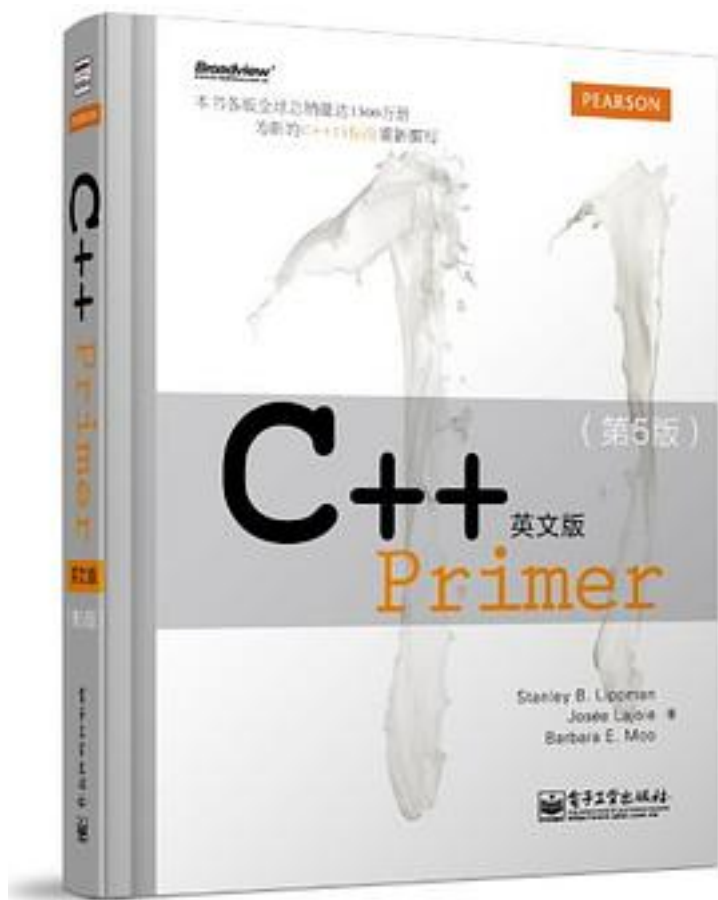


C++ Primer 英文版（第5版）



[C++ Primer 英文版（第5版） 下载链接1](#)

著者:[美] Stanley B. Lippman

出版者:电子工业出版社

出版时间:2013-5-1

装帧:平装

isbn:9787121200380

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Josée 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，书中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++语言的重大进展极其全面实践。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。

作者介绍:

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。

Josée 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。

目录: Preface xxiii

Chapter 1 Getting Started	1
1.1 Writing a Simple C++ Program	2
1.1.1 Compiling and Executing Our Program	3
1.2 A First Look at Input/Output	5
1.3 A Word about Comments	9
1.4 Flow of Control	11
1.4.1 The while Statement	11
1.4.2 The for Statement	13
1.4.3 Reading an Unknown Number of Inputs	14
1.4.4 The if Statement	17
1.5 Introducing Classes	19
1.5.1 The Sales_item Class	20
1.5.2 A First Look at Member Functions	23
1.6 The Bookstore Program	24
Chapter Summary	26
Defined Terms	26
Part I The Basics 29	
Chapter 2 Variables and Basic Types	31
2.1 Primitive Built-in Types	32
2.1.1 Arithmetic Types	32
2.1.2 Type Conversions	35
2.1.3 Literals	38
2.2 Variables	41
2.2.1 Variable Definitions	41
2.2.2 Variable Declarations and Definitions	44
2.2.3 Identifiers	46
2.2.4 Scope of a Name	48
2.3 Compound Types	50
2.3.1 References	50

