高等教育质量工程信息技术系列示范教材

新概念

C++程序设计大学教程

（第3版）

张基温 编著

清华大学出版社

高等教育质量工程信息技术系列示范教材

新概念
C++程序设计大学教程
（第3版）

张基温 编著

清华大学出版社
北京

内 容 简 介

本书是一本面向大学计算机专业的C++程序设计教材，以面向对象程序设计为主线，突出C++的基本特点，介绍C++1y的重要新特性。全书共分为4篇11个单元。

第1篇：C++面向对象起步。用4个单元帮助初学者建立面向对象的问题分析思维，掌握相关方法和语法知识，树立面向对象程序中“一切皆对象，一切来自类”的意识，初步领略面向对象程序设计的奥妙。

第2篇：C++面向抽象程序设计。用两个单元介绍C++的继承机制，并帮助读者理解如何在一个程序中组织类，以及什么样的类结构才是好的程序结构。

第3篇：C++泛型程序设计。用两个单元介绍多态性和STL。C++的泛型的通用、灵活的特点将给读者的学习带来一定乐趣，也为读者将来从事程序开发工作提供了更多便捷方法。

第4篇：C++深入编程。用3个单元介绍C++实体访问、函数和I/O流等方面的细节和内容，进一步丰富程序设计语言机制，使读者在程序开发上能够锦上添花。

本书理念先进、概念清晰、讲解透彻、便于理解。书中例题经典、习题丰富、覆盖面广，适合作为高等学校各专业的面向对象程序设计教材。本书还可供培训机构使用，也可供相关领域人员自学。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目（CIP）数据

新概念C++ 程序设计大学教程 / 张基温编著. —3版. —北京：清华大学出版社，2018

（高等教育质量工程信息技术系列示范教材）

ISBN 978-7-302-48154-6

Ⅰ. ①新… Ⅱ. ①张… Ⅲ. ①C语言-程序设计-高等学校-教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字（2017）第207474号

责任编辑： 白立军　王冰飞
封面设计： 常雪影
责任校对： 梁　毅
责任印制： 李红英

出版发行： 清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 装 者： 清华大学印刷厂
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 26.5　　**字　　数：** 629千字
版　　次： 2013年3月第1版　　2018年1月第3版　　**印　　次：** 2018年1月第1次印刷
印　　数： 1～1500
定　　价： 49.80元

产品编号：075145-01

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

（一）

1979 年，Bjarne Stroustrup（C++之父）正在准备他的博士毕业论文，他有机会使用一种称为 Simula 的语言。顾名思义，Simula 语言主要用于仿真。其 Simula 67 版被公认是首款支持面向对象的语言。Stroustrup 发现面向对象的思想对于软件开发非常有用，但是因 Simula 语言执行效率低，其实用性不强。于是他决定自行开发一种面向对象的语言，这就是今日的 C++。

笔者一直关心 TIOBE 社区的程序设计语言排行榜，因为它能为开发和教学人员提供一份程序设计语言的行情变化资料。图 1 为 2002 年到 2017 年 2 月之间主要 TIOBE 程序设计语言排行的变化情况。在这个排行榜上发生了戏剧性变化的程序设计语言就是 C++。其第一次戏剧性的变化发生在 2004 年，在这一年中它的市场份额急剧下滑。但在之后的 10 年间基本稳定，一直保持在第三位。本书的第 1 版就是在这样的情况下编写的。其第二次戏剧性变化是在本书的第 1 版出版之后，它先在 2014 年间急剧下跌，又在 2015 年奇迹般地回归。

