

# 《深入理解C++11》

## 图书基本信息

书名：《深入理解C++11》

13位ISBN编号：9787111426608

10位ISBN编号：7111426606

出版时间：2013-6

出版社：Michael Wong、IBM XL编译器中国开发团队 机械工业出版社 (2013-06出版)

作者：Michael Wong, IBM XL编译器中国开发团队

页数：328

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](http://www.tushu000.com)

## 前言

前言为什么要写这本书相比其他语言的频繁更新，C++语言标准已经有十多年没有真正更新过了。而上一次标准制定，正是面向对象概念开始盛行的时候。较之基于过程的编程语言，基于面向对象、泛型编程等概念的C++无疑是非常先进的，而C++98标准的制定以及各种符合标准的编译器的出现，又在客观上推动了编程方法的革命。因此在接下来的很多年中，似乎人人都在学习并使用C++。商业公司在邀请C++专家为程序员讲课，学校里老师在为学生绘声绘色地讲解面向对象编程，C++的书籍市场也是百花齐放，论坛、BBS的C++板块则充斥了大量各种关于C++的讨论。随之而来的，招聘启事写着“要求熟悉C++编程”，派生与继承成为了面试官审视毕业生基础知识的重点。凡此种种，不一而足。于是C++语言“病毒性”地蔓延到各种编程环境，成为了使用最为广泛的编程语言之一。十来年的时光转瞬即逝，各种编程语言也在快马加鞭地向前发展。如今流行的编程语言几乎无一不支持面向对象的概念。即使是古老的语言，也通过了制定新标准，开始支持面向对象编程。随着Web开发、移动开发逐渐盛行，一些新流行起来的编程语言，由于在应用的快速开发、调试、部署上有着独特的优势，逐渐成为了这些新领域中的主流。不过这并不意味着C++正在失去其阵地。身为C的“后裔”，C++继承了C能够进行底层操作的特性，因此，使用C/C++编写的程序往往具有更佳运行时性能。在构建包括操作系统的各种软件层，以及构建一些对性能要求较高的应用程序时，C/C++往往是最佳选择。更一般地讲，即使是由其他语言编写的程序，往往也离不开由C/C++编写的编译器、运行库、操作系统，或者虚拟机等提供支持。因此，C++已然成为了编程技术中的中流砥柱。如果用个比喻来形容C++，那么可以说这十年来C++正是由“锋芒毕露”的青年时期走向“成熟稳重”的中年时期。不过十年来对于编程语言来说也是个很长的时间，长时间的沉寂甚至会让有的人认为，C++就是这样一种语言：特性稳定，性能出色，易于学习而难于精通。长时间使用C++的程序员也都熟悉了C++毛孔里每一个特性，甚至是现实上的一些细微的区别，比如各种编译器对C++扩展的区别，也都熟稔于心。于是这个时候，C++11标准的横空出世，以及C++之父Bjarne Stroustrup的一句“看起来像一门新语言”的说法，无疑让很多C++程序员有些诚惶诚恐：C++11是否又带来了编程思维的革命？C++11是否保持了对C++98及C的兼容？旧有的C++程序到了C++11是否需要被推倒重来？事实上这些担心都是多余的。相比于C++98带来的面向对象的革命性，C++11带来的却并非“翻天覆地”式的改变。很多时候，程序员保持着“C++98式”的观点来看待C++11代码也同样是合理的。因为在编程思想上，C++11依然遵从了一贯的面向对象的思想，并深入加强了泛型编程的支持。从我们的观察来看，C++11更多的是对步入“成熟稳重”的中年时期的C++的一种改造。比如，像auto类型推导这样的新特性，展现出的是语言的亲和力；而右值引用、移动语义的特性，则着重于改变一些使用C++程序库时容易发生的性能不佳的状况。当然，C++11中也有局部的创新，比如lambda函数的引入，以及原子类型的设计等，都体现了语言与时俱进的活力。语言的诸多方面都在C++11中再次被锤炼，从而变得更加合理、更加条理清晰、更加易用。C++11对C++语言改进的每一点，都呈现出了经过长时间技术沉淀的编程语言的特色与风采。所以从这个角度上看，学习C++11与C++98在思想上是一脉相承的，程序员可以用较小的代价对C++的知识进行更新换代。而在现实中，只要修改少量已有代码（甚至不修改），就可以使用C++11编译器对旧有代码进行升级编译而获得新标准带来的好处，这也非常具有实用性。因此，从很多方面来看，C++程序员都应该乐于升级换代已有的知识，而学习及使用C++11也正是大势所趋。在本书开始编写的时候，C++11标准刚刚发布一年，而本书出版的时候，C++11也只不过才诞生了两年。这一两年，各个编译器厂商或者组织都将支持C++11新特性作为了一项重要工作。不过由于C++11的语言特性非常的多，因此本书在接近完成时，依然没有一款编译器支持C++11所有的特性。但从从业者的角度看，C++11迟早会普及，也迟早会成为C++程序员的首选，因此即使现阶段编译器对C++新特性的支持还不充分，但还是有必要在这个时机推出一本全面介绍C++11新特性的中文图书。希望通过这样的图书，使得更多的中国程序员能够最快地了解C++11新语言标准的方方面面，并且使用最新的C++11编译器来从各方面提升自己编写的C++程序。读者对象本书针对的对象是已经学习过C++，并想进一步学习、了解C++11的程序员。这里我们假定读者已经具备了基本的C++编程知识，并掌握了一定的C++编程技巧（对于C++的初学者来说，本书阅读起来会有一定的难度）。通过本书，读者可以全面而详细地了解C++11对C++进行的改造。无论是试图进行更加精细的面向对象程序编写，或是更加容易地进行泛型编程，或是更加轻松地改造使用程序库等，读者都会发现C++11提供了更好的支持。本书作者和书籍支持本书的作者都是编译器行业的从业者

