异步图书
www.epubit.com.cn

SAMS

畅销全球的
轻量级C++经典教程

中文版
累计销量
70000册

Sams Teach Yourself C++
in One Hour a Day
Eighth Edition

涵盖C++14和C++17新标准

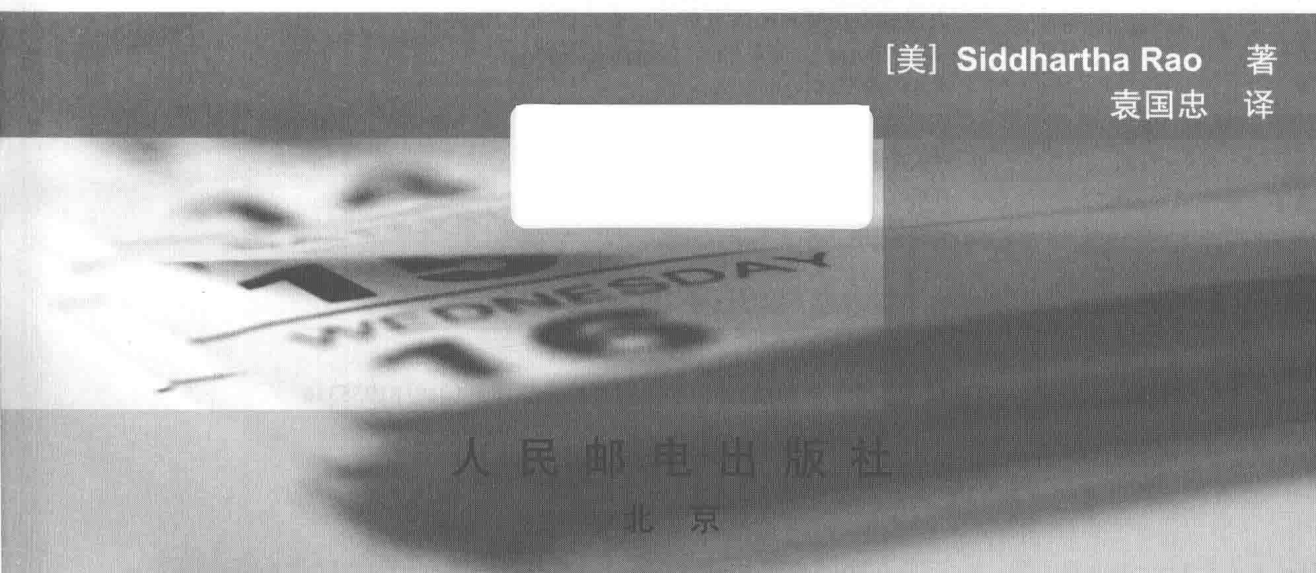
21天学通C++（第8版）

[美] Siddhartha Rao 著
袁国忠 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

21天学通C++（第8版）

[美] Siddhartha Rao 著
袁国忠 译



人民邮电出版社
北京

图书在版编目（CIP）数据

21天学通C++ : 第8版 / (美) 悉达多·饶 (Siddhartha Rao) 著 ; 袁国忠译. -- 北京 : 人民邮电出版社, 2017.9
ISBN 978-7-115-46588-7

Ⅰ. ①2… Ⅱ. ①悉… ②袁… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2017)第189196号

版权声明

◆ 著　[美] Siddhartha Rao
译　袁国忠
责任编辑　傅道坤
责任印制　焦志炜

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路 11 号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：34.25
字数：1008 千字　2017 年 9 月第 1 版
印数：1 – 3 000 册　2017 年 9 月河北第 1 次印刷

著作权合同登记号　图字：01-2016-9627 号

定价：79.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

本书通过大量短小精悍的程序详细而全面地阐述了C++基本概念和技术，以及C++11、C++14和C++17新增的功能，包括管理输入/输出、循环和数组、面向对象编程、模板、使用标准模板库、列表初始化、lambda表达式、自动类型推断等。这些内容被组织成结构合理、联系紧密的章节，每章都可在1小时内阅读完毕；每章都提供了示例程序清单，并辅以示例输出和代码分析，以阐述该章介绍的主题。为加深读者对所学内容的理解，每章末尾都提供了常见问题及其答案以及练习和测验。读者可对照附录E提供的测验和练习答案，了解自己对所学内容的掌握程度。