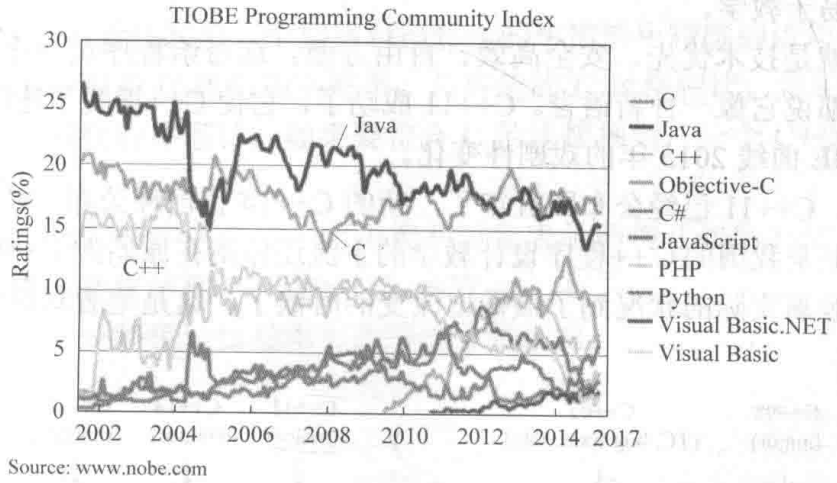

图 1　2002 年到 2017 年 2 月 TIOBE 程序设计语言排行变化情况

C++的这些变化似乎有些莫名其妙，但认真地分析一下，这些变化还是可以解释的：其一是其他新兴语言（主要是 C#和 Object-C）对于市场份额的分割；其二则是其自身标准变化的影响。下面主要分析第二方面的因素。

C++是 Bjarne Stroustrup 于 1979 年准备一个项目时着手开发的一种程序设计语言，1985 年被市场化。C++标准委员会于 1998 年 11 月推出了其第一个 ISO 标准（俗称 C++98），2003 年推出其 ISO 标准第 2 版（俗称 C++03），C++11 则是从 2005 年就开始提交，到 2011 年 8

月才发布的 C++标准（俗称 C++11，提交时称为 C++0x）。从图 1 可以看出，每个标准出台到影响其市场份额有一个窗口期，这是该标准投石问路的过程。

C++03 是 C++98 的修正版，其初衷是修正 C++98 的一些不足。但是由于 C++脱胎于 C，遵循着 C 是 C++子集的原则，同时 Bjarne Stroustrup 坚持要保持其“适合教学”及既支持面向过程又支持面向对象的多泛型特色，成就了其概念清晰、设计严密、功能强大、效率较高的优点，但也带来过于复杂（如指针）、标准库功能不足，被人称为有精英化倾向的语言。因此它比较受教育界欢迎，而程序员觉得难用。不过，在通过 C++03 标准之前，人们还没有认识到这些问题，反而降低了效率，加剧了其缺陷的影响，使其在 2004 年遭受到第一次强力冲击。

2004—2005 年间的滑铁卢之惨使 C++的设计者和标准制定者开始清醒起来，将指导思想修订如下。

（1）维持与 C++98，可能的话还有与 C 语言之间的兼容性与稳定性。

（2）尽可能通过标准程序库来引进新的特性，而不是扩展核心语言。

（3）能够促进编程技术的变更优先。

（4）改进 C++ 以帮助系统和程序库的设计，而不是引进只对特定应用有用的新特性。

（5）增强类型安全，给现行不安全的技术提供更安全的替代方案。

（6）增强直接与硬件协同工作的性能和能力。

（7）为现实世界中的问题提供适当的解决方案。

（8）实行零负担原则(如果某些功能要求额外支持，那么只有在该功能被用到时这些额外的支持才被用到) 。

（9）使 C++易于教学。

简单地说，就是技术优先、安全高效、自由方便。这带给程序员一个全新的面貌，以至于连 C++之父都说它像一种新语言。C++11 成功了，它使 C++摆脱了连续 10 年的步步下降，造就了 TIOBE 曲线 2015 年的戏剧性变化。

如图 2 所示，C++11 已经公布四五年了，新的 C++14 也已经公布，C++17 也在紧锣密鼓地部署之中。但是我国的C ++程序设计教学的主流还停留在原始的 C++98 甚至更旧的版本上。如此严重脱离实际的状况到了该做出改变的时候了。这是笔者对这本 C++教材进行改编的动因之一。