2.3.2 Pointers	52
vii	
viii Contents	
2.3.3 UnderstandingCompoundTypeDeclarations	57
2.4 constQualifier	59
2.4.1 References to const	61
2.4.2 Pointers and const	62
2.4.3 Top-Level const	63
2.4.4 constexprandConstantExpressions	65
2.5 DealingwithTypes	67
2.5.1 TypeAliases	67
2.5.2 The autoTypeSpecifier	68
2.5.3 The decltypeTypeSpecifier	70
2.6 DefiningOurOwnDataStructures	72
2.6.1 Defining the Sales_dataType	72
2.6.2 Using the Sales_dataClass	74
2.6.3 Writing Our Own Header Files	76
ChapterSummary	78
DefinedTerms	78
Chapter 3 Strings, Vectors, and Arrays	81
3.1 Namespace usingDeclarations	82
3.2 Library stringType	84
3.2.1 Defining and Initializing strings	84
3.2.2 Operations on strings	85
3.2.3 Dealing with the Characters in a string	90
3.3 Library vectorType	96
3.3.1 Defining and Initializing vectors	97
3.3.2 Adding Elements to a vector	100
3.3.3 Other vectorOperations	102
3.4 IntroducingIterators	106
3.4.1 UsingIterators	106
3.4.2 IteratorArithmetic	111
3.5 Arrays	113
3.5.1 DefiningandInitializingBuilt-inArrays	113
3.5.2 AccessingtheElementsof anArray	116
3.5.3 Pointers andArrays	117
3.5.4 C-StyleCharacterStrings	122
3.5.5 InterfacingtoOlderCode	124
3.6 MultidimensionalArrays	125
ChapterSummary	131
DefinedTerms	131
Chapter 4 Expressions	133
4.1 Fundamentals	134
4.1.1 BasicConcepts	134
4.1.2 PrecedenceandAssociativity	136
4.1.3 OrderofEvaluation	137
4.2 ArithmeticOperators	139
4.3 Logical andRelationalOperators	141
Contents ix	
4.4 AssignmentOperators	144
4.5 Increment andDecrementOperators	147
4.6 TheMemberAccessOperators	150
4.7 TheConditionalOperator	151
4.8 TheBitwiseOperators	152

4.9 The sizeof Operator	156
4.10 Comma Operator	157
4.11 Type Conversions	159
4.11.1 The Arithmetic Conversions	159
4.11.2 Other Implicit Conversions	161
4.11.3 Explicit Conversions	162
4.12 Operator Precedence Table	166
Chapter Summary	168
Defined Terms	168
Chapter 5 Statements	171
5.1 Simple Statements	172
5.2 Statement Scope	174
5.3 Conditional Statements	174
5.3.1 The if Statement	175
5.3.2 The switch Statement	178
5.4 Iterative Statements	183
5.4.1 The while Statement	183
5.4.2 Traditional for Statement	185
5.4.3 Range for Statement	187
5.4.4 The do while Statement	189
5.5 Jump Statements	190
5.5.1 The break Statement	190
5.5.2 The continue Statement	191
5.5.3 The goto Statement	192
5.6 try Blocks and Exception Handling	193
5.6.1 A throw Expression	193
5.6.2 The try Block	194
5.6.3 Standard Exceptions	197
Chapter Summary	199
Defined Terms	199
Chapter 6 Functions	201
6.1 Function Basics	202
6.1.1 Local Objects	204
6.1.2 Function Declarations	206
6.1.3 Separate Compilation	207
6.2 Argument Passing	208
6.2.1 Passing Arguments by Value	209
6.2.2 Passing Arguments by Reference	210
6.2.3 const Parameters and Arguments	212
6.2.4 Array Parameters	214
x Contents	
6.2.5 main: Handling Command-Line Options	218
6.2.6 Functions with Varying Parameters	220
6.3 Return Types and the return Statement	222
6.3.1 Functions with No Return Value	223
6.3.2 Functions That Return a Value	223
6.3.3 Returning a Pointer to an Array	228
6.4 Overloaded Functions	230
6.4.1 Overloading and Scope	234
6.5 Features for Specialized Uses	236
6.5.1 Default Arguments	236
6.5.2 Inline and constexpr Functions	238
6.5.3 Aids for Debugging	240
6.6 Function Matching	242