，主要来自于IBM XL编译器中国开发团队。IBM XL编译器中国开发团队创立于2010年，拥有编译器前端、后端、性能分析、测试等各方面的人员，工作职责涵盖了IBM XL C/C++及IBM XL Fortran编译器的开发、测试、发布等与编译器产品相关的方方面面。虽然团队成立时间不长，成员却都拥有比较丰富的编译器开发经验，对C++11的新特性也有较好的理解。此外，IBM北美编译器团队成员Michael（他是C++标准委员会的成员）也参加了本书的编写工作。在书籍的编写上，Michael为本书拟定了提纲、确定了章节主题，并直接编写了本书的首章。其余作者则分别对C++11各种新特性进行了详细研究讨论，并完成了书稿其余各章的撰写工作。在书稿完成后，除了请Michael为本书的部分章节进行了审阅并提出修改意见外，我们又邀请了IBM中国信息开发部及IBM北京编译器团队的一些成员对本书进行了详细的审阅。虽然在书籍的策划、编写、审阅上我们群策群力，尽了最大的努力，以保证书稿质量，不过由于C++11标准发布时间不长，理解上的偏差在所难免，因此本书也可能在特性描述中存在一些不尽如人意或者错误的地方，希望读者、同行等一一为我们指出纠正。我们也会通过博客、微博发布与本书相关的所有信息，并与本书读者共同讨论、进步。如何阅读本书读者在书籍阅读中可能会发现，本书的一些章节对C++基础知识要求较高，而某些特性很可能很难应用于自己的编程实践。这样的情况应该并不少见，但这并不是这门语言缺乏亲和力，或是读者缺失了背景知识，这诚然是由于C++的高成熟度导致的。在C++11中，不少新特性都会局限于一些应用场景，比如说库的编写，而编写库却通常不是每个程序员必须的任务。为了避免这样的状况，本书第1章对C++11的语言新特性进行了分类，因此读者可以选择按需阅读，对不想了解的部分予以略过。一些本书的使用约定，读者也可以在第1章中找到。致谢在这里我们要对IBM中国信息开发部的陈晶（作者之一）、卢昉、付琳，以及IBM北京编译器团队的冯威、许小羽、王颖对本书书稿详尽细致的审阅表示感谢，同时也对他们专业的工作素养表示由衷的钦佩。此外，我们也要感谢IBM XL编译器中国开发团队的舒蓓、张嗣元两位经理在本书编写过程中给予的大力支持。而IBM图书社区的刘慎峰及华章图书的杨福川编辑的辛勤工作则保证了本书的顺利出版，在这里我们也要对他们以及负责初审工作的孙海亮编辑说声谢谢。此外，我们还要感谢各位作者的家人在书籍编写过程中给予作者的体谅与支持。最后要感谢的是本书的读者，感谢你们对本书的支持，希望通过这本书，我们能够一起进入C++编程的新时代。IBM XL编译器中国开发团队

