

SAMS

计算机技术

译林

精选系列

21天学通

C++
(第四版)

[美] Jesse Liberty 著

康博创作室 译

人民邮电出版社

POSTS & TELECOMMUNICATIONS PRESS

图书在版编目(CIP)数据

21 天学通 C++ ;第4版/(美)利伯特(Liberty, J.)著;康博创作室译. —北京:人民邮电出版社, 2002. 3
(计算机技术译林精选系列)

ISBN 7-115-09978-2

I . 2 . . . II . ①利 . . . ②康 . . . III . C 语言 - 程序设计 IV . TP312

中国版本图书馆 CIP 数据核字(2001)第 096534 号

版权声明

Jesse Liberty: Teach Yourself C++ in 21 Days (Fourth Edition)。

Authorized translation from English language edition published by SAMS Publishing。

Copyright © 2001 by SAMS Publishing。

All rights reserved. For sale in mainland China only。

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, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

计算机技术译林精选系列

21 天学通 C++ (第四版)

- ◆ 著 [美] Jesse Liberty
译 康博创作室
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 010-67180876
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 40
字数: 1281 千字 2002 年 3 月第 1 版
印数: 1.4 000 册 2002 年 3 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2001 - 2046 号

ISBN 7-115-09978-2/TP · 2690

定价: 58.00 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

本书全面系统地介绍了 C++ 编程的基本知识,如控制输入/输出、循环和数组、目标编程、模板以及创建 C++ 应用程序,所有这些内容都包括在结构合理、联系紧密的章节之中。本书各章都提供了例程的清单,这些清单包括例程的输出以及代码的分析。此外,为了帮助读者加深理解所学内容,每章结束时都提供了一组常见问题及答案、练习题和测验。通过对照检查附录 D 所给出的测验和练习题的答案,可评估自己的学习进展情况。

本书可作为没有任何编程经验的新手学习 C++ 的教材,另外由于第四版是这本书的最新版本,所以本书也可作为有经验的 C++ 程序员了解最新编程内容的参考书。

作者简介

Jesse Liberty 是编著了 12 本有关 C++ , C# 及面向对象分析和设计的书籍的作者。他是 Liberty Associates(<http://WWW.LibertyAssociates.com>)公司的总裁,该公司提供 .net 开发、编程、顾问、咨询以及培训。

Jesse 曾是一名杰出的 AT&T 软件工程师, Xerox 和 LinkNet(PBS)的软件设计师,花旗银行开发部的副总裁。他和他的妻子 Stacey 及两个女儿 Robin 和 Rachel 共同生活在 Massachusetts 州的 Cambridge 郊区。在其 Web 站点 <http://www.libertyassociates.com> 上单击 Books and Resources 就可得到 Jesse 对他的著作的支持。

本书的目的是帮助你学会使用 C++ 来编写程序。没有人可以在仅仅 3 个星期之内学好一种结构严谨的编程语言,这本书将向你介绍 C++ 语言的主要概念。在这本书中,每一章节的编写可以让你在一天之内掌握其所讲的内容。

在这 21 章中,你将学习一些基本知识,比如控制输入/输出、循环和数组、面向对象编程、模板以及创建 C++ 应用程序,所有这些内容都包括在结构合理、联系紧密的章节之中。所有章节都提供例程的清单用来举例说明每天所学的题目,这些清单包括例程的输出以及代码的分析。

为了帮助读者加深理解所学内容,每章结束时都提供了一组常见问题及答案、练习题和测验。通过对照检查附录 D 所给出的测验和练习题的答案,读者可评估自己的学习进展情况。

本书读者对象

通过本书学习 C++ 时,并不要求读者以前有任何编程经验。本书从入门开始,既讲授 C++ 语言,也介绍 C++ 编程中涉及到的一些概念。本书提供了大量语法实例及代码的详细分析,它们是引导读者快速进入 C++ 编程的优秀向导。无论读者是刚开始编程还是已经有了一些编程经验,书中精心安排的内容都会使你的 C++ 学习变得既快又轻松。

本书约定

注意:

这种字体样式用来强调一些信息,它们可使 C++ 编程更有效、更实用。

常见问题:

常见问题有何作用?