6.6.1 ArgumentTypeConversions	245
6.7 Pointers toFunctions	247
ChapterSummary	251
DefinedTerms	251
Chapter 7 Classes	253
7.1 DefiningAbstractDataTypes	254
7.1.1 Designing the Sales_dataClass	254
7.1.2 Defining the Revised Sales_dataClass	256
7.1.3 DefiningNonmemberClass-RelatedFunctions	260
7.1.4 Constructors	262
7.1.5 Copy,Assignment, andDestruction	267
7.2 AccessControl andEncapsulation	268
7.2.1 Friends	269
7.3 AdditionalClassFeatures	271
7.3.1 ClassMembersRevisited	271
7.3.2 Functions That Return *this	275
7.3.3 ClassTypes	277
7.3.4 FriendshipRevisited	279
7.4 ClassScope	282
7.4.1 NameLookupandClassScope	283
7.5 ConstructorsRevisited	288
7.5.1 Constructor InitializerList	288
7.5.2 DelegatingConstructors	291
7.5.3 TheRoleof theDefaultConstructor	293
7.5.4 ImplicitClass-TypeConversions	294
7.5.5 AggregateClasses	298
7.5.6 LiteralClasses	299
7.6 staticClassMembers	300
ChapterSummary	305
DefinedTerms	305
Contents xi	
Part II The C++ Library 307	
Chapter 8 The IO Library	309
8.1 The IOClasses	310
8.1.1 NoCopyorAssignfor IOObjects	311
8.1.2 ConditionStates	312
8.1.3 ManagingtheOutputBuffer	314
8.2 File Input and Output	316
8.2.1 Using File Stream Objects	317
8.2.2 File Modes	319
8.3 stringStreams	321
8.3.1 Using an istringstream	321
8.3.2 Using ostringstreams	323
ChapterSummary	324
DefinedTerms	324
Chapter 9 Sequential Containers	325
9.1 Overviewof the SequentialContainers	326
9.2 ContainerLibraryOverview	328
9.2.1 Iterators	331
9.2.2 ContainerTypeMembers	332
9.2.3 begin and endMembers	333
9.2.4 DefiningandInitializingaContainer	334
9.2.5 Assignment and swap	337
9.2.6 ContainerSizeOperations	340

9.2.7 Relational Operators	340
9.3 Sequential Container Operations	341
9.3.1 Adding Elements to a Sequential Container	341
9.3.2 Accessing Elements	346
9.3.3 Erasing Elements	348
9.3.4 Specialized forward_list Operations	350
9.3.5 Resizing a Container	352
9.3.6 Container Operations May Invalidate Iterators	353
9.4 How a vector Grows	355
9.5 Additional string Operations	360
9.5.1 Other Ways to Construct strings	360
9.5.2 Other Ways to Change a string	361
9.5.3 string Search Operations	364
9.5.4 The compare Functions	366
9.5.5 Numeric Conversions	367
9.6 Container Adaptors	368
Chapter Summary	372
Defined Terms	372
xii Contents	
Chapter 10 Generic Algorithms	375
10.1 Overview	376
10.2 A First Look at the Algorithms	378
10.2.1 Read-Only Algorithms	379
10.2.2 Algorithms That Write Container Elements	380
10.2.3 Algorithms That Reorder Container Elements	383
10.3 Customizing Operations	385
10.3.1 Passing a Function to an Algorithm	386
10.3.2 Lambda Expressions	387
10.3.3 Lambda Captures and Returns	392
10.3.4 Binding Arguments	397
10.4 Revisiting Iterators	401
10.4.1 Insert Iterators	401
10.4.2 iostream Iterators	403
10.4.3 Reverse Iterators	407
10.5 Structure of Generic Algorithms	410
10.5.1 The Five Iterator Categories	410
10.5.2 Algorithm Parameter Patterns	412
10.5.3 Algorithm Naming Conventions	413
10.6 Container-Specific Algorithms	415
Chapter Summary	417
Defined Terms	417
Chapter 11 Associative Containers	419
11.1 Using an Associative Container	420
11.2 Overview of the Associative Containers	423
11.2.1 Defining an Associative Container	423
11.2.2 Requirements on Key Type	424
11.2.3 The pair Type	426
11.3 Operations on Associative Containers	428
11.3.1 Associative Container Iterators	429
11.3.2 Adding Elements	431
11.3.3 Erasing Elements	434
11.3.4 Subscripting a map	435
11.3.5 Accessing Elements	436
11.3.6 A Word Transformation Map	440

