

《C++ Primer 中文版（第5版）》

图书基本信息

书名：《C++ Primer 中文版（第5版）》

13位ISBN编号：9787121155352

10位ISBN编号：7121155354

出版时间：2013-9-1

出版社：电子工业出版社

作者：[美] Stanley B. Lippman,[美] Jos é e Lajoie,[美] Barbara E. Moo

页数：838

译者：王刚,杨巨峰

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

内容概要

这本久负盛名的 C++ 经典教程，时隔八年之久，终迎来史无前例的重大升级。除令全球无数程序员从中受益，甚至为之迷醉的——C++ 大师 Stanley B. Lippman 的丰富实践经验，C++ 标准委员会原负责人 Jos é e Lajoie 对 C++ 标准的深入理解，以及 C++ 先驱 Barbara E. Moo 在 C++ 教学方面的真知灼见外，更是基于全新的 C++11 标准进行了全面而彻底的内容更新。非常难能可贵的是，《C++ Primer 中文版(第5版)》所有示例均全部采用 C++11 标准改写，这在经典升级版中极其罕见——充分体现了 C++ 语言的重大进展及其全面实践。书中丰富的教学辅助内容、醒目的知识点提示，以及精心组织的编程示范，让这本书在 C++ 领域的权威地位更加不可动摇。无论是初学者入门，或是中、高级程序员提升，本书均为不容置疑的首选。

作者简介

Stanley B. Lippman目前是微软公司 Visual C++ 团队的架构师。他从1984年开始在贝尔实验室与C++的设计者Bjarne Stroustrup一起从事C++的设计与开发。他在迪士尼和梦工厂从事动画制作，还担任过JPL的高级顾问。

Jos é e Lajoie曾经是IBM加拿大研究中心C/C++编译器开发团队的成员，在ISO C++标准委员会工作了7年，担任过ISO核心语言工作组的主席和C++ Report杂志的专栏作家。

Barbara E. Moo是拥有25年软件经验的独立咨询顾问。在AT&T，她与Stroustrup、Lippman一起管理过复杂的C++开发项目。

书籍目录

第1章 开始

1

1.1 编写一个简单的C++程序

2

1.1.1 编译、运行程序

3

1.2 初识输入输出

5

1.3 注释简介

8

1.4 控制流

10

1.4.1 while语句

10

1.4.2 for语句

11

1.4.3 读取数量不定的输入数据

13

1.4.4 if语句

15

1.5 类简介

17

1.5.1 Sales_item类

17

1.5.2 初识成员函数

20

1.6 书店程序

21

小结

23

术语表

23

第 部分 C++基础

27

第2章 变量和基本类型

29

2.1 基本内置类型

30

2.1.1 算术类型

30

2.1.2 类型转换

32

2.1.3 字面值常量

35

2.2 变量

38

2.2.1 变量定义

38	
2.2.2	变量声明和定义的关系
41	
2.2.3	标识符
42	
2.2.4	名字的作用域
43	
2.3	复合类型
45	
2.3.1	引用
45	
2.3.2	指针
47	
2.3.3	理解复合类型的声明
51	
2.4	const限定符
53	
2.4.1	const的引用
54	
2.4.2	指针和const
56	
2.4.3	顶层const
57	
2.4.4	constexpr和常量表达式
58	
2.5	处理类型
60	
2.5.1	类型别名
60	
2.5.2	auto类型说明符
61	
2.5.3	decltype类型指示符
62	
2.6	自定义数据结构
64	
2.6.1	定义Sales_data类型
64	
2.6.2	使用Sales_data类
66	
2.6.3	编写自己的头文件
67	
小结	
69	
术语表	
69	
第3章	字符串、向量和数组
73	
3.1	命名空间的using声明
74	

3.2 标准库类型string	75
3.2.1 定义和初始化string对象	76
3.2.2 string对象上的操作	77
3.2.3 处理string对象中的字符	81
3.3 标准库类型vector	86
3.3.1 定义和初始化vector对象	87
3.3.2 向vector对象中添加元素	90
3.3.3 其他vector操作	91
3.4 迭代器介绍	95
3.4.1 使用迭代器	95
3.4.2 迭代器运算	99
3.5 数组	101
3.5.1 定义和初始化内置数组	101
3.5.2 访问数组元素	103
3.5.3 指针和数组	105
3.5.4 C风格字符串	109
3.5.5 与旧代码的接口	111
3.6 多维数组	112
小结	117
术语表	117
第4章 表达式	119
4.1 基础	120
4.1.1 基本概念	120
4.1.2 优先级与结合律	121
4.1.3 求值顺序	

