串连接 ……274
17.2.4 在 string 中查找字符或子字符串 ……275
17.2.5 截短 STL string ……276
17.2.6 字符串反转 ……278
17.2.7 字符串的大小写转换 ……278
17.3 基于模板的 STL string 实现 ……279
17.4 总结 ……279
17.5 问与答 ……279
17.6 作业 ……280
17.6.1 测验 ……280
17.6.2 练习 ……280
第 18 章 STL 动态数组类 ……281
18.1 std::vector 的特点 ……281
18.2 典型的 vector 操作 ……281
18.2.1 实例化 vector ……281
18.2.2 在 vector 中插入元素 ……282
18.2.3 访问 vector 中的元素 ……285
18.2.4 删除 vector 中的元素 ……286
18.3 理解 size()和 capacity() ……287
18.4 STL deque 类 ……288
18.5 总结 ……290
18.6 问与答 ……290
18.7 作业 ……290
18.7.1 测验 ……291
18.7.2 练习 ……291
第 19 章 STL list ……292
19.1 std::list 的特点 ……292
19.2 基本的 list 操作 ……292
19.2.1 实例化 std::list 对象 ……292
19.2.2 在 list 开头插入元素 ……293
19.2.3 在 list 末尾插入元素 ……293
19.2.4 在 list 中间插入元素 ……294
19.2.5 删除 list 中的元素 ……296
19.3 对 list 中元素进行反转和排序 ……297
19.3.1 反转元素的排列顺序 ……297
19.3.2 元素排序 ……298
19.4 总结 ……305
19.5 问与答 ……305
19.6 作业 ……305
19.6.1 测验 ……305
19.6.2 练习 ……305

第 20 章 STL set 与 multiset……306

20.1 简介……306
20.2 STL set 和 multiset 的基本操作……306
20.2.1 实例化 std::set 对象……306
20.2.2 在 STL set 或 multiset 中插入元素……307
20.2.3 在 STL set 或 multiset 中查找元素……308
20.2.4 删除 STL set 或 multiset 中的元素……309
20.3 使用 STL set 和 multiset 的优缺点……315
20.4 总结……316
20.5 问与答……316
20.6 作业……316
20.6.1 测验……316
20.6.2 练习……316

第 21 章 STL map 和 multimap……317

21.1 简介……317
21.2 STL map 和 multimap 的基本操作……317
21.2.1 实例化 std::map 对象……317
21.2.2 在 STL map 或 multimap 中插入元素……318
21.2.3 在 STL map 或 multimap 中查找元素……320
21.2.4 删除 STL map 或 multimap 中的元素……321
21.3 提供自定义的排序谓词……323
21.4 总结……325
21.5 问与答……325
21.6 作业……326
21.6.1 测验……326
21.6.2 练习……326

第四部分 再谈 STL

第 22 章 理解函数对象……328

22.1 函数对象与谓词的概念……328
22.2 函数对象的典型用途……328
22.2.1 一元函数……328
22.2.2 一元谓词……331
22.2.3 二元函数……332
22.2.4 二元谓词……334
22.3 总结……336
22.4 问与答……336
22.5 作业……336
22.5.1 测验……336
22.5.2 练习……336

第 23 章 STL 算法……337

23.1 什么是 STL 算法……337
23.2 STL 算法的分类……337
23.2.1 非变序算法……337
23.2.2 变序算法……338
23.3 STL 算法的应用……339
23.3.1 计算元素个数与查找元素……339
23.3.2 在集合中搜索元素或序列……340
23.3.3 将容器中的元素初始化为指定值……342
23.3.4 用 for_each 处理范围内的元素……344
23.3.5 使用 std::transform 对范围进行变换……345
23.3.6 复制和删除操作……347
23.3.7 替换值以及替换满足给定条件的元素……349
23.3.8 排序、在有序集合中搜索以及删除重复元素……350
23.3.9 将范围分区……351
23.3.10 在有序集合中插入元素……353
23.4 总结……354
23.5 问与答……354
23.6 作业……355
23.6.1 测验……355
23.6.2 练习……355

第 24 章 自适应容器：栈和队列……356

24.1 栈和队列的行为特征……356
24.1.1 栈……356
24.1.2 队列……356
24.2 使用 STL stack 类……356
24.2.1 实例化 stack……357
24.2.2 stack 的成员函数……357
24.3 使用 STL queue 类……359
24.3.1 实例化 queue……359
24.3.2 queue 的成员函数……359
24.4 使用 STL 优先级队列……361
24.4.1 实例化 priority_queue 类……361
24.4.2 priority_queue 的成员函数……362