11.4 TheUnorderedContainers	443
ChapterSummary	447
DefinedTerms	447
Chapter 12 DynamicMemory	449
12.1 DynamicMemoryandSmartPointers	450
12.1.1 The shared_ptrClass	450
12.1.2 ManagingMemoryDirectly	458
12.1.3 Using shared_ptrs with new	464
12.1.4 SmartPointers andExceptions	467
12.1.5 unique_ptr	470
Contents xiii	
12.1.6 weak_ptr	473
12.2 DynamicArrays	476
12.2.1 newandArrays	477
12.2.2 The allocatorClass	481
12.3 UsingtheLibrary:AText-QueryProgram	484
12.3.1 Designof theQueryProgram	485
12.3.2 DefiningtheQueryProgramClasses	487
ChapterSummary	491
DefinedTerms	491
Part III Tools for Class Authors	493
Chapter 13 Copy Control	495
13.1 Copy,Assign, andDestroy	496
13.1.1 TheCopyConstructor	496
13.1.2 TheCopy-AssignmentOperator	500
13.1.3 TheDestructor	501
13.1.4 TheRuleofThree/Five	503
13.1.5 Using = default	506
13.1.6 PreventingCopies	507
13.2 CopyControl andResourceManagement	510
13.2.1 ClassesThatActLikeValues	511
13.2.2 DefiningClassesThatActLikePointers	513
13.3 Swap	516
13.4 ACopy-ControlExample	519
13.5 ClassesThatManageDynamicMemory	524
13.6 MovingObjects	531
13.6.1 RvalueReferences	532
13.6.2 MoveConstructor andMoveAssignment	534
13.6.3 RvalueReferencesandMemberFunctions	544
ChapterSummary	549
DefinedTerms	549
Chapter 14 Overloaded Operations and Conversions	551
14.1 BasicConcepts	552
14.2 Input andOutputOperators	556
14.2.1 Overloading the Output Operator <<	557
14.2.2 Overloading the Input Operator >>	558
14.3 Arithmetic andRelationalOperators	560
14.3.1 EqualityOperators	561
14.3.2 RelationalOperators	562
14.4 AssignmentOperators	563
14.5 SubscriptOperator	564
14.6 Increment andDecrementOperators	566
14.7 MemberAccessOperators	569
14.8 Function-CallOperator	571

xiv	Contents	
14.8.1	LambdasAreFunctionObjects	572
14.8.2	Library-DefinedFunctionObjects	574
14.8.3	Callable Objects and function	576
14.9	Overloading,Conversions, andOperators	579
14.9.1	ConversionOperators	580
14.9.2	AvoidingAmbiguousConversions	583
14.9.3	FunctionMatchingandOverloadedOperators	587
	ChapterSummary	590
	DefinedTerms	590
	Chapter 15 Object-Oriented Programming	591
15.1	OOP:AnOverview	592
15.2	DefiningBaseandDerivedClasses	594
15.2.1	DefiningaBaseClass	594
15.2.2	DefiningaDerivedClass	596
15.2.3	Conversions andInheritance	601
15.3	VirtualFunctions	603
15.4	AbstractBaseClasses	608
15.5	AccessControl andInheritance	611
15.6	ClassScopeunder Inheritance	617
15.7	Constructors andCopyControl	622
15.7.1	VirtualDestructors	622
15.7.2	SynthesizedCopyControl andInheritance	623
15.7.3	Derived-ClassCopy-ControlMembers	625
15.7.4	InheritedConstructors	628
15.8	Containers andInheritance	630
15.8.1	Writing a BasketClass	631
15.9	TextQueriesRevisited	634
15.9.1	AnObject-OrientedSolution	636
15.9.2	The Query_base and QueryClasses	639
15.9.3	TheDerivedClasses	642
15.9.4	The evalFunctions	645
	ChapterSummary	649
	DefinedTerms	649
	Chapter 16 Templates and Generic Programming	651
16.1	DefiningaTemplate	652
16.1.1	FunctionTemplates	652
16.1.2	ClassTemplates	658
16.1.3	TemplateParameters	668
16.1.4	MemberTemplates	672
16.1.5	Controlling Instantiations	675
16.1.6	Efficiency and Flexibility	676
16.2	TemplateArgumentDeduction	678
16.2.1	Conversions andTemplateTypeParameters	679
16.2.2	Function-TemplateExplicitArguments	681
16.2.3	Trailing Return Types and Type Transformation	683
	Contents xv	
16.2.4	FunctionPointers andArgumentDeduction	686
16.2.5	TemplateArgumentDeductionandReferences	687
16.2.6	Understanding std::move	690
16.2.7	Forwarding	692
16.3	OverloadingandTemplates	694
16.4	VariadicTemplates	699
16.4.1	WritingaVariadicFunctionTemplate	701