123	
4.2 算术运算符	124
4.3 逻辑和关系运算符	126
4.4 赋值运算符	129
4.5 递增和递减运算符	131
4.6 成员访问运算符	133
4.7 条件运算符	134
4.8 位运算符	135
4.9 sizeof运算符	139
4.10 逗号运算符	140
4.11 类型转换	141
4.11.1 算术转换	142
4.11.2 其他隐式类型转换	143
4.11.3 显式转换	144
4.12 运算符优先级表	147
小结	149
术语表	149
第5章 语句	153
5.1 简单语句	154
5.2 语句作用域	155
5.3 条件语句	156
5.3.1 if语句	156
5.3.2 switch语句	159
5.4 迭代语句	165
5.4.1 while语句	165

5.4.2 传统的for语句	166
5.4.3 范围for语句	168
5.4.4 do while语句	169
5.5 跳转语句	170
5.5.1 break语句	170
5.5.2 continue语句	171
5.5.3 goto语句	172
5.6 TRY语句块和异常处理	172
5.6.1 throw表达式	173
5.6.2 try语句块	174
5.6.3 标准异常	176
小结	178
术语表	178
第6章 函数	181
6.1 函数基础	182
6.1.1 局部对象	184
6.1.2 函数声明	186
6.1.3 分离式编译	186
6.2 参数传递	187
6.2.1 传值参数	187
6.2.2 传引用参数	188
6.2.3 const形参和实参	190
6.2.4 数组形参	193
6.2.5 main：处理命令行选项	196
6.2.6 含有可变形参的函数	

197
6.3 返回类型和return语句
199
6.3.1 无返回值函数
200
6.3.2 有返回值函数
200
6.3.3 返回数组指针
205
6.4 函数重载
206
6.4.1 重载与作用域
210
6.5 特殊用途语言特性
211
6.5.1 默认实参
211
6.5.2 内联函数和constexpr函数
213
6.5.3 调试帮助
215
6.6 函数匹配
217
6.6.1 实参类型转换
219
6.7 函数指针
221
小结
225
术语表
225
第7章 类
227
7.1 定义抽象数据类型
228
7.1.1 设计Sales_data类
228
7.1.2 定义改进的Sales_data类
230
7.1.3 定义类相关的非成员函数
234
7.1.4 构造函数
235
7.1.5 拷贝、赋值和析构
239
7.2 访问控制与封装
240
7.2.1 友元
241

7.3 类的其他特性	243
7.3.1 类成员再探	243
7.3.2 返回*this的成员函数	246
7.3.3 类类型	249
7.3.4 友元再探	250
7.4 类的作用域	253
7.4.1 名字查找与类的作用域	254
7.5 构造函数再探	257
7.5.1 构造函数初始值列表	258
7.5.2 委托构造函数	261
7.5.3 默认构造函数的作用	262
7.5.4 隐式的类类型转换	263
7.5.5 聚合类	266
7.5.6 字面值常量类	267
7.6 类的静态成员	268
小结	273
术语表	273
第 部 C++标准库	275
第8章 IO库	277
8.1 IO类	278
8.1.1 IO对象无拷贝或赋值	279
8.1.2 条件状态	279
8.1.3 管理输出缓冲	281
8.2 文件输入输出	283
8.2.1 使用文件流对象	

284	
8.2.2 文件模式	
286	
8.3 string流	
287	
8.3.1 使用istringstream	
287	
8.3.2 使用ostringstream	
289	
小结	
290	
术语表	
290	
第9章 顺序容器	
291	
9.1 顺序容器概述	
292	
9.2 容器库概览	
294	
9.2.1 迭代器	
296	
9.2.2 容器类型成员	
297	
9.2.3 begin和end成员	
298	
9.2.4 容器定义和初始化	
299	
9.2.5 赋值和swap	
302	
9.2.6 容器大小操作	
304	
9.2.7 关系运算符	
304	
9.3 顺序容器操作	
305	
9.3.1 向顺序容器添加元素	
305	
9.3.2 访问元素	
309	
9.3.3 删除元素	
311	
9.3.4 特殊的forward_list操作	
312	
9.3.5 改变容器大小	
314	
9.3.6 容器操作可能使迭代器失效	
315	
9.4 vector对象是如何增长的	
317	

9.5 额外的string操作	320
9.5.1 构造string的其他方法	321
9.5.2 改变string的其他方法	322
9.5.3 string搜索操作	325
9.5.4 compare函数	327
9.5.5 数值转换	327
9.6 容器适配器	329
小结	332
术语表	332
第10章 泛型算法	335
10.1 概述	336
10.2 初识泛型算法	338
10.2.1 只读算法	338
10.2.2 写容器元素的算法	339
10.2.3 重排容器元素的算法	342
10.3 定制操作	344
10.3.1 向算法传递函数	344
10.3.2 lambda表达式	345
10.3.3 lambda捕获和返回	349
10.3.4 参数绑定	354
10.4 再探迭代器	357
10.4.1 插入迭代器	358
10.4.2 istream迭代器	359
10.4.3 反向迭代器	363
10.5 泛型算法结构	