24.5 总结……364
24.6 问与答……364
24.7 作业……364
24.7.1 测验……364
24.7.2 练习……364

**第 25 章 使用 STL 位标志……365**

25.1 bitset 类……365
25.2 使用 std::bitset 及其成员……366
25.2.1 std:bitset 的运算符……366
25.2.2 std::bitset 的成员方法……366
25.3 vector<bool>……368
25.3.1 实例化 vector<bool>……368
25.3.2 使用 vector<bool>……369
25.4 总结……370
25.5 问与答……370
25.6 作业……370
25.6.1 测验……370
25.6.2 练习……370

## 第五部分 高级 C++概念

**第 26 章 理解智能指针……372**

26.1 什么是智能指针……372
26.1.1 使用常规（原始）指针有何问题……372
26.1.2 智能指针有何帮助……372
26.2 智能指针是如何实现的……373
26.3 智能指针类型……374
26.3.1 深度复制……374
26.3.2 写时复制机制……375
26.3.3 引用计数智能指针……375
26.3.4 引用链接智能指针……376
26.3.5 破坏性复制……376
26.4 使用 std::auto_ptr……377
26.5 流行的智能指针库……378
26.6 总结……378
26.7 问与答……379
26.8 作业……379
26.8.1 测试……379
26.8.2 练习……379

**第 27 章 处理流……380**

27.1 流概述……380
27.1.1 数据流的封装……380
27.1.2 理解缓冲技术……381
27.2 流和缓冲区……382
27.3 标准 I/O 对象……382
27.4 重定向标准流……382
27.5 使用 cin 进行输入……382
27.5.1 输入字符串……384
27.5.2 字符串的问题……384
27.5.3 >>的返回值……386
27.6 cin 的其他成员函数……386
27.6.1 单字符输入……386
27.6.2 从标准输入读取字符串……388
27.6.3 使用 cin.ignore()……390
27.6.4 查看和插入字符：peek()和putback()……391
27.7 使用 cout 进行输出……391
27.7.1 刷新输出……391
27.7.2 执行输出的函数……392
27.7.3 控制符、标记和格式化指令……393
27.8 流和 printf()函数之比较……396
27.9 文件输入和输出……398
27.9.1 使用 ofstream……398
27.9.2 条件状态……398
27.9.3 打开文件进行输入和输出……398
27.9.4 修改 ofstream 打开文件时的默认行为……400
27.10 二进制文件和文本文件……401
27.11 命令行处理……403
27.12 总结……405
27.13 问与答……405
27.14 作业……406
27.14.1 测验……406
27.14.2 练习……406

**第 28 章 处理异常……407**

28.1 程序中的各种错误……407
28.2 异常的基本思想……408
28.2.1 异常处理的组成部分……409
28.2.2 手工引发异常……411
28.2.3 创建异常类……412
28.3 使用 try 块和 catch 块……414
28.4 捕获异常的工作原理……415
28.4.1 使用多条 catch 语句……415

28.4.2 异常层次结构……417
28.5 异常中的数据及给异常对象命名……419
28.6 异常和模板……424
28.7 没有错误的异常……426
28.8 bug 和调试……426
28.8.1 断点……427
28.8.2 监视点……427
28.8.3 查看内存……427
28.8.4 查看汇编代码……427
28.9 总结……427
28.10 问与答……427
28.11 作业……428
28.11.1 测验……428
28.11.2 练习……428
**第 29 章 杂项内容**……430
29.1 预处理器和编译器……430
29.2 预编译器指令#define……430
29.2.1 使用#define 定义常量……431
29.2.2 将#define 用于检测……431
29.2.3 预编译器命令#else……431
29.3 包含和防范多重包含……432
29.4 字符串操纵……433
29.4.1 字符串化……433
29.4.2 拼接……433
29.5 预定义的宏……433
29.6 assert( )宏……434
29.6.1 使用 assert( )进行调试……435
29.6.2 assert( )与异常之比较……435
29.6.3 副作用……435
29.6.4 类的不变量……436
29.6.5 打印中间值……439
29.7 位运算……440
29.7.1 “与”运算符……441
29.7.2 “或”运算符……441
29.7.3 “异或”运算符……441
29.7.4 “求反”运算符……441
29.7.5 设置位……441
29.7.6 清除位……441
29.7.7 反转位……442
29.7.8 位字段……442
29.8 编程风格……444
29.8.1 缩进……444
29.8.2 大括号……444
29.8.3 长代码行和函数长度……445
29.8.4 格式化 switch 语句……445
29.8.5 程序文本……445
29.8.6 标识符命名……446
29.8.7 名称的拼写和大写……446
29.8.8 注释……446
29.8.9 设置访问权限……447
29.8.10 类定义……447
29.8.11 包含文件……447
29.8.12 使用 assert( )……447
29.8.13 使用 const……447
29.9 C++开发工作的下一步……447
29.9.1 从何处获得帮助和建议……448
29.9.2 相关的 C++主题：托管 C++、C#和微软的.NET……448
29.10 总结……448
29.11 问与答……449
29.12 作业……450
29.12.1 测验……450
29.12.2 练习……450
**附录 A 二进制和十六进制**……451
A.1 其他进制……451
A.2 不同进制之间的转换……452
A.2.1 二进制……452
A.2.2 为什么使用二进制……453
A.2.3 位、字节和半字节……453
A.2.4 什么是 KB……453
A.2.5 二进制数……454
A.3 十六进制……454
**附录 B C++关键字**……457
**附录 C 运算符优先级**……458
**附录 D 答案**……459