16.4.2 PackExpansion	702
16.4.3 ForwardingParameterPacks	704
16.5 Template Specializations	706
ChapterSummary	713
DefinedTerms	713
Part IV Advanced Topics	715
Chapter 17 Specialized Library Facilities	717
17.1 The tupleType	718
17.1.1 Defining and Initializing tuples	718
17.1.2 Using a tuple toReturnMultipleValues	721
17.2 The bitsetType	723
17.2.1 Defining and Initializing bitsets	723
17.2.2 Operations on bitsets	725
17.3 RegularExpressions	728
17.3.1 UsingtheRegularExpressionLibrary	729
17.3.2 TheMatchandRegex IteratorTypes	734
17.3.3 UsingSubexpressions	738
17.3.4 Using regex_replace	741
17.4 RandomNumbers	745
17.4.1 Random-NumberEngines andDistribution	745
17.4.2 OtherKinds ofDistributions	749
17.5 The IOLibraryRevisited	752
17.5.1 FormattedInput andOutput	753
17.5.2 UnformattedInput/OutputOperations	761
17.5.3 RandomAccess toaStream	763
ChapterSummary	769
DefinedTerms	769
Chapter 18 Tools for Large Programs	771
18.1 ExceptionHandling	772
18.1.1 ThrowinganException	772
18.1.2 CatchinganException	775
18.1.3 Function tryBlocks andConstructors	777
18.1.4 The noexceptExceptionSpecification	779
18.1.5 ExceptionClassHierarchies	782
18.2 Namespaces	785
18.2.1 NamespaceDefinitions	785
xvi Contents	
18.2.2 UsingNamespaceMembers	792
18.2.3 Classes,Namespaces,andScope	796
18.2.4 OverloadingandNamespaces	800
18.3 Multiple andVirtual Inheritance	802
18.3.1 Multiple Inheritance	803
18.3.2 Conversions andMultipleBaseClasses	805
18.3.3 ClassScopeunderMultiple Inheritance	807
18.3.4 Virtual Inheritance	810
18.3.5 Constructors andVirtual Inheritance	813
ChapterSummary	816
DefinedTerms	816
Chapter 19 Specialized Tools and Techniques	819
19.1 Controlling Memory Allocation	820
19.1.1 Overloading new and delete	820
19.1.2 Placement newExpressions	823
19.2 Run-TimeTypeIdentification	825
19.2.1 The dynamic_castOperator	825

19.2.2 The typeidOperator	826
19.2.3 UsingRTTI	828
19.2.4 The type_infoClass	831
19.3 Enumerations	832
19.4 Pointer toClassMember	835
19.4.1 Pointers toDataMembers	836
19.4.2 Pointers toMemberFunctions	838
19.4.3 UsingMemberFunctions asCallableObjects	841
19.5 NestedClasses	843
19.6 union:ASpace-SavingClass	847
19.7 LocalClasses	852
19.8 InherentlyNonportableFeatures	854
19.8.1 Bit-fields	854
19.8.2 volatileQualifier	856
19.8.3 Linkage Directives: extern "C"	857
ChapterSummary	862
DefinedTerms	862
Appendix A The Library	865
A.1 LibraryNames andHeaders	866
A.2 ABriefTourof theAlgorithms	870
A.2.1 Algorithms toFindanObject	871
A.2.2 OtherRead-OnlyAlgorithms	872
A.2.3 BinarySearchAlgorithms	873
A.2.4 AlgorithmsThatWriteContainerElements	873
A.2.5 PartitioningandSortingAlgorithms	875
A.2.6 GeneralReorderingOperations	877
A.2.7 PermutationAlgorithms	879
A.2.8 SetAlgorithms forSortedSequences	880
Contents xvii	
A.2.9 MinimumandMaximumValues	880
A.2.10 NumericAlgorithms	881
A.3 RandomNumbers	882
A.3.1 RandomNumberDistributions	883
A.3.2 RandomNumberEngines	884
Index	
New Features in C++11	
2.1.1 long longType	33
2.2.1 List Initialization	43
2.3.2 nullptrLiteral	54
2.4.4 constexprVariables	66
2.5.1 TypeAliasDeclarations	68
2.5.2 The autoTypeSpecifier	68
2.5.3 The decltypeTypeSpecifier	70
2.6.1 In-Class Initializers	73
3.2.2 Using auto or decltype forTypeAbbreviation	88
3.2.3 Range forStatement	91
3.3 Defining a vector of vectors	97
3.3.1 List Initialization for vectors	98
3.4.1 Container cbegin and cendFunctions	109
3.5.3 Library begin and endFunctions	118
3.6 Using auto or decltype to SimplifyDeclarations	129
4.2 RoundingRules forDivision	141
4.4 Assignment fromaBracedListofValues	145
4.9 sizeofAppliedtoaClassMember	157