## 内容概要

《深入理解C++11:C++11新特性解析与应用》内容简介：国内首本全面深入解读C++11新标准的专著，由C++标准委员会代表和IBM XL编译器中国开发团队共同撰写。不仅详细阐述了C++11标准的设计原则，而且系统地讲解了C++11新标准中的所有新语言特性、新标准库特性、对原有特性的改进，以及如何应用所有这些新特性。

《深入理解C++11:C++11新特性解析与应用》一共8章：第1章从设计思维和应用范畴两个维度对C++11新标准中的所有特性进行了分类，呈现了C++11新特性的原貌；第2章讲解了在保证与C语言和旧版C++标准充分兼容的原则下增加的一些新特性；第3章讲解了具有广泛可用性、能与其他已有的或者新增的特性结合起来使用的、具有普适性的一些新特性；第4章讲解了C++11新标准对原有一些语言特性的改进，这些特性不仅能让C++变得更强大，还能提升程序员编写代码的效率；第5章讲解了C++11在安全方面所做的改进，主要涵盖枚举类型安全和指针安全两个方面的内容；第6章讲解了为了进一步提升和挖掘C++程序性能和让C++能更好地适应各种新硬件的发展而设计的新特性，如多核、多线程、并行编程方面的新特性；第7章讲解了一些颠覆C++一贯设计思想的新特性，如lambda表达式等；第8章讲解了C++11为了解决C++编程中各种典型实际问题而做出的有效改进，如对Unicode的深入支持等。附录中则介绍了C++11标准与其他相关标准的兼容性和区别、C++11中弃用的特性、编译器对C++11的支持情况，以及学习C++11的相关资源。

## 作者简介

作者：（加拿大）Michael Wong IBM XL编译器中国开发团队Michael Wong，C++11标准委员会（WG21）委员，WG21加拿大代表团团长及IBM公司代表（投票人），WG21 SG5(transactional Memory study group)学习小组组长。除此之外，Michael还是OpenMP委员会的CEO，SCC(Standards Council of Canada)的副主席。Michael是多个C++11/OpenMP/STM新特性的合作编写者，活跃于各种技术会议，而且为Programming Committee of Boost以及IWOMP等国际会议服务。Michael也是IBM公司的高级技术专家，领导IBM XL C++编译器、XL C编译器的开发工作，当前致力于IBM编译器的C++11新特性部署。Michael在并行编程、C++基准测试、对象模型、泛型编程、模板元编程等多个技术领域均有涉猎，并且在对象模型以及事务内存等方面取得过多个专利。早年Michael于多伦多大学获取了天体物理学的学士学位，并在滑铁卢大学获得了硕士学位。在加入IBM之前，Michael曾先后供职于大卫·邓禄普天文台（David Dunlap Observatory），CDC公司（Control Data Corporation），BMO Nesbitt Burns公司等。IBM XL编译器中国开发团队，IBM拥有悠久的编译器开发历史（始于上世纪80年代），在全球有将近400名高素质工程师组成的研发团队，其中包括许多世界知名的研究学者和技术专家。IBM一直以来都是编程语言的制定者和倡导者之一，并将长期在编译领域进行研发和投资。IBM编译器中国开发团队于2010年在上海成立，现拥有编译器前端开发人员（C/C++）、后端开发人员、测试人员，以及性能分析人员共16人。团队与IBM北美编译器团队紧密合作，共同开发、测试和发布基于POWER系统的AIX及Linux平台下的XL C/C++和XL Fortran系列产品，并对其提供技术支持。虽然团队成立时间不长，但已于2012年成功发布最新版本的XL C/C++ for Linux V12.1 & XL Fortran for Linux V14.1，并获得7项发明专利。团队成员拥有较丰富的编译器开发经验，对编译技术、编程语言、性能优化和并行计算等各领域都有一定的研究，也对C++11标准的各种新特性有较早的研究和理解，并正在实际地参与C++11新特性的开发工作。