本书是针对C++初学者编写的，不要求读者有C语言方面的背景知识，可作为高等院校教授C++课程的教材，也可供初学者自学C++时使用。

作者简介

Siddhartha Rao 是全球领先的企业软件提供商 SAP SE 负责安全响应的副总裁。C++的发展让他深信，您能编写速度更快、更简洁、更强大的 C++应用程序。Siddhartha 酷爱旅游，还是山地自行车运动的狂热爱好者；他期待着您对本书的反馈。

献辞

谨以本书献给我的父亲，您永远是我的灵感来源。

致谢

感谢家人对我无尽的支持，感谢我的妻子 Clara，感谢为本书贡献颇多的编辑人员。

前言

对 C++来说，2011 年和 2014 年都是很特别的年份。C++11 做了重大的改进，新增了一些可提高编程效率的关键字和结构，而 C++14 做了增量式改进，完善了 C++11 引入的新特性。

本书旨在帮助您循序渐进地学习 C++，其中的章节经过深思熟虑，从实用的角度介绍这种面向对象编程语言的基本知识。读者只需每天花 1 小时，就能掌握 C++。

学习 C++的最佳方式是动手实践。本书包含了丰富的代码示例，有助于提高编程技能，请读者务必亲自动手尝试这些代码。这些代码片段都使用了最新版本的编译器进行了测试，具体地说是 Microsoft Visual C++和 GNU C++编译器，它们都支持大量的 C++14 功能。

针对的读者

本书从最基本的 C++知识开始介绍，读者只需具备学习 C++的愿望及了解工作原理的好奇心即可；虽然具备一些 C++知识会有所帮助，但这并非必须。本书也可供熟悉 C++但想了解其新增功能的读者参考。如果您是专业程序员，第三部分（学习标准模板库）将有助于您创建更优质、更实用的 C++应用程序。

本书内容

读者可根据自己对 C++的熟练程度，阅读感兴趣的部分。C++11 和 C++14 引入的新概念散落在本书各个相关的章节中。本书包含如下 5 部分。

- 第 1 部分，“基本知识”，引导读者编写一些简单的 C++应用程序，并介绍一些常见的 C++关键字。
- 第 2 部分，“C++面向对象编程基础”，介绍类的概念，您将学习 C++如何支持封装、抽象、继承和多态等重要的面向对象编程原则。第 9 章将介绍 C++11 新增的移动构造函数，而第 12 章将介绍移动赋值运算符。这些功能有助于避免不必要的复制步骤，从而提升应用程序的性能。第 14 章是一个跳板，帮助您编写功能强大的 C++通用代码。
- 第 3 部分，“学习标准模板库（STL）”，将帮助您使用 STL string 类和容器编写高效而实用的 C++代码。您将了解到，使用 std::string 可安全而轻松地拼接字符串，您不再需要使用 C 风格的字符串 char*。您可使用 STL 动态数组和链表，而无需自己编写这样的类。
- 第 4 部分，“再谈 STL”，专注于算法，您将学习如何通过迭代器对 vector 等容器进行 sort 操作。在这部分，您将发现，通过使用 C++11 新增的关键字 auto，可极大地简化冗长的迭代器声明。第 22 章将介绍 C++11 新增的 lambda 表达式，这可极大地简化使用 STL 算法的代码。
- 第 5 部分，“高级 C++概念”，阐述智能指针和异常处理等 C++功能。对 C++应用程序来说，这些功能并非必需的，但可极大地提高应用程序的稳定性和品质。在这部分的最后，简要地介绍了有助于编写杰出 C++应用程序的最佳实践，还展望了下一个 ISO 标准——C++17 有望引入的新特性。

本书使用的约定

本书使用了下述提供更多信息的元素。

注意 提供与读者阅读的内容相关的信息。

警告 提醒读者注意在特定情况下可能出现的问题或副作用。

提示 提供 C++编程最佳实践。

应该	不应该
提供当前章介绍的基本原理的摘要。	提供一些有用的信息。

示例代码

本书的示例代码可从 www.epubit.com/book/details/4780 下载。

目录

第1章
绪　论

欢迎使用本书！通过阅读本章，您将迈出成为高级C++程序员的第一步。

在本章中，您将学习：

- 为何C++是软件开发的标准；
- 输入、编译和链接第一个C++程序；
- C++新增的功能。

1.1　C++简史

编程语言旨让人更容易使用计算资源，C++并非一种新语言，但被广泛采用，仍在不断改进。本书编写期间，国际标准组织（ISO）批准的最新C++标准为2014年12月发布的C++14。