5.4.3 Range forStatement	187
6.2.6 Library initializer_listClass	220
6.3.2 List InitializingaReturnValue	226
6.3.3 Declaring a Trailing Return Type	229
6.3.3 Using decltype to Simplify Return Type Declarations	230
6.5.2 constexprFunctions	239
7.1.4 Using = default toGenerateaDefaultConstructor	265
7.3.1 In-class Initializers forMembersofClassType	274
7.5.2 DelegatingConstructors	291
7.5.6 constexprConstructors	299
8.2.1 Using strings for File Names	317
9.1 The array and forward_listContainers	327
9.2.3 Container cbegin and cendFunctions	334
9.2.4 List InitializationforContainers	336
9.2.5 Container Nonmember swapFunctions	339
9.3.1 Return Type for Container insertMembers	344
9.3.1 Container emplaceMembers	345
xix	
xx New Features in C++11	
9.4 shrink_to_fit	357
9.5.5 Numeric Conversion Functions for strings	367
10.3.2 LambdaExpressions	388
10.3.3 Trailing Return Type in Lambda Expressions	396
10.3.4 The Library bindFunction	397
11.2.1 List Initializationof anAssociativeContainer	423
11.2.3 List Initializing pairReturnType	427
11.3.2 List Initialization of a pair	431
11.4 TheUnorderedContainers	443
12.1 SmartPointers	450
12.1.1 The shared_ptrClass	450
12.1.2 List InitializationofDynamicallyAllocatedObjects	459
12.1.2 autoandDynamicAllocation	459
12.1.5 The unique_ptrClass	470
12.1.6 The weak_ptrClass	473
12.2.1 Range for Doesn't Apply to Dynamically Allocated Arrays .	477
12.2.1 List InitializationofDynamicallyAllocatedArrays	478
12.2.1 autoCan'tBeUsedtoAllocateanArray	478
12.2.2 allocator::constructCanUseanyConstructor	482
13.1.5 Using = default forCopy-ControlMembers	506
13.1.6 Using = delete toPreventCopyingClassObjects	507
13.5 MovingInsteadofCopyingClassObjects	529
13.6.1 RvalueReferences	532
13.6.1 The Library moveFunction	533
13.6.2 MoveConstructor andMoveAssignment	534
13.6.2 Move Constructors Usually Should Be noexcept	535
13.6.2 MoveIterators	543
13.6.3 ReferenceQualifiedMemberFunctions	546
14.8.3 The functionClassTemplate	577
14.9.1 explicitConversionOperators	582
15.2.2 overrideSpecifier forVirtualFunctions	596
15.2.2 Preventing Inheritance by Defining a Class as final	600
15.3 override and final Specifiers for Virtual Functions	606
15.7.2 DeletedCopyControl andInheritance	624
15.7.4 InheritedConstructors	628