## 书籍目录

免责声明

序

前言

第1章 新标准的诞生

1.1 曙光：c++11标准的诞生

1.1.1 c++11/c++0x（以及c11/c1x）—新标准诞生

1.1.2 什么是c++11/c++0x

1.1.3 新c++语言的设计目标

1.2 今时今日的c++

1.2.1 c++的江湖地位

1.2.2 c++11语言变化的领域

1.3 c++11特性的分类

1.4 c++特性一览

1.4.1 稳定性与兼容性之间的抉择

1.4.2 更倾向于使用库而不是扩展语言来实现特性

1.4.3 更倾向于通用的而不是特殊的手段来实现特性

1.4.4 专家新手一概支持

1.4.5 增强类型的安全性

1.4.6 与硬件紧密合作

1.4.7 开发能够改变人们思维方式的特性

1.4.8 融入编程现实

1.5 本书的约定

1.5.1 关于一些术语的翻译

1.5.2 关于代码中的注释

1.5.3 关于本书中的代码示例与实验平台

第2章 保证稳定性和兼容性

2.1 保持与c99兼容

2.1.1 预定义宏

2.1.2 `__func__`预定义标识符

2.1.3 `_pragma`操作符

2.1.4 变长参数的宏定义以及`__va_args__`

2.1.5 宽窄字符串的连接

2.2 `long long`整型

2.3 扩展的整型

2.4 宏`_cplusplus`

2.5 静态断言

2.5.1 断言：运行时与预处理时

2.5.2 静态断言与`static_assert`

2.6 `noexcept`修饰符与`noexcept`操作符

2.7 快速初始化成员变量

2.8 非静态成员的`sizeof`

2.9 扩展的`friend`语法

2.10 `final/override`控制

2.11 模板函数的默认模板参数

2.12 外部模板

2.12.1 为什么需要外部模板

2.12.2 显式的实例化与外部模板的声明

- 2.13 局部和匿名类型作模板实参
- 2.14 本章小结
- 第3章 通用为本，专用为末
  - 3.1 继承构造函数
  - 3.2 委派构造函数
  - 3.3 右值引用：移动语义和完美转发
    - 3.3.1 指针成员与拷贝构造
    - 3.3.2 移动语义
    - 3.3.3 左值、右值与右值引用
    - 3.3.4 `std::move`：强制转化为右值
    - 3.3.5 移动语义的一些其他问题
    - 3.3.6 完美转发
  - 3.4 显式转换操作符
  - 3.5 列表初始化
    - 3.5.1 初始化列表
    - 3.5.2 防止类型收窄
  - 3.6 pod类型
  - 3.7 非受限联合体
  - 3.8 用户自定义字面量
  - 3.9 内联名字空间
  - 3.10 模板的别名
  - 3.11 一般化的sfinae规则
  - 3.12 本章小结
- 第4章 新手易学，老兵易用
  - 4.1 右尖括号>的改进
  - 4.2 auto类型推导
    - 4.2.1 静态类型、动态类型与类型推导
    - 4.2.2 auto的优势
    - 4.2.3 auto的使用细则
  - 4.3 decltype
    - 4.3.1 typeid与decltype
    - 4.3.2 decltype的应用
    - 4.3.3 decltype推导四规则
    - 4.3.4 cv限制符的继承与冗余的符号
  - 4.4 追踪返回类型
    - 4.4.1 追踪返回类型的引入
    - 4.4.2 使用追踪返回类型的函数
  - 4.5 基于范围的for循环
  - 4.6 本章小结
- 第5章 提高类型安全
  - 5.1 强类型枚举
    - 5.1.1 枚举：分门别类与数值的名字
    - 5.1.2 有缺陷的枚举类型
    - 5.1.3 强类型枚举以及c++11对原有枚举类型的扩展
  - 5.2 堆内存管理：智能指针与垃圾回收
    - 5.2.1 显式内存管理
    - 5.2.2 c++11的智能指针
    - 5.2.3 垃圾回收的分类
    - 5.2.4 c++与垃圾回收