365	
10.5.1	5类迭代器
365	
10.5.2	算法形参模式
367	
10.5.3	算法命名规范
368	
10.6	特定容器算法
369	
	小结
371	
	术语表
371	
	第11章 关联容器
373	
11.1	使用关联容器
374	
11.2	关联容器概述
376	
11.2.1	定义关联容器
376	
11.2.2	关键字类型的要求
378	
11.2.3	pair类型
379	
11.3	关联容器操作
381	
11.3.1	关联容器迭代器
382	
11.3.2	添加元素
383	
11.3.3	删除元素
386	
11.3.4	map的下标操作
387	
11.3.5	访问元素
388	
11.3.6	一个单词转换的map
391	
11.4	无序容器
394	
	小结
397	
	术语表
397	
	第12章 动态内存
399	
12.1	动态内存与智能指针
400	

12.1.1 shared_ptr类	400
12.1.2 直接管理内存	407
12.1.3 shared_ptr和new结合使用	412
12.1.4 智能指针和异常	415
12.1.5 unique_ptr	417
12.1.6 weak_ptr	420
12.2 动态数组	423
12.2.1 new和数组	423
12.2.2 allocator类	427
12.3 使用标准库：文本查询程序	430
12.3.1 文本查询程序设计	430
12.3.2 文本查询程序类的定义	432
小结	436
术语表	436
第 部分 类设计者的工具	437
第13章 拷贝控制	439
13.1 拷贝、赋值与销毁	440
13.1.1 拷贝构造函数	440
13.1.2 拷贝赋值运算符	443
13.1.3 析构函数	444
13.1.4 三/五法则	447
13.1.5 使用=default	449
13.1.6 阻止拷贝	449
13.2 拷贝控制和资源管理	452
13.2.1 行为像值的类	

453	
13.2.2	定义行为像指针的类
455	
13.3	交换操作
457	
13.4	拷贝控制示例
460	
13.5	动态内存管理类
464	
13.6	对象移动
470	
13.6.1	右值引用
471	
13.6.2	移动构造函数和移动赋值运算符
473	
13.6.3	右值引用和成员函数
481	
	小结
486	
	术语表
486	
	第14章 操作重载与类型转换
489	
14.1	基本概念
490	
14.2	输入和输出运算符
494	
14.2.1	重载输出运算符<<
494	
14.2.2	重载输入运算符>>
495	
14.3	算术和关系运算符
497	
14.3.1	相等运算符
497	
14.3.2	关系运算符
498	
14.4	赋值运算符
499	
14.5	下标运算符
501	
14.6	递增和递减运算符
502	
14.7	成员访问运算符
504	
14.8	函数调用运算符
506	
14.8.1	lambda是函数对象
507	

14.8.2 标准库定义的函数对象	509
14.8.3 可调用对象与function	511
14.9 重载、类型转换与运算符	514
14.9.1 类型转换运算符	514
14.9.2 避免有二义性的类型转换	517
14.9.3 函数匹配与重载运算符	521
小结	523
术语表	523
第15章 面向对象程序设计	525
15.1 OOP：概述	526
15.2 定义基类和派生类	527
15.2.1 定义基类	528
15.2.2 定义派生类	529
15.2.3 类型转换与继承	534
15.3 虚函数	536
15.4 抽象基类	540
15.5 访问控制与继承	542
15.6 继承中的类作用域	547
15.7 构造函数与拷贝控制	551
15.7.1 虚析构函数	552
15.7.2 合成拷贝控制与继承	552
15.7.3 派生类的拷贝控制成员	554
15.7.4 继承的构造函数	557
15.8 容器与继承	558
15.8.1 编写Basket类	

559	
15.9	文本查询程序再探
562	
15.9.1	面向对象的解决方案
563	
15.9.2	Query_base类和Query类
567	
15.9.3	派生类
568	
15.9.4	eval函数
571	
	小结
575	
	术语表
575	
	第16章 模板与泛型编程
577	
16.1	定义模板
578	
16.1.1	函数模板
578	
16.1.2	类模板
583	
16.1.3	模板参数
592	
16.1.4	成员模板
595	
16.1.5	控制实例化
597	
16.1.6	效率与灵活性
599	
16.2	模板实参推断
600	
16.2.1	类型转换与模板类型参数
601	
16.2.2	函数模板显式实参
603	
16.2.3	尾置返回类型与类型转换
604	
16.2.4	函数指针和实参推断
607	
16.2.5	模板实参推断和引用
608	
16.2.6	理解std::move
610	
16.2.7	转发
612	
16.3	重载与模板
614	