16.1.2 Declaring a Template Type Parameter as a Friend	666
16.1.2 Template Type Aliases	666
16.1.3 Default Template Arguments for Template Functions	670
16.1.5 Explicit Control of Instantiation	675
16.2.3 Template Functions and Trailing Return Types	684
16.2.5 Reference Collapsing Rules	688
16.2.6 static_cast from an Lvalue to an Rvalue	691
16.2.7 The Library forward Function	694
16.4 Variadic Templates	699
16.4 The sizeof... Operator	700
16.4.3 Variadic Templates and Forwarding	704
New Features in C++11 xxi	
17.1 The Library Tuple Class Template	718
17.2.2 New bitset Operations	726
17.3 The Regular Expression Library	728
17.4 The Random Number Library	745
17.5.1 Floating-Point Format Control	757
18.1.4 The noexcept Exception Specifier	779
18.1.4 The noexcept Operator	780
18.2.1 Inline Namespaces	790
18.3.1 Inherited Constructors and Multiple Inheritance	804
19.3 Scoped enums	832
19.3 Specifying the Type Used to Hold an enum	834
19.3 Forward Declarations for enums	834
19.4.3 The Library mem_fn Class Template	843
19.6 Union Members of Class Types	848
• • • • • (收起)	

[C++ Primer 英文版（第 5 版） 下载链接1](#)

标签

C++

编程

Programming

C/C++

计算机

编程语言

计算机科学

英文原版

评论

大概看五六遍了吧。之后工作要实打实用C/C++，回家又刷一遍，意外发现还有静心功效。C++如此逆天，读了这本七八百页，才算刚刚入门。需要区分pointer，const pointer，pointer to const（Python：什么是pointer，什么是const？），需要在类里构造，析构，复制，赋值和右值引用，需要new和delete（Python：我就一个__init__），需要自己造各种轮子（Python：看看我的库），都是血和泪的坑。配合More Effective C++服用，疗效更佳

可曾想过我居然会读这本书...

C++11 大法好。template 是黑魔法。rvalue reference 平添多少烦恼……

我觉得 c++ 真的有点过了 所以现在 java 的很多特性都很受欢迎 刚刚好解决问题 刚刚好够用 typescript 和 go 的语言特性设计也或多或少参考了 java scalar在 java 基础上 向更易用 开发效率更高迈进 ==

终于在C++14推出之后读过介绍C++11的第五版了

对于C++这种复杂的语言来说，算是很平易近人的书了。讲解很清楚；很早就引入了标准库的介绍，并且有很多Best Practices的小提醒，免得把C++写成C with class；与时俱进全面介绍了C++11特性。但是，好厚啊。。。

awesome, especially C++11 features is well introduced.

英文原本，彩页印刷，书籍经典。

水太深，呛数口，喝饱

C++的语义无甚出奇之处，但是继承自C的语法已经被其搞的很奇葩了。

又看了一遍英文版的，收获颇丰，真是常看常新，每次都会有不同的收获

第一遍读，感觉对我这样的初学者很有帮助，实例也很多

C++经典重读

c++必读，1000+页的庞然大物.
引入了C++11的新特性，介绍的比较分散. 回答了C++语法是怎么样的问题. 但对于应该如何写C++, 还需要别的经典C++书补充.
C++11还是很有必要的，正如作者说的，让C++看起来像一个新的语言

改写了挺多细节，话说point那章真是改了又改。。。我司何时才能用上C++ 11呢

作为C++ Premier Plus的补充，提供另一个理解的视角。

详实，需多读

有点厚

温故而知新，一直零零散散阅读了不少c++0x的特性，这次终于系统的看了一下了。

- C++-Primer-ed5-c++11: 目前为止看到的最厚的技术电子书, c++ 的内容实在太多了

[C++ Primer 英文版（第 5 版）_下载链接1](#)

书评

说实话，我学完C以后进阶到C++的第一本书就是这本CP4。CP4确实是经典，但是它却不适合初学者，我当时读了大概有一半了，但是对很多东西还是一知半解。因为这本书是站在编译器的角度来写的，很多东西非常细节，事无巨细的论述对初学者一点好处也没有。初学者需要快速能先吸引眼...

那个时候不知道大学期间应该找个女朋友，不知道应该出去跟同学喝酒唱歌，不知道还有很多更诱人的技术，我只知道我现在要学c++,而且我手头这本书，是个大砖头，有1000多页，英文版，不争分夺秒看的话，一学期肯定看不完。
学期一开始的时候，我就握着这本书，拿出一张纸制定我...