- 5.2.5 c++11与最小垃圾回收支持
- 5.2.6 垃圾回收的兼容性
- 5.3 本章小结
- 第6章 提高性能及操作硬件的能力
  - 6.1 常量表达式
    - 6.1.1 运行时常量性与编译时常量性
    - 6.1.2 常量表达式函数
    - 6.1.3 常量表达式值
    - 6.1.4 常量表达式的其他应用
  - 6.2 变长模板
    - 6.2.1 变长函数和变长的模板参数
    - 6.2.2 变长模板：模板参数包和函数参数包
    - 6.2.3 变长模板：进阶
  - 6.3 原子类型与原子操作
    - 6.3.1 并行编程、多线程与c++11
    - 6.3.2 原子操作与c++11原子类型
    - 6.3.3 内存模型，顺序一致性与memory\_order
  - 6.4 线程局部存储
  - 6.5 快速退出：quick\_exit与at\_quick\_exit
  - 6.6 本章小结
- 第7章 为改变思考方式而改变
  - 7.1 指针空值—nullptr
    - 7.1.1 指针空值：从0到null，再到nullptr
    - 7.1.2 nullptr和nullptr\_t
    - 7.1.3 一些关于nullptr规则的讨论
  - 7.2 默认函数的控制
    - 7.2.1 类与默认函数
    - 7.2.2 “= default”与“= deleted”
  - 7.3 lambda函数
    - 7.3.1 lambda的一些历史
    - 7.3.2 c++11中的lambda函数
    - 7.3.3 lambda与仿函数
    - 7.3.4 lambda的基础使用
    - 7.3.5 关于lambda的一些问题及有趣的实验
    - 7.3.6 lambda与stl
    - 7.3.7 更多的一些关于lambda的讨论
  - 7.4 本章小结
- 第8章 融入实际应用
  - 8.1 对齐支持
    - 8.1.1 数据对齐
    - 8.1.2 c++11的alignof和alignas
  - 8.2 通用属性
    - 8.2.1 语言扩展到通用属性
    - 8.2.2 c++11的通用属性
    - 8.2.3 预定义的通用属性
  - 8.3 unicode支持
    - 8.3.1 字符集、编码和unicode
    - 8.3.2 c++11中的unicode支持
    - 8.3.3 关于unicode的库支持

8.4 原生字符串字面量

8.5 本章小结

附录a c++11对其他标准的不兼容项目

附录b 弃用的特性

附录c 编译器支持

附录d 相关资源

## 编辑推荐

《深入理解C++11:C++11新特性解析与应用》编辑推荐：C++标准委员会成员和IBM XL编译器中国开发团队共同撰写，权威性毋庸置疑。系统、深入、详尽地讲解了C++11新标准中的新语言特性、新标准库特性、对原有特性的改进，以及所有这些新特性的应用。