16.4 可变参数模板	618
16.4.1 编写可变参数函数模板	620
16.4.2 包扩展	621
16.4.3 转发参数包	622
16.5 模板特例化	624
小结	630
术语表	630
第 部分 高级主题	633
第17章 标准库特殊设施	635
17.1 tuple类型	636
17.1.1 定义和初始化tuple	637
17.1.2 使用tuple返回多个值	638
17.2 BITSET类型	640
17.2.1 定义和初始化bitset	641
17.2.2 bitset操作	643
17.3 正则表达式	645
17.3.1 使用正则表达式库	646
17.3.2 匹配与Regex迭代器类型	650
17.3.3 使用子表达式	653
17.3.4 使用regex_replace	657
17.4 随机数	659
17.4.2 其他随机数分布	663
bernoulli_distribution类	665
17.5 IO库再探	666
17.5.1 格式化输入与输出	

666	
17.5.2	未格式化的输入/输出操作
673	
17.5.3	流随机访问
676	
	小结
680	
	术语表
680	
	第18章 用于大型程序的工具
683	
18.1	异常处理
684	
18.1.1	抛出异常
684	
18.1.2	捕获异常
687	
18.1.3	函数try语句块与构造函数
689	
18.1.4	noexcept异常说明
690	
18.1.5	异常类层次
693	
18.2	命名空间
695	
18.2.1	命名空间定义
695	
18.2.2	使用命名空间成员
701	
18.2.3	类、命名空间与作用域
705	
18.2.4	重载与命名空间
708	
18.3	多重继承与虚继承
710	
18.3.1	多重继承
711	
18.3.2	类型转换与多个基类
713	
18.3.3	多重继承下的类作用域
715	
18.3.4	虚继承
717	
18.3.5	构造函数与虚继承
720	
	小结
722	
	术语表
722	

第19章 特殊工具与技术	725
19.1 控制内存分配	726
19.1.1 重载new和delete	726
19.1.2 定位new表达式	729
19.2 运行时类型识别	730
19.2.1 dynamic_cast运算符	730
19.2.2 typeid运算符	732
19.2.3 使用RTTI	733
19.2.4 type_info类	735
19.3 枚举类型	736
19.4 类成员指针	739
19.4.1 数据成员指针	740
19.4.2 成员函数指针	741
19.4.3 将成员函数用作可调用对象	744
19.5 嵌套类	746
19.6 union：一种节省空间的类	749
19.7 局部类	754
19.8 固有的不可移植的特性	755
19.8.1 位域	756
19.8.2 volatile限定符	757
19.8.3 链接指示：extern "C"	758
小结	762
术语表	762
附录A 标准库	765
A.1 标准库名字和头文件	

766	
A.2 算法概览	
770	
A.2.1 查找对象的算法	
771	
A.2.2 其他只读算法	
772	
A.2.3 二分搜索算法	
772	
A.2.4 写容器元素的算法	
773	
A.2.5 划分与排序算法	
775	
A.2.6 通用重排操作	
776	
A.2.7 排列算法	
778	
A.2.8 有序序列的集合算法	
778	
A.2.9 最小值和最大值	
779	
A.2.10 数值算法	
780	
A.3 随机数	
781	
A.3.1 随机数分布	
781	
A.3.2 随机数引擎	
783	
C++11的新特性	
2.1.1 long long类型	
31	
2.2.1 列表初始化	
39	
2.3.2 nullptr常量	
48	
2.4.4 constexpr变量	
59	
2.5.1 类型别名声明	
60	
2.5.2 auto类型指示符	
61	
2.5.3 decltype类型指示符	
62	
2.6.1 类内初始化	
65	
3.2.2 使用auto或decltype缩写类型	
79	
3.2.3 范围for语句	

82	
3.3	定义vector对象的vector（向量的向量）
87	
3.3.1	vector对象的列表初始化
88	
3.4.1	容器的cbegin和cend函数
98	
3.5.3	标准库begin和end函数
106	
3.6	使用auto和decltype简化声明
115	
4.2	除法的舍入规则
125	
4.4	用大括号包围的值列表赋值
129	
4.9	将sizeof用于类成员
139	
5.4.3	范围for语句
168	
6.2.6	标准库initializer_list类
197	
6.3.2	列表初始化返回值
203	
6.3.3	定义尾置返回类型
206	
6.3.3	使用decltype简化返回类型定义
6.5.2	constexpr函数
214	
7.1.4	使用=default生成默认构造函数
237	
7.3.1	类对象成员的类内初始化
246	
7.5.2	委托构造函数
261	
7.5.6	constexpr构造函数
268	
8.2.1	用string对象处理文件名
284	
9.1	array和forward_list容器
293	
9.2.3	容器的cbegin和cend函数
298	
9.2.4	容器的列表初始化
300	
9.2.5	容器的非成员函数swap
303	
9.3.1	容器insert成员的返回类型
308	
9.3.1	容器的emplace成员的返回类型