更新： 不少同学要代码。我折腾了一晚上，终于把习题的代码发到github上了：
<https://github.com/Mooophy/Cpp-Primer>
要讨论的同学要熟悉一下git的用法，需要以下步骤： 1、注册帐号； 2、fork

我的repository； 3、在你自己的repository里commit，比如填加个注释什么的； 4、 ...

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...

这本书的第四版前言里有一段对第三版的评论，原文记不得了，但大意是第三版讨论了很多很深的原理，和书名不符。所以第四版基本上完全重写了，和第三版可以算是两本不同的书。确实，第三版不适合初学者，反而是第四版，有了些入门的意思。不过第三版还是很经典，很值得看的， ...

之前太穷，没有名著的概念。于是就借着国内的几个草根版本的C++教程学编程，结果倒是还不错，会用C++了。

两年前找工作，我对C++的很多要点都没有概念。为了能够应付笔试，去网上找了很多面筋，也在很多公司去体验笔试的感觉。路虽然走得有点偏，但是和凑合。最...

这本书有很多typos，让人抓狂！！！！！！！！

下面好像是官方的勘误表，好久没更新了，只纠正了部分错误

http://ptgmedia.pearsoncmg.com/images/9780201721485/Errata/Lippman_0201721481_Errata.html 这个是某网友找到的错误

<http://wwliu.is-programmer.com/posts/22784> ...

1. 这本书是2010年再版的，麻烦编辑们能不能看看原书的勘误表呢？不要只是重印好吗？

2.同上，原书已经出第五版了，好多错误都改过来了。这本特别版还是没有任何改变。

3.对于这样一本经典读物，翻译需要极端准确的。我希望中文版也能成为经典，而不是语言难懂，有些地方还翻...

有阵子没摸C++，也不太了解C++11，就买了这本经典，大概一个月的茶余饭后读了一遍，确实好。这本书讲的很细致，比如container, algorithm-iterato的关系，条理清楚。
如前面评论所述，很多C++入门书籍是先讲C，再加入C++的内容。这本书把过时的东西放到边边角角，直接讲C++...

还没有读完，但是，一开始就讲vector和iterator让我有些吃惊。但看过之后，就发现了，C++就应该这么讲。而不是像国内的教材那样，学C++都把C给先讲一遍，讲完数组讲指针，搞得很多人最后学的还是C，甚至很多计算机专业的人不知道vector是什么东西。

世界上没有任何一本书，你读完了之后就能完全明白C++。C++究竟是个什么东西？一种语言、一种工具、一些聪明人编造的编程规范。如果说世界上最权威的、最全面的C++读物，那就该是ISO的C++手册，上千页，看也看不完，而且不同的C++编译器根本实现程度不同，所以C++的ISO在某种...

零零散散的读了半年多终于看完了，之前没有系统地学过C++，做项目时对很多东西一知半解，所以坚持学习了这本1000多页的大全。书中的内容覆盖面很广，从基本变量到类模版，知识点很细碎，看过之后发现自己之前对C++就好像没学过一样。书中的习题一定要做，作为检验学过章节的测...

首先很厚 900 多页，包含交叉索引，原版的书纸的质量不用说了。基本覆盖了C++11的内容，风格例子都没有太大变化，加入了一个图标机制，难以理解的部分都加了图片，对于初学者可以略过，对于有经验的可以着重理解。随书还附赠了45天免费的在线电子版。safari在线书店。

这是本非常经典的C++书籍，系统全面地介绍了C++语言。本书有一千多页，可以看成是学习C++的百科全书。市面上没有哪本其他书籍比本书论述得更广泛了，几乎所有的C++知识点本书都有论述而且写得还不错。因为这点，这本书值得每一位C++程序员拥有。我是通过两本书籍来学习C++的...

很不错的书，上学时候由于c++没有学好，想买来补考的，因为这本书让我迈入了编程的行列，一直受用至今，书讲的很细，在学校很多知识都一掠而过了，但是这些知识在使用的时候还是有很大的作用的。很好的入门图书。

[C++ Primer 英文版（第 5 版）_下载链接1](#)