## 精彩短评

- 1、不错这本书，适合想了解c++11的同学
  - 2、IBM中国的程序员中文不好，或者我的中文不好
  - 3、难得的一本有参与C++11标准制定和实现的中国人用中文写的书
  - 4、C++11还是很厉害，一下子就变成高级语言了..
- 这本书基本上是把新特性罗列了一遍，确实很像是做编译器的人写的，罗列的非常细致，很多一看就非常冷门的东西也都讲到了。缺点可能就是过于细致了，不能突出C++11常用的重点。看完还是很有启发，尤其是POD、lambda、atomic/thread、智能指针、auto/decltype、右值引用/完美转发等等。书中也有些错误，尤其是POD中对标准的翻译都是错误的，还得看标准的英文原文。对于想了解C++11的人，可以快速过一遍，就可以快速上手了。
- 5、2014-10-06 想读。2016-11-28 入手 kindle 电子版并开始阅读。2016-12-05 读完。通俗易懂地介绍了C++11 新增语言特性。我也是粗略看一遍，有个印象，具体使用时再查相关资料。本书并不涉及新增标准库内容。
  - 6、干货挺多，适合对C++较为熟悉的又想了解C++11的同学，看完即可，可作为参考书。
  - 7、真是讲解c++11的好书，又全面又细致
  - 8、总体来说还不错，适合快速概览C++11新特性。
  - 9、不错的书
  - 10、几个重要的特性都有涉及，讲解的也清晰。还算不错。
  - 11、内容不全，也并没有太多的深入。标题党
  - 12、买书的时候这本是第一本专门介绍C++11的书籍，通读一遍感觉还不错
  - 13、重点不突出 一些例子举的实在无力 实在对不起这个书名 换个名字 比如浅析c++11才恰当
  - 14、名过其实
  - 15、写的还挺好的
  - 16、对特性的讲解很深入，只可惜本书没有涉及太多新的STL库的内容。
  - 17、感觉还是不错的，是一本方便了解C++11新特性的书
  - 18、不太懂作者分类的思路，为什么智能指针要放在类型安全一章，多线程跟可变参数模板在一章，而lambda跟 nullptr 又在同一章...
  - 19、文档性质的书籍 另外语言不是很通顺
  - 20、算不上深入~~
  - 21、对新特性有个大致的了解。
  - 22、了解C++11 很不错的书籍
  - 23、翻译的很一般。内容也不怎么样。
  - 24、例子很多都是标准提案里面的例子，很多很好，但是对名词解释或者说使用的阐述，总感觉那么拗口，像是翻译加人工的结果。。书不错，不适合新手，不适合对c++11一点都没有了解的人。
  - 25、挺好 就是没有库内容。
  - 26、对C++11的特性描述比较详尽，可以当成参考书。与《C++标准库》C++11版配合，可以充分理解C++11。
  - 27、前100页很不错...
  - 28、不错，读中文书还是快，基本了解了c++11的新特性
  - 29、所有::符号都印成了:，代码里全角半角不分，也不对齐。书还行，但是没讲全
  - 30、给我感受最深的就是细致。用了很多代码描述特定的需求是从什么样的问题中产生的（比如在移动语义之前，连拷贝构造都要先介绍一遍），这样的需求在C++11之前需要怎样的“奇技淫巧”来解决，而在C++11中提供了怎样的标准语法来简单、漂亮的解决，即使资历较浅如我者，也能通过这样的方式理解C++11，也更深地理解了C++98以及编程本身。不愧是编译器的团队，对涉及编译器特性的地方，比如RVO、内存模型都有讲解（内存模型讲得很细）。虽然是团队作品，但保持了文字的规范（编辑错误还是略有）、叙述的耐心。非常用心，力荐！我略感遗憾的是对新标准库特性介绍得好

像就很不完整了，这本书题材非常好，希望有C++14或C++17的升级版，把新标准库特性也系统介绍一下。

31、找不到合适的书，算是不错了，支持一下！

32、从编译器制造者的角度写的，而不是从一个干活的一线码农的角度写。一开始就给了一个大表，列了一下新定义的宏。这种罗列谁有兴致看啊。总的来说，罗列的有些多，似乎是对着change list写的一本书，而不是侧重于C++11的新特性如何帮助码农们提高生产效率。