答:常见问题提供了对 C++ 语言用法的深入剖析,并澄清一些容易混淆的问题。

警告：

警告用来提醒读者注意在特殊情况下可能出现的问题或副作用。

建议做的事：

通过这部分内容,读者可找出某一章中基本原理的简洁摘要。

不该做的事：

不要小看这部分所提供的有用信息,它很重要。

在本书的程序清单中,每行代码都编了号。如果在程序清单中看到某行未编号,表示该行是前一行的续行(有的代码行太长,只能分行列出)。这种情况下,输入时必须将其按一行输入,而不能分开。

第一篇

第一章 开始	2
1.1 引言	2
1.2 C++ 历史简介	2
1.2.1 本书要解决的问题	2
1.2.2 过程编程、结构化编程和面向对象的编程	3
1.2.3 C++ 和面向对象编程	4
1.3 C++ 的发展历程	4
1.4 学习 C++ 应该先学 C 吗	5
1.5 C++ 和 Java 及 C#	5
1.6 ANSI 标准	5
1.7 准备编程	5
1.8 开发环境	6
1.9 生成程序	6
1.9.1 用编译器生成对象文件	6
1.9.2 用链接器生成可执行文件	6
1.10 程序开发周期	7
1.11 HELLO.cpp——你的第一个 C++ 程序	8
1.12 开始学习编译器	9
1.12.1 创建 Hello World 项目	9
1.13 编译错误	9
1.14 小结	10
1.15 问与答	10
1.16 习题	10
1.16.1 测验	11
1.16.2 练习	11
第二章 C++ 程序的组成部分	12
2.1 一个简单程序	12
2.2 cout 简介	13
2.3 使用标准的 Namespace	15

2.4	注释	17
2.4.1	注释的类型	17
2.4.2	使用注释	17
2.4.3	对注释的忠告	18
2.5	函数	18
2.5.1	函数的使用	19
2.6	小结	20
2.7	问与答	20
2.8	习题	21
2.8.1	测验	21
2.8.2	练习	21
第三章	变量与常量	22
3.1	什么是变量	22
3.1.1	将数据存储在内存中	22
3.1.2	留出内存	22
3.1.3	整型量的大小	23
3.1.4	有符号和无符号	24
3.1.5	基本变量类型	24
3.2	定义一个变量	25
3.2.1	大小写敏感性	26
3.2.2	关键字	26
3.3	一次创建多个变量	27
3.4	给变量赋值	27
3.5	typedef	28
3.6	short 与 long 的使用	29
3.6.1	无符号整型量的回绕	29
3.6.2	有符号整型量的回绕	30
3.7	字符型变量	31
3.7.1	字符和数字	31
3.7.2	特殊打印字符	32
3.8	常量	32
3.8.1	字面常量	33
3.8.2	符号常量	33
3.9	枚举型常量	34
3.10	小结	35
3.11	问与答	36
3.12	习题	36
3.12.1	测验	36

3.12.2 练习	37
第四章 表达式和语句	38
4.1 语句	38
4.1.1 空白	38
4.1.2 块和复合语句	38
4.2 表达式	39
4.3 运算符	40
4.3.1 赋值运算符	40
4.3.2 数学运算符	40
4.3.3 整型量除法和取模	41
4.4 赋值运算符与数学运算符的组合	41
4.5 自加与自减	42
4.5.1 前置和后置	42
4.6 优先级	44
4.7 括号的嵌套	44
4.8 真值的本质	45
4.8.1 关系运算符	45
4.9 if 语句	46
4.9.1 缩进样式	48
4.9.2 else	49
4.9.3 高级 if 语句	50
4.10 在嵌套的 if 语句中使用大括号	52
4.11 逻辑运算符	54
4.11.1 逻辑“与”	54
4.11.2 逻辑“或”	54
4.11.3 逻辑“非”	54
4.12 简化求值	55
4.13 关系运算符的优先级	55
4.14 有关真假关系的进一步讨论	55
4.15 条件运算符(三目运算符)	56
4.16 小结	57
4.17 问与答	57
4.18 习题	58
4.18.1 测验	58
4.18.2 练习	58
第五章 函数	60
5.1 什么是函数	60

5.2	函数的返回值、参数和变元	61
5.3	声明和定义函数	61
5.3.1	函数的声明	61
5.3.2	函数原型	61
5.3.3	定义函数	62
5.4	函数的执行	63
5.5	局部变量	63
5.5.1	作用范围	65
5.6	全局变量	65
5.7	使用全局变量的注意事项	66
5.8	有关局部变量的更多知识	66
5.9	函数语句	67
5.10	有关函数变元的更多知识	67
5.10.1	使用函数作为函数的参数	68
5.11	参数作为局部变量	68
5.12	有关返回值的更多知识	69
5.13	默认参数	71
5.14	函数重载	72
5.15	有关函数的一些专门话题	75
5.15.1	内嵌函数	75
5.15.2	递归	76
5.16	函数是如何工作的	80
5.16.1	抽象层次	80
5.16.2	划分 RAM	80
5.16.3	堆栈与函数	81
5.17	小结	82
5.18	问与答	82
5.19	习题	83
5.19.1	测验	83
5.19.2	练习	83
第六章	面向对象编程	85
6.1	C++ 是面向对象的吗	85
6.2	创建新类型	85
6.2.1	为什么要创建新类型	86
6.3	类和成员	86
6.3.1	声明一个类	86
6.3.2	有关命名习惯的一点说明	87
6.3.3	定义一个对象	87