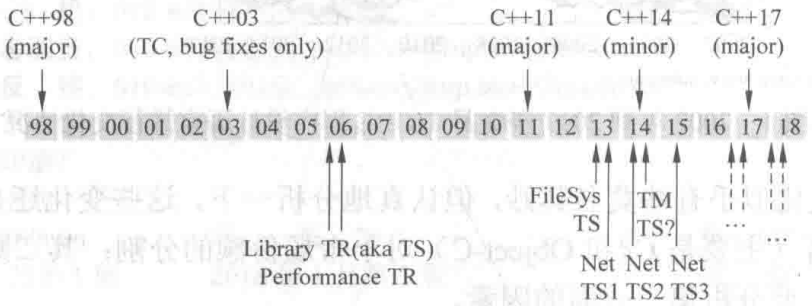

图 2　近期 C++标准修订步伐

不过，世界万物都有惯性。要一下子全部改到 C++11 上，很多人还是难以接受的，并且笔者自己也还在学习消化之中。在本书中，仅仅给出了 C++11 的部分新特点的接口，先

让大家了解一下这些性能。在适当的时候，再做较全面和深入的介绍。

（二）

撇开 C++11 和 C++14 不谈，仅从一般性来讲，目前的 C++教学也是不尽如人意的，把 C++当作 C 的教学模式还广泛存在。在本书的第 1 版中虽然做了不少努力，但还不够。

Bjarne Stroustrup 曾经感慨地说："我不是使用支持工具进行巧妙设计的信徒，但是我强烈支持系统地使用数据抽象、面向对象编程和类属编程。不拥有支持库和模板，不进行事先的总体设计，而是埋头写下一页页的代码，这是在浪费时间，这是在给维护增加困难。"他还认为，"一个人对 C 了解得越深，在写 C++程序时就越难避免 C 的风格，并会因此丢掉 C++的某些潜在优势。"为此，他提出了以下几个相关的要点，在这些情况下做同样的事情时，在 C++中存在比 C 中更好的处理方式。

（1）在 C++中几乎不需要用宏。用 const 或 enum 定义显式常量，用 inline 避免函数调用的额外开销，用 template 去刻画一簇函数或者类型，用 namespace 去避免名字冲突。

（2）不用在需要变量之前去声明它，以保证立即对其进行初始化。声明可以出现在能出现语句的所有位置上，可以出现在 for 语句的初始化部分，也可以出现在条件中。

（3）不要用 malloc()，new 运算符能将同样的事情做得更好。对于 realloc()，试一试 vector()。

（4）试着去避免 void*、指针算术、联合和强制，除了在某些函数或类实现的深层之外。在大部分情况下，强制都是设计错误的指示器。如果必须使用某个显式的类型转换，设法用一个"新的强制"，设法写出一个描述自己想做的事情的更精确的语句。

（5）尽量少用数组和 C 风格的字符串。与传统的 C 风格相比，使用 C++标准库 string 和 vector 常常可以简化程序设计。如果要符合 C 的连接规则，一个 C++函数就必须被声明为具有 C 连接的。

最重要的是，要将程序考虑为一组由类和对象表示的相互作用的概念，而不是一堆数据结构和一些去拨弄数据结构中二进制位的函数。

探索如何彰显 C++特色，也是本书改编的重要动因。

（三）

本次修改，将全书划分为 4 篇。

第 1 篇：C++面向对象起步。用 4 个单元帮助初学者建立面向对象的分析问题的思维，掌握相关方法和相关知识，树立面向对象程序设计中"一切皆对象，一切来自类"的意识。

第 2 篇：C++面向抽象程序设计。用两个单元帮助读者理解如何在一个程序中组织类，以及什么样的类结构才是好的程序结构。

第 3 篇：C++泛型程序设计。用两个单元介绍多态性和 STL。C++的泛型、通用、灵活的特点给读者的学习带来了一定乐趣，也为读者将来从事程序开发工作提供了更多便捷

方法。

第 4 篇：C++深入编程。用 3 个单元介绍 C++在名字和实体、常量、函数、I/O 流等几个方面的细节，让读者在程序开发上能够做到锦上添花。进一步提升读者“程序设计=计算思维+语言艺术”的观念。