33、c++11的部分特性 适合老司机

34、有很多不容易理解，学习新的C++11标准

35、C++11太强大了，primer已经不够用了。解释还是不错的，但当我翻阅C++标准时候才发现很多都是直接摘抄标准文档。

36、讲解非常详细，需要一定的基础才能看懂

37、我觉得很好看啊，薄薄的但全是干货。

38、错误还是很多的，编校质量堪忧。目前看到第三章转移语义，关于引用折叠，书中给的表格实在是费解，网上随便一查就懂了。纸质版要50块，我认为连一半都不值。。

39、懒得读标准的可以读这本书了解c++11的新特性，配不上书名说的“深入”；错误太多，虽然都不是大错，说utf-16是定长编码的真是太没常识了。

40、C++的一次重大进化，翻译的還是有點說不明白的警扭，但是本好書。

41、讲C++11新出来的特性，对于已经熟悉C++的人来说，很合适，能够快速了解C++11有哪些新特性，讲的也非常好，简洁易懂，适度深入。

42、很好，c++11确实改进很多，该书对新特性阐述得很到位！

43、如果你想快速了解c++11有哪些特性，该书可以帮助你。c++11标准库添加的东西比较多，该书甚少涉及，需要通过额外的资料来了解

44、一般般，对c++11完全没有概念的可以翻翻

45、作者将C++11新特性解析得清晰易懂。

46、C++ Primer 第5版是从零讲C++，很厚很浪费时间，而这本是讲C++11新出来的特性，对于已经熟悉C++的人来说，很合适，能够快速了解C++11有哪些新特性，讲的也非常好，简洁而易懂

47、哎,感觉c++现在的发展让人有点望闻止步了

48、还是期待Rust 1.0了

49、纯粹从描述C++11的新特性来说，比C++ Primer（第5版）更集中，适合对C++11有一定了解的人

。

50、便于快速了解c++11新特性，但是只讲语言部分，没有标准库的更新感觉有点耍流氓。。。

51、感觉没有按标准来定大纲, 11里还有大量内容都是没讲的。

52、一般般，基本是标准的一些罗列和演示，直接看cppreference其实就差不多了，就当存个档。说深入理解也严重了。有些特性没实现，也不常用。C++11也算是C++最具革命性的一个改进了。