# 第一部分

# 基础知识


第 1 章　绪论

第 2 章　C++程序的组成部分

第 3 章　使用变量和常量

第 4 章　管理数组和字符串

第 5 章　使用表达式、语句和运算符

第 6 章　使用函数组织代码

第 7 章　控制程序流程

第 8 章　阐述指针

第 9 章　使用引用

# 第1章 绪 论

欢迎使用本书！通过阅读本章，您将迈出成为高级 C++程序员的第一步。

在本章中，您将学习：

- 为何 C++是软件开发的标准
- 开发 C++程序的步骤
- 输入、编译和链接第一个 C++程序

## 1.1 C++简史

自第一代电子计算机诞生后，计算机语言经历了翻天覆地的变化。起初，程序员们使用最原始的计算机指令，即机器语言，这些指令是由 0 和 1 组成的字符串。很快，人们就发明了汇编语言，将机器指令映射为人们可以阅读和易于处理的助记符，如 ADD 和 MOV。

然而，随着编写的软件应用程序执行的任务日益复杂（如计算弹道），程序员意识到需要一种能够执行相对复杂的数学指令的语言，这些数学指令可转换为众多的汇编代码（机器语言指令）。FORTRAN 应运而生，它是编程领域中第一种针对数值和科学计算进行了优化的高级编程语言，支持子程序、函数和循环等。随后出现了更高级的语言，如 BASIC 和 COBOL，它们让程序员能够使用类似于单词或句子的源代码（如 Let I=100）进行编程。

C 语言对 B 语言做了革命性改进，而 B 语言是 BCPL（Basic Combined Programming Language）语言的改进版本。虽然发明 C 语言旨在帮助程序员使用当时新出现的硬件功能，但它得以流行应主要归功于其可移植性和速度。C 语言是一种过程化语音，但随着计算机语言进入面向对象时代，Bjarne Stroustrup 于 1981 年发明了 C++，它是发展最快、使用最广泛的编程语言之一。除新增了诸如运算符重载和内联函数等功能外，C++还实现了诸如继承（支持多继承）、封装、抽象和多态等面向对象概念。C++还实现并不断改进了模板（泛型类或函数）概念，而诸如 Java 和 C#等较新的语言直到最近才支持这种概念。

在 C++之后，Java 给编程领域带来了又一次革命。它得以流行的主要原因是 Java 应用程序可在多种流行的平台中运行；另一个原因是其简单性，它不支持众多让 C++功能强大的功能。除不支持指针外，Java 还负责为用户管理内存和执行垃圾收集。在 Java 之后，C#是最先开发的基于框架（微软.NET 框架）的语言之一。C#借鉴了 Java 和 C++的设计思想和语法，但在有些方面与这两种语言都不同。.NET 框架支持管理版 C++（称为托管 C++），它向 C++程序员提供了.NET 框架的优点（如自动管理内存和收集垃圾），且执行速度比其他基于框架的语言（如 C#）快。

当前，很多应用程序仍是使用 C++编写的，这不仅是因为更新的语言仍不能满足众多应用程序的需求，还因为 C++向程序员提供了灵活性和强大功能。C++是一种不断发展的语言，它遵循 ANSI 标准。

试读结束：需要全本请在线购买： www.ertongbook.com