1.1.1　与C语言的关系

C++最初由Bjarne Stroustroup于1979年在贝尔实验室开发，旨在作为C语言的继任者。但不同于C语言，C++是一种面向对象的语言，实现了继承、抽象、多态和封装等概念。C++支持类，而类包含成员数据以及操作成员数据的成员方法。其结果是，程序员需要考虑数据以及要用它们来做什么。一直以来，很多C++编译器都支持C语言。

注意

要学习C++，并不要求您具备C语言编程方面的知识和经验。如果您的终极目标是学习C++等面向对象编程语言，并不需要先学习C语言等过程性语言。

1.1.2　C++的优点

C++是一种中级编程语言，这意味着使用它既可以高级编程方式编写应用程序，又可以低级编程方式编写与硬件紧密协作的库。在很多程序员看来，C++既是一种高级语言，让他们能够开发复杂的应用程序，又提供了极大的灵活性，让开发人员能够控制资源的使用和可用性，从而最大限度地提高性能。

虽然有更新的语言面世，如Java以及其他基于.NET的语言，但C++始终深受欢迎并在不断发展。较新的语言因提供了某些功能（如通过垃圾收集管理内存）让一些程序员钟爱有加，但在需要控制应用程序的资源使用和性能时，他们还是会选择C++。当前，在分层架构中，常常使用C++编写Web服务器，并使用HTML、Java或.NET编写其他组件。

1.1.3 C++标准的发展历程

经过多年的发展，C++被众多不同的平台接受和采纳，这些平台使用不同的编译器。鉴于不同的编译器之间存在差异，这导致了众多互操作性和移植方面的问题，因此需要对 C++进行标准化，让编译器厂商能够遵循标准的 C++语言规范。

1998 年，第一个 C++标准获得了 ISO 标准委员会的批准，这就是 ISO/IEC 14882:1998。从此以后，C++标准发生了翻天覆地的变化，极大地提高了 C++语言的可用性，并扩展了对标准库的支持。编写本书期间，获得批准的最新标准为 ISO/IEC 14882:2014，俗称 C++14。

注意

并非所有流行的编译器都会迅速而全面地支持最新标准，因此虽然从学术角度看，最好了解最新标准新增的功能，但就编写良好的 C++应用程序而言，这些新增功能并非是必不可少的。

1.1.4 哪些人使用 C++程序

使用 C++编写的应用程序、操作系统、Web 服务、数据库和企业软件多如牛毛，因此无论您从事什么工作，使用计算机来做什么，都可能正在使用 C++编写的软件。除软件工程师外，C++还常被物理学家和数学家用来从事研究工作。

1.2 编写 C++应用程序

当您在 Windows 系统上启动 Notepad 或在 Linux 系统上启动 Terminal 时，实际上是命令处理器运行该程序的可执行文件。可执行文件是可运行的成品，应按程序员期望的那样做。

1.2.1 生成可执行文件的步骤

要创建可在操作系统中运行的可执行文件，第一步是编写一个 C++程序。创建 C++应用程序的基本步骤如下。

1. 使用文本编辑器编写 C++代码。
2. 使用 C++编译器对代码进行编译，将代码转换为包含在目标文件中的机器语言版本。
3. 使用链接器链接编译器的输出，生成一个可执行文件（如 Windows 中的.exe 文件）。

在编译过程中，C++代码（通常包含在.cpp 文本文件中）被转换为处理器能够执行的字节码。编译器每次转换一个代码文件，生成一个扩展名为.o 或.obj 的目标文件，并忽略这个 cpp 文件可能对其他文件中代码的依赖。解析这些依存关系的工作由链接程序负责，如果链接成功，则创建一个可执行文件，供程序员执行和分发。整个过程也被称为构建可执行文件。

1.2.2 分析并修复错误

大多数应用程序很少能够一次通过编译并按预期运行。无论使用什么语言（包括 C++）编写，庞大或复杂的应用程序都需要运行很多次，以便通过测试来找出代码中的错误，即 bug。修复 bug 后，重新生成程序，再重复测试过程。因此，除编写、编译和链接等 3 个步骤外，软件开发过程通常还包括调试步骤。在这个步骤中，程序员对代码中的错误进行分析和修复。优秀的开发环境提供了帮助调试的工具和功能。