6.3.4 类与对象	87
6.4 访问类成员	87
6.4.1 给对象赋值而不是给类赋值	88
6.4.2 如果你不声明它,你的类就没有它	88
6.5 私有和公有	88
6.5.1 使成员数据私有	90
6.5.2 私有性与安全性	91
6.6 实现类的方法	92
6.7 构造函数和析构函数	94
6.7.1 默认的构造函数和析构函数	95
6.7.2 使用默认构造函数	95
6.8 成员函数 const	97
6.9 界面与实现	98
6.10 在何处放置类声明和方法定义	100
6.11 内嵌实现	100
6.12 用其他类作为成员数据的类	102
6.13 结构体	105
6.13.1 为什么两个关键字做同一件事	106
6.14 小结	106
6.15 问与答	106
6.16 习题	107
6.16.1 测验	107
6.16.2 练习	107
第七章 程序流程	109
7.1 循环	109
7.1.1 循环语句的起源——goto 语句	109
7.1.2 为什么现在不再使用 goto 语句	110
7.2 while 循环	110
7.2.1 复杂的 while 语句	111
7.2.2 continue 和 break	112
7.2.3 while(true) 循环	114
7.3 do...while 循环	115
7.4 do...while	116
7.5 for 循环	118
7.5.1 高级 for 循环	119
7.5.2 多重初始化和自增运算	120
7.5.3 在 for 循环中使用空语句	120
7.5.4 空 for 循环	122

7.5.5 嵌套循环	123
7.5.6 for 循环的作用域	124
7.6 循环总结	125
7.7 switch 语句	126
7.7.1 使用带有菜单的 switch 语句	129
7.8 小结	131
7.9 问与答	131
7.10 习题	132
7.10.1 测验	132
7.10.2 练习	132
第一篇内容回顾	134

第二篇

第八章 指针	140
8.1 什么是指针	140
8.1.1 在指针中存储地址	141
8.1.2 指针名	142
8.1.3 间接引用运算符	142
8.1.4 指针、地址和变量	143
8.1.5 使用指针操作数据	144
8.1.6 检查地址	145
8.2 为什么要使用指针	146
8.3 栈和自由存储区(堆)	147
8.3.1 new	147
8.3.2 delete	148
8.4 内存泄漏	149
8.5 在自由存储区内创建对象	150
8.6 删除对象	150
8.7 访问数据成员	151
8.8 自由存储区中的成员数据	152
8.9 this 指针	155
8.10 迷途指针、失控指针或悬浮指针	156
8.11 const 指针	158
8.11.1 const 指针和 const 成员函数	158
8.11.2 const this 指针	160
8.12 指针算法高级主题	160
8.13 小结	162
8.14 问与答	162
8.15 习题	163

8.15.1	测验	163
8.15.2	练习	163
第九章	引用	165
9.1	什么是引用	165
9.2	对引用使用取址运算符	166
9.2.1	引用不能被重新赋值	167
9.3	什么可以被引用	168
9.4	空指针和空引用	170
9.5	通过引用传递函数参数	170
9.5.1	用指针使 <code>swap()</code> 工作	171
9.5.2	使用引用实现 <code>swap()</code>	172
9.6	理解函数头文件和原型	174
9.7	返回多个值	174
9.7.1	通过引用返回值	175
9.8	使用引用传递来提高效率	177
9.8.1	传递 <code>const</code> 指针	179
9.8.2	使用引用代替指针	181
9.9	何时使用引用以及何时使用指针	183
9.10	混合使用引用和指针	183
9.11	不要返回对不在作用域中的对象的引用	184
9.12	返回对一个在堆中的对象的引用	185
9.13	指针的传递	187
9.14	小结	188
9.15	问与答	188
9.16	习题	188
9.16.1	测验	188
9.16.2	练习	188
第十章	高级函数	190
10.1	重载成员函数	190
10.2	使用默认值	192
10.3	默认值和重载函数的选择	194
10.4	默认构造函数	194
10.5	重载构造函数	194
10.6	初始化对象	196
10.7	复制构造函数	196
10.8	运算符重载	200
10.8.1	写一个自加函数	200