308	
9.4	shrink_to_fit
318	
9.5.5	string的数值转换函数
327	
10.3.2	Lambda表达式
346	
10.3.3	Lambda表达式中的尾置返回类型
353	
10.3.4	标准库bind函数
354	
11.2.1	关联容器的列表初始化
377	
11.2.3	列表初始化pair的返回类型
380	
11.3.2	pair的列表初始化
384	
11.4	无序容器
394	
12.1	智能指针
400	
12.1.1	shared_ptr类
12.1.2	动态分配对象的列表初始化
407	
12.1.2	auto和动态分配
408	
12.1.5	unique_ptr类
417	
12.1.6	weak_ptr类
420	
12.2.1	范围for语句不能应用于动态分配数组
424	
12.2.1	动态分配数组的列表初始化
424	
12.2.1	auto不能用于分配数组
424	
12.2.2	allocator::construct可使用任意构造函数
428	
13.1.5	将=default用于拷贝控制成员
449	
13.1.6	使用=default阻止拷贝类对象
449	
13.5	用移动类对象代替拷贝类对象
469	
13.6.1	右值引用
471	
13.6.1	标准库move函数
472	
13.6.2	移动构造函数和移动赋值

473	
13.6.2	移动构造函数通常应该是noexcept
473	
13.6.2	移动迭代器
480	
13.6.3	引用限定成员函数
483	
14.8.3	function类模板
512	
14.9.1	explicit类型转换运算符
516	
15.2.2	虚函数的override指示符
530	
15.2.2	通过定义类为final来阻止继承
533	
15.3	虚函数的override和final指示符
538	
15.7.2	删除的拷贝控制和继承
553	
15.7.4	继承的构造函数
557	
16.1.2	声明模板类型形参为友元
590	
16.1.2	模板类型别名
590	
16.1.3	模板函数的默认模板参数
594	
16.1.5	实例化的显式控制
597	
16.2.3	模板函数与尾置返回类型
605	
16.2.5	引用折叠规则
609	
16.2.6	用static_cast将左值转换为右值
612	
16.2.7	标准库forward函数
614	
16.4	可变参数模板
618	
16.4	sizeof...运算符
619	
16.4.3	可变参数模板与转发
622	
17.1	标准库Tuple类模板
636	
17.2.2	新的bitset运算
643	
17.3	正则表达式库
645	

17.4 随机数库	659
17.5.1 浮点数格式控制	670
18.1.4 noexcept异常指示符	690
18.1.4 noexcept运算符	691
18.2.1 内联名字空间	699
18.3.1 继承的构造函数和多重继承	712
19.3 有作用域的enum	736
19.3 说明类型用于保存enum对象	738
19.3 enum的提前声明	738
19.4.3 标准库mem_fn类模板	746
19.6 类类型的联合成员	751