此外，本书每个单元都围绕一个主题展开，部分单元还增添了“知识链接”部分，其目的是引申基本内容，或为以后的学习作一些铺垫。

本书的结构体系安排是考虑了以下几个因素和写作思想的结果。

（1）B.S 的建议。

（2）重要先学，特色优先。

（3）思维开路，语法补充。

（4）多层次教学需要。

需要说明一点：本书给出的许多示例，虽然有用，但主要用于说明一种语法概念或给出一种编程思路，还不是精益求精的实用程序。

（四）

由 J.Piaget、O.Kernberg、R.J.Sternberg、D.Katz、Vogotsgy 等人创建的建构主义(constructivism)学习理论认为，知识不是通过教师传授得到的，而是学习者在一定的情境即社会文化背景下借助其他人（包括教师和学习伙伴）的帮助，利用必要的学习资料，通过意义建构的方式而获得的。在信息时代，人们获得知识的途径发生了根本性的变化，教师不再是单一的“传道、授业、解惑”者，帮助学习者构建一个良好的学习环境也成为其一个重要职责。当然，这也是现代教材的责任。本书充分考虑了这些问题。

为了给读者创造一个良好的学习环境，本书的部分单元设置了“知识链接”栏目。这个栏目的设置可以帮助学习者对 C++的机制作更深入的了解，也为精力充沛、不够消化者提供一点“加餐”。

除了正文外，在每个单元后面都安排了“概念辨析”“代码分析”“开发实践”和“探索验证”4 种自测和训练实践环节，从而建立起一个全面的学习环境。

（1）概念辨析主要提供选择和判断两类自测题目，帮助学习者理解本单元学习过的有关概念，把当前学习内容所反映的事物尽量和自己已经知道的事物相联系，并认真思考这种联系，通过“自我协商”与“相互协商”，形成新知识的同化与顺应。

（2）代码分析。代码阅读是程序设计者所应掌握的基本能力之一。代码分析部分的主要题型是通过阅读程序找出错误或给出程序执行结果。

（3）开发实践。提高程序开发能力是本书的主要目标。本书在绝大多数单元后面都给出了相应的作业题目。但是，完成这些题目并非只是简单地写出其代码，而要将其看作一个“思维+语法+方法”的工程训练。因此，要求每道题的作业都要以文档的形式完成。文档中应包括以下内容。

① 问题分析与建模。

② 源代码设计。

③ 测试用例设计。

④ 程序运行结果分析。

⑤ 编程心得（包括运行中出现的问题与解决方法、对于测试用例的分析、对于运行结果的分析等）。

⑥ 文档的排版也要遵循统一的格式。

（4）探索验证。建构主义提倡，学习者要用探索法和发现法去建构知识的意义。学习者要在意义建构的过程中主动地收集和分析有关的信息资料，对碰到的问题提出各种假设，并努力加以验证。

（五）

笔者从事程序设计教学近 30 多年。这 30 多年是在不断探索中走过来的。从 20 世纪 80 年代末，笔者就开始探索程序设计课程从语法体系到问题驱动的改革；到了 20 世纪 90 年代中期又在此基础上考虑让学生在学习程序设计的同时掌握程序测试技能；2003 年开始考虑如何改变学习了 C++而设计出的程序却是面向过程的状况。每个阶段的探索都反映在自己不同时期的相关作品中。本书则是自我认识又一次深化的表达。

在这 30 多年的探索中，笔者越来越感觉到编写教材的责任很大、困难很多。要编写一本好的教材，不仅需要对本课程涉及内容有深刻的了解，还要熟悉相关领域的知识，特别是要不断探讨贯穿其中的教学理念和教育思想。所以，越到后来，就越感到自己知识和能力的不足。可是，作为一项历史性任务的研究，笔者又不愿意将之半途而废，只能硬着头皮写下去。每一次任务的完成，都得益于一些热心者的支持和帮助。在本书的写作过程中，赵忠孝教授、姚威博士、张展为博士以及张秋菊、史林娟、张有明、戴璐、张展赫、董兆军、吴灼伟（插图）等参加了有关部分的写作。在此谨表谢意。同时，一如既往地希望得到读者广泛的批评和建议，以便将这本书改得更好。

本书就要出版了。它的出版，是笔者在这项教学改革工作中跨上的一个新的台阶。笔者衷心希望得到有关专家和读者的批评和建议，也希望能多结交一些志同道合者，把这项教学改革推向更新的境界。

张基温

丁酉秋于穗小海之畔

目　录

第 1 篇　C++面向对象起步

此为试读，需要完整PDF请访问：www.ertongbook.com

第 2 篇 C++面向抽象程序设计

第 3 篇　C++泛型程序设计

第 4 篇 C++深入编程

此为试读，需要完整PDF请访问：www.ertongbook.com