10.8.2	重载前置运算符	201
10.8.3	重载运算符函数的返回类型	203
10.8.4	返回无名临时对象	204
10.8.5	使用 this 指针	205
10.8.6	重载后置运算符	206
10.8.7	前置和后置之间的区别	207
10.8.8	加法运算符	208
10.8.9	重载 operator +	210
10.8.10	运算符重载中存在的问题	211
10.8.11	运算符重载的局限性	211
10.8.12	重载什么	211
10.8.13	赋值运算符	212
10.9	处理数据类型转换	214
10.9.1	转换运算符	216
10.10	小结	217
10.11	问与答	217
10.12	习题	218
10.12.1	测验	218
10.12.2	练习	218
第十一章	面向对象的分析及设计	220
11.1	构造模型	220
11.2	软件设计:建模语言	220
11.3	软件设计:过程	221
11.4	Extreme 编程	222
11.5	概览	223
11.6	需求分析	223
11.6.1	使用情况	223
11.6.2	应用分析	230
11.6.3	系统分析	230
11.6.4	计划文档	231
11.6.5	可视化	231
11.6.6	人工制品	231
11.7	设计	232
11.7.1	什么是类	232
11.7.2	变换	233
11.7.3	静态模型	234
11.7.4	动态模型	240
11.8	小结	242

11.9 问与答	242
11.10 习题	243
11.10.1 测验	243
11.10.2 练习	243
第十二章 继承	244
12.1 什么是继承	244
12.1.1 继承和派生	244
12.1.2 动物世界	245
12.1.3 派生的语法	245
12.2 私有型与保护型	247
12.3 构造函数和析构造函数	248
12.3.1 向基构造函数传递参数	250
12.4 覆盖函数	254
12.4.1 隐藏基类方法	255
12.4.2 调用基方法	257
12.5 虚函数	258
12.5.1 虚函数是怎样工作的	261
12.5.2 不能从基类中访问派生类中的函数	262
12.5.3 切片(slicing)	262
12.5.4 虚析构造函数	264
12.5.5 虚复制构造函数	265
12.5.6 虚函数的代价	267
12.6 小结	267
12.7 问与答	268
12.8 习题	268
12.8.1 测验	268
12.8.2 练习	269
第十三章 数组和链表	270
13.1 什么是数组	270
13.2 数组元素	270
13.3 超过数组末尾写入数据	271
13.4 篱笆标志错误	274
13.5 初始化数组	274
13.6 声明数组	275
13.7 对象的数组	276
13.8 多维数组	277
13.9 初始化多维数组	278

13.10	有关内存的一些话题	279
13.11	指针数组	279
13.12	在自由存储区声明数组	281
13.13	指向数组的指针和指针数组	281
13.14	指针与数组名	281
13.15	删除自由存储区中的数组	283
13.16	char 数组	283
13.17	strcpy()和 strncpy()	285
13.18	字符串类	286
13.19	链表和其他结构	291
13.20	链表案例研究	292
13.20.1	责任授权	292
13.21	组件部分	292
13.22	你学到了什么	299
13.23	数组类	299
13.24	小结	300
13.25	问与答	300
13.26	习题	300
13.26.1	测验	301
13.26.2	练习	301
第十四章	多态性	302
14.1	单一继承的问题	302
14.1.1	升迁	304
14.1.2	下降	304
14.1.3	添加到两个链表	306
14.2	多重继承	306
14.2.1	多重继承对象的构成	309
14.2.2	多重继承对象中的构造函数	309
14.2.3	歧义解析	312
14.2.4	从共享基类中继承	312
14.2.5	虚继承	315
14.2.6	多重继承中存在的问题	318
14.2.7	混合继承和功能类	319
14.3	抽象数据类型	319
14.3.1	纯虚函数	322
14.3.2	实现纯虚函数	323
14.3.3	复杂的抽象层次	326
14.3.4	哪些类型是抽象的	329

14.4 小结	329
14.5 问与答	329
14.6 习题	330
14.6.1 测验	330
14.6.2 练习	330
第二篇内容回顾	331

第三篇

第十五章 特殊类和函数	340
15.1 静态成员数据	340
15.2 静态成员函数	344
15.3 函数指针	346
15.3.1 为什么使用函数指针	349
15.3.2 函数指针数组	351
15.3.3 把函数指针传递给其他函数	353
15.3.4 在函数指针中使用 typedef	356
15.4 成员函数指针	358
15.4.1 成员函数指针数组	360
15.5 小结	361
15.6 问与答	362
15.7 习题	362
15.7.1 测验	362
15.7.2 练习	362
第十六章 高级继承	364
16.1 包容	364
16.1.1 访问被包容类的成员	369
16.1.2 过滤对被包容成员的访问	369
16.1.3 包容的代价	370
16.1.4 按值来复制	372
16.2 按继承/包容条款实现与代理	375
16.2.1 授权	376
16.3 私有继承	383
16.4 友元类	390
16.5 友元函数	397
16.6 友元函数和运算符重载	397
16.7 重载插入运算符	401
16.8 小结	405