53、写的确实非常好，写编译器的人就是厉害啊。

54、比较全面，而且有些东西先的C++编译器好像都不支持。

55、C++11的新特性目前对于我个人来说没什么用; 本书的优点是每个新特性注明了适用的用户类别, 读起来比较方便; 编译命令写得很清楚。

56、读的第二本IBM出的书，感觉一般，仅仅适合入门或者学习C++的人读。

57、c++11很有趣

58、好多新特性

## 精彩书评

- 1、书籍其实不错，对C++11的不少特性都有详细的描述（照书中说法应该还有下册，期待一下~），作者都是一线的编译器开发者，所以内容上还是有不少干货的，当然，说其鞭辟入里还不至于，但是用以扩充视野、查漏补缺绝对没有问题，推荐~~~更难得的是书籍作者基本都是国人，更应该好好支持一下~PS：如今C++11标准发布已近两年，但编译器的（完全）支持仍然有待时日，话说C++14马上就要到了，感觉跟不上节奏啊~又PS：C++11的不少特性还是很激动人心的，试用一番之后也挺觉顺手，不过欢欣之余又不免惆怅：这C++貌似又复杂了啊.....
- 2、对C++11期待很久了。本书选取大部分C++11中新增和修订（相对于C++98来说）的特性作为主题，并按照一定的相关性将这些特性划分成章节，然后逐个地对这些特性进行讲解，这对于有C++经验的人来讲，是一种非常好的组织方式和成书方法，这样，你只需要将本书中的内容吃透，基本上可以说一窥C++11的全貌，与其他覆盖全面的C++书籍相比，这本书更具针对性。但是，本书并没有覆盖一些我认为非常重要的C++11主题，比如thread，bind和function等，使本书稍微有点失色。不过，要做到能够讲这些新的特性，投入到实际的应用中，不仅需要对这些特性的理解，还需要将这些特性应用中实际项目中，在项目中观察这些特性的优势。
- 3、正如老头说的11算是新的一门语言.大家都要了解.从vs2010起就断断续续了解0x,了解着新的特性,有高兴,也有感觉更多的坑.所以,作为一个不大可能写出个排山倒海库的半吊子c++程序猿,心情决定着我对重点了解大的,对性能,工作有利的完美特性.新容器,右值引用,lambda,thread,并行.对这本书的评价是不错的c++11中文原创书,结构不错,每个特性都会标识出最针对的用户群。内容很有营养,在特性的来由,特性的介绍上很丰满.但使用上不够深入.比如lamdba的最有价值的例子,可以做到像go语言里的defer关键字一样,做到像我们平时在一个函数最开始一行,lock l(&m\_lockobject);一样,来在函数退出时自动帮我们处理各种资源的释放.在并行计算里,没有一个合适的例子来表明在实际工作中的锁的进步。也没有篇幅介绍新的unordered和新容器.完全忽视了我这样高层..的半吊子啊。
- 4、现在真是什么书都敢把书名叫做“深入理解XXX”。觉得这本书叫做《C++11初探》之类的更合适，因为基本是在罗列C++11的局部特性，然后放上一些提案中或者Wikipedia中代码的例子，就凑成了一本书。并没有很多深入的探讨。不过作为第一次专门看C++11相关的书对我来说，还是学到了不少东西，起码知道了可以去哪里找权威的资料：要想知道某项技术如何被采纳到标准中，直接去<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/> 中找。还有一个感觉就是C++这个东西真的好恐怖。C++11中看着那么显然的右值引用和类型推导，在之前的版本中竟然几乎都是不存在的！而新加入的这些概念和功能，有些是借鉴自其它语言（类型推导，lambda表达式），有些我在其它语言中都没见过，例如右值引用，完美转发。感觉是跳到了C给它挖的大坑中，但是却能自己搞一套看起来很优雅很自然但实际上又很复杂的解决方案。我很怀疑它解决的很多问题可能在有些语言里都不存在。不过既然看起来这么炫酷又严谨，而且那么多人都说C++如何如何强大，我还是看看好了。。不过还是没理解它哪里强大。还有很多新特性其实是来解决C++98的bug的，用了那么久的C++原来背后一直都有那么多问题，想象都觉得卧槽。我不懂编译原理，当初用STL的时候，两个尖括号不能连在一起的限制真是让我觉得很奇怪，但以为是必须分开编译器才会识别，原来不是啊。任何书只要跟C++有关，都会刻显得复杂。想想POD的定义、decltype的特殊情况、编译器默认生成的函数们。。从来没有哪个是一条规则可以说清楚的，而且为了讲明白一个问题，往往要引入更多的概念。这些概念看起来还挺熟悉，但却绝不是你理解的那个意思。。。我表示服！
- 5、C++11特性的基本介绍，对于std::move的介绍没我想象中的深度，不过Lambda函数还是学到以前不知道的知识，要看完全版的话看C++ Standard Library 11就够了。其他的只是简单的举例，离深入理解还有一段距离。
- 6、刚看了3分钟，就在书本正文第二页，发现了一个错误：ISO/IEC 14882，被写成了：IOS/IEC 15882情何以堪呀，这作者是难道是做苹果 IOS 开发的，帮导师干活？幸亏是先下载 pdf 来看，也没亏本
- 7、6.3.3 explains memory orders which are newly supported by c++11. The explanation may be the most readable text about memory order in Chinese. Crystally clear! In my idea, c++11 memory model is as same as Java. If you are confused about the concept of memory order in multicore computer. This part must be useful.

## 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](http://www.tushu000.com)