精彩短评

- 1、已经有第六版了，泛型编程和模板，不用过于考虑实现的编程，关键的是给出了C与C++继承关系
- 2、经典书籍。轮子哥审校质量有保证！读了这本书之后会发现再也找不到如此经典的教材了。介绍了C++的一些常用的函数和原理。深度广度具备。不过想用好C++还是要继续学习。
- 3、C++11是一个较大的转变，算是本书的特色了。虽然C++11中增加了很多好用的特性，但是在我们公司使用C++11的还是很少。
- 4、反复看了几遍，每次都有收获。
- 5、主要看了模板，容器，IO，动态内存
- 6、习题
- 7、读了三分之一，题目还得认真做啊。
- 8、C++11必看
- 9、多读几遍才行。
- 10、把c++问题讲得很仔细，面面俱到，需要重复看的书。
- 11、==字典。待看：动态内存 泛型 IO库。有些概念连续的却跳的很远导致一直跳页码? 黑人问号.jpg
- 12、厉害了我大CPP，泛型编程后面大概看了下，实在是太难了~
- 13、io那部分用到了再看~
- 14、重点看看C++11相关
- 15、补标；C++经典
- 16、CSer最幸福的事就是大量的阅读材料必然带来经典和良心之作
- 17、为什么叫做经典教程书，是因为书的参考是书本身，而不需要其他参考书。佩服作者，也悲哀中国没有能编辑出这种书的人。
- 18、最后还是决定把C++这个坑弃了，非常非常好的一本教材
- 19、我还没读完呢
- 20、这种书，看懂了的话就是钱啊。
- 21、C++ 11是个好语言
- 22、每一个C++程序员的必读书籍，不解释。
- 23、翻译得相当不错，很多点都讲得很好，但是还是快速地过了一遍跑去看Effective三部曲了。
- 24、卧槽这么厚.....我把学校的教材翻了一遍再看都觉得有些吃不消啊，但是学校的教材是叫你用c++设计程序，而这本是用设计的实例来解释c++语言的特性，简单来说，把你变成一个c++编译器.....感觉如果没有一定量的代码量，读起来还是有些困难.....
- 25、好书，还要再读。
- 26、我的C++入门书。这本书给我的感觉和读奥本海默的《信号与系统》的感觉一样——不仅仅是基础技术的讲解，更像是在跟你聊设计的哲学。
- 27、C++ Primer,5th Edition
Stanley B.Lippman
王刚/杨巨峰译
- 28、光是有一本完整的答案（尽管有错），就已经很值得推荐了
- 29、配合其他C++11的资料一起，在了解新特性的同时，加深基础
- 30、跳了一个大坑...读了大半才能懂一点，现在又放下了，估计以后会都忘掉吧2333
- 31、现在都要c11，买了本速成的
- 32、经典
- 33、以后应该会时常翻阅的书
- 34、学校的教材大概就是告诉你什么是面向对象编程，学了也不会用C++。看了中间大半，C++还真是一锅大杂烩，入门不易哈哈
- 35、讲的很细致，很好的c++11介绍书籍
- 36、对此书感激不尽。
- 37、非常不错的书籍 值得细读通读一遍。
- 38、虽然标榜是入门的书，但深度和广度远远超过新手的接受范围。入门还是看看其他的教材，然后

是《effective c++》

39、c++经典书籍

40、经典书籍

41、基础内容很完备，而且c++11特性进行了解释，目录结构清晰，有实例代码分析

42、很不错的经典入门

43、跟着轮子哥的步伐入门

44、很好的书

45、真心不喜欢这本书的风格，完全不顾初学者的心情。

46、入门c++的第一本书，经典厚重，细节性的东西很多。目前读到1/3，已经觉得有些疲倦。建议把此书当做一本字典式的读物，写程序时遇到不明白的地方就去查。这本书需要静下心来去看，边看边敲代码，因为东西太多太杂，乱花渐欲迷人眼。希望有朝一日自己能把这本大部头啃下来。

47、细节太多，还需要看

48、醍醐灌顶！

49、我在上学的时候读过第四版，可惜终归没有目的性很快就读不下去了，还被一位成绩很好的同学评论该书无用，参加工作才发现短板明显，重读起来C++已经和上学时候大不一样了。花了近一年才读完此书，希望有用。

50、无需多说，除了thread相关东西基本上都能覆盖上。必看的神书。

精彩书评

1、《C++ Primer》作为久负盛名的C++经典教程，丰富的教学辅助内容、精心组织的编程示范，无论是初学者入门，或是中、高级程序员提升，都是不容置疑的首选。一本好书只有读过才有价值，然而《C++ Primer》这本厚如砖头的好书，很多人在购买之后，都仅仅只是简单翻阅了前面几个章节，而并未能够认真的完整读完该书。实验楼推出了深入学习《C++ Primer 第五版》训练营，以任务驱动模式学习经典教材，提供专属在线环境，由一名导师全程指导有问必答，带领你7周完成《C++ Primer》第五版的学习，扎实你的C++基础。结合个人经验和网络资源为大家整理了一些建议，分享给大家作为参考，希望大家能够正确的通过《C++ Primer》学习C++。《C++ Primer》该如何学习？1.《C++ Primer》这本书一定要从头开始看，仔细阅读千万别跳，C++11穿插在书中的每一个角落，如果直接跳到后面你可能看到新标准又返回来看。而且这种经典的书籍反复看多遍每次也会有新收获。2.一行一行的把书上的代码手敲一遍，边敲，边看效果，然后思考为什么这样做，这样做的效果是什么，遇到不懂的，再仔细的在书中寻找答案，书中都有很详细的解释。刚开始敲代码会遇到比较多语法错误，积累多了就能够很快改正过来。3.做好学习笔记，书中的内容较多，而且大部分的内容不保证所有都常用，所以有一些忘记的会比较快，所以做好学习笔记快速记忆起来自己看，也可以作为以后查漏补缺的工具。超实用的《C++ Primer》读书指南，以《C++ Primer》第五版为例：Part1也就是前八章，除了6.6，6.7节，都要通读。尤其是第三章初步介绍了vector和string，简直就是新手福音，搞定这两个容器就能写一些简单的程序。Part2基本就是数据结构和算法，如果有基础读起来很轻松。9，11两章介绍的容器，以及12.1节的智能指针要通读。多用智能指针和容器，远离segment fault. 第10章里的泛型算法可以慢慢读，读完以后可以写出高逼格的函数式风格C++。12.2节讲了怎么用new和delete分配空间，题主作为新手，知道这种写法就行，写程序时尽量用容器代替原始数组，尤其是代码里最好不要有delete。Part3是块硬骨头，标题就是Tools for Class Authors. 作为一个“class user”，有些部分第一次是可以略过的。13章很重要，要细读。初始化，复制，赋值，右值引用是C++里很微妙很重要的部分，别的语言对于这些概念很少有区分得这么细的。这一章不但要精读，还要完全掌握。14章的操作符重载第一次可以观其大略；14.9节第一次可以跳过。15章讲OOP，重要性不言而喻。如果之前一点概念都没有，学起来会觉得比较抽象。网上关于OOP有很多通俗有趣的文章，可以一起看看。16章讲泛型编程，第一次读16.1节，掌握最基本的函数模板和类模板就行了。Part4就更高档了，很多内容第一次就算啃下来，长久不用又忘了。第一次读推荐把18.2节读懂，命名空间简单易用效果好。别的内容可以观其大略，用时再看。17.1节的tuple是个有趣的东东，可以读一读。17.3节的正则表达式和17.4节的随机数也许有用，也可以读一读。如果需要读写文件，要读一下17.5.2节的raw I/O和17.5.3节的random I/O。

2、科学计算、分布式应用、嵌入式行业、智能控制、算法研究，乃至学术讨论和上机考试都会有一种语言的身影，那就是大名鼎鼎的C++语言。原汁原味的C++目前已经执到C++11标准。听到这门编程语言，多数人伴随而来的是晦涩、复杂、强大等关键词！这样的感觉实际上是真实的。Lippman（著名C++语言专家，《C++ Primer》，《Essential C++》等多本经典著作作者）自己也认为，C++语言是一门复杂的编程语言。说到这里，不禁想起了《C++ Primer》这本著作，这可谓不折不扣之C++蓝本。最早接触《C++ Primer》这本书是大学时代学习C++语言的时候，当时这本书是学校的教学用书之一。现在回想起来，还得感谢我的C++语言老师，为我们选定了这本书，使得我们学习这门语言有了一个比较好的理论参照。参加工作的最初几年，有空的话会随手翻看这本大部头，不断验证在实践中的所见所闻并吸收新的知识点。已经掌握的知识点，也得到了很好的复习和巩固，印在了脑海里。世上无难事只怕有心人，在学习过程中适当做笔记，以备回顾。今冬伊始，适逢此书新版付梓，并有幸与作译者见面。首先，无论如何要感谢他们为本书所付出的艰辛劳动。已经63岁的Lippman面容清瘦，讲话井井有条，富有激情。闲谈之中显露出严谨诙谐的风格。他说起了cfront（C++编译器），说起了在迪斯尼使用C++语言处理动画过程中遇到的奇闻轶事，说起了在微软工作的日子，他对这些往事如数家珍。当座位旁边一位计算机专业的学生谈起对金融贸易感兴趣的时候，他向这位学生还谈起了D.E.Shaw这位驰骋华尔街的科学狂人。他不仅是一位出色的学者，也是一位循循善诱的长者。而与此版译者王刚教授初次见面，他递过名片，给人的印象低调谦和，席间谈论了许多专业内外的趣事。此版译本邀请了业内知名的C++技术专家承担了审校工作，亦采用最新的C++11标准做了修订。再一次向在出版过程中付出了辛勤汗水的朋友们致敬。虽不至韦编三绝，枕籍经史，但我还会继续品读，

这是一种学习，也是一种修行。

3、说起Lippman的C++ Primer，我总是有种特殊感情。这本书既是我进入C++领域的敲门砖，也是我第一次在网络上发表技术文章的对象。当年读书笔记中的青涩迷惘和年少轻狂都还历历在目，转眼已经从第三版出到了第五版。十年的光阴改变了许多，不变的是经典。C++ Primer一直是我给初学者推荐的首选入门书籍。首先，相对于Bjarne Stroustrup（另一本C++经典The C++ Programming Language的作者）来说，Lippman更倾向于实例化教学，这使得他的书更平易近人。这对于初学者非常重要，他们需要一个好的老师，一个不但能够答疑解惑，而且能够引发学生强烈兴趣的老师。而对于语言的兴趣，必须来自于现实生活中的表达能力，这离不开大量的实例，以及对于这些实例的推演变化。而这正是C++ Primer这本书的专长。其次，需要指出的是，C++是一个多范式的编程语言。从C风格的过程式编程到面向对象编程，再到面向STL的泛型编程，最近又在新的标准中引入了新的函数式编程。可以说C++为我们的计算机表达能力提供了无限的可能。程序员可以根据自己的需要选择合适的编程范式。而能系统性涵盖所有范式的C++教程并不多，C++ Primer是其中之一。当然，C++ 11是不能不提的。这是自C++ 98标准发布以来，C++在语言标准化方面最为重要的修订。它引入了能自动推导类型的auto关键字，引入了智能指针类型和Lambda表达式等。尽管对于这东西的讨论在C++社区里已经持续了很长一段时间了，但基本上都是BBS或者邮件列表中的片言只语，既不全面也缺乏体系性的论述。现在，C++ Primer终于为我们提供了这样一种选择。最后感谢中国电子工业出版社对于这本巨作的及时引进和翻译，技术出版的任务任重而道远，期待你们为业界作出更多的努力，培养出更多，更优秀的程序员。

4、第一次拿到此书的时候还是第4版，是它为我打开了C++的大门。可惜当时没能耐心读完。现在重读此书，每一句话，每行代码，都让人获益匪浅。第五版囊括了C++11中的新特性，这是一次振奋人心的改动，很多不合理的设计得到了纠正，亦从Java，Python等其他语言中吸收了众多魔法糖，而正则表达式，lambda函数的加入则让C++有了更强的能力。作者已边白标注的形式标明了C++11中的新特性，并耐心指出新旧的不同。书中包含了大量的代码示例，虽然短小，随着C++使用经验越发的丰富，方察觉书中代码在平时开发中是那么的常见。辅以详细注释，指明其中的要点与陷阱，可见作者写书时是多么的用心良苦。现在在读effective C++，期间交叉对比primer，尤其是作者特别标注的C++11新特性，对程序设计语言的使用与进化产生了新的认识。待我读完effective系列之后，会再回头重读primer，相信那时此书还能给我带来惊喜。

5、===== 分割线=====相比B.S的the C Plus Plus Programming Language, 这本更接近实践。作者很多地方从编译器角度看语言。新加了c++11内容，配合B.S电子书做入门用。===== 分割线=====

6、《c++ primer 5th》+ vs2013是我学习c++的标配。那时我工作了一年，日子渐渐安稳，还想学些其他的東西。之前对c++的了解仅仅是大学老师讲的那点皮毛，几乎可以算作为零。每天下班回到几平米的群租屋，翻开《c++ primer》，每天不间断，读了大概2个月才读完。期间我从对c++的一点不懂到有所熟悉，虽算不上十分熟稔，但是也罢最困难的第一遍在无人帮助的情况下读完了，很有成就感。这本书让我从0到1，体验到了c++的乐趣，仅此一点我就已经很有收获了。这本书伴我度过了一段充实、安静的好时光，并定令人怀念。谈谈我的阅读感觉：翻译的很好，非常精准，书籍的制作也比较精良，品质很高。

7、翻译真的差的太离谱了，不知道是不是中国人翻译的。难道是我语文没学好？第21页，1.6书店程序那段话，我读了好几遍都没看懂下面的代码到底要干什么。比如：“我们假定每个ISBN书号的所有销售记录在文件中是聚在一起保存的。”这个“聚在一起”是什么意思？？在文件中保存数据的方式太多了，一个聚在一起能表达什么？？如果英语好的人我建议去读英文原版，中文版真的是太难读懂了。

8、一直很讨厌这个C++ primer，当时第四版就不喜欢，如今出了第五版，抱着试一试态度，结果还是很垃圾。比如7.1章节，引入const成员函数时，讲的好生硬好生硬!!! 为何凭空说要吧this绑定到一个常量对象上？而且翻译的也是很拗口，看一句话得要反应半天才明白字面所表达的意思！立马去翻了C++设计者之一的书《C++编程思想》（英文版，据说中文版翻译的不好）里面就讲了const成员函数的来龙去脉：首先是像定义普通类型的const变量一样：const int a = 5; //表示这个a定义后就不想改变了，那么，定义一个const类对象呢？const class Type A; //怎么保证这个对象A里面的数据成员不会变呢？只有设计出一个const成员函数，这个成员函数保证不修改对象的的数据成员。而且，const对

象只能调用这个const成员函数，千万不能调用别的了哦~ 不然报错哦~ 而函数前加的const 仅仅表示函数返回值是个const,满足不了我们设计要求，所以就在函数参数列表后面加个const 表示这是个const成员函数，保证不修改数据成员，甚至在编译阶段发现代码企图修改数据成员，编译器就果断报错！这样的解释才好理解，C++ primer的作者硬是往this 指针上靠，来解释这个const 成员函数，简直就像个小学生牵强附会！！！！

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com