

Francis Glassborow Roberta Allen 著
徐波 译
荣耀 审校

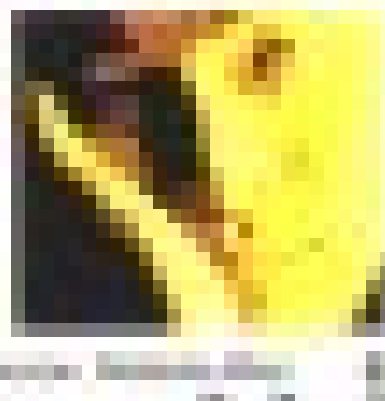
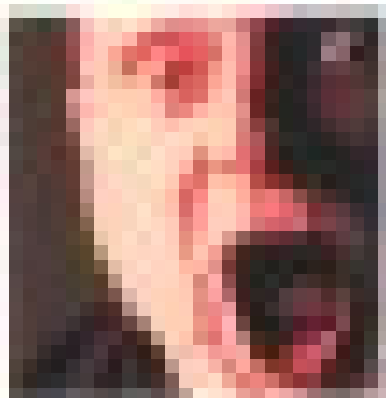
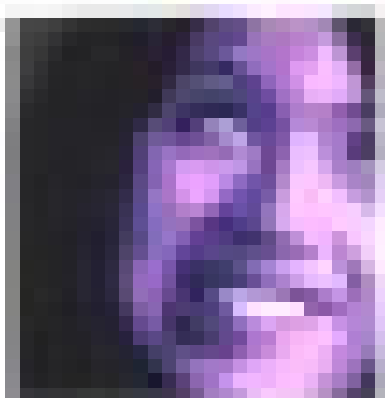
You Can Do It!

—— C++编程新手互动教程



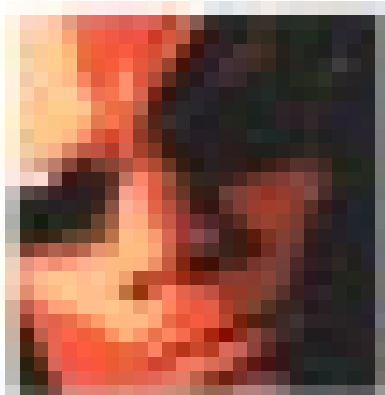
随书光盘中包含所需
软件。本书附录和词汇表。

 人民邮电出版社
POSTS & TELECOM PRESS



You Can Do It!

CHALLENGE YOURSELF



You Can Do It!

—— C++编程新手互动教程

Francis Glassborow Roberta Allen 著

徐波 译

荣耀 审校



 人民邮电出版社

图书在版编目 (CIP) 数据

You Can Do It: C++编程新手互动教程 / (英) 格拉斯博罗 (Glassborow, F.),
(英) 艾伦 (Allen, R). 著; 徐波译. —北京: 人民邮电出版社, 2005.4

ISBN 7-115-13252-6

I. Y... II. ①格... ②艾... ③徐... III. C 语言—程序设计—教材 IV. TP312
中国版本图书馆 CIP 数据核字 (2005) 第 016348 号

版 权 声 明

You Can Do It: A Beginner's Introduction to Computer Programming

Copyright © 2004

All Rights Reserved.

Authorized translation from the English language edition published by John Wiley & Sons, Inc.

本书中文简体字版由 John Wiley & Sons 公司授权人民邮电出版社出版, 专有出版权属于人民邮电出版社。

You Can Do It! ——C++编程新手互动教程

◆ 著 Francis Glassborow Roberta Allen
译 徐 波
审 校 荣 耀
责任编辑 陈冀康

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16
印张: 25
字数: 563 千字 2005 年 4 月第 1 版
印数: 1—5 000 册 2005 年 4 月北京第 1 次印刷
著作权合同登记 图字: 01-2004-3781

ISBN 7-115-13252-6/TP · 4566

定价: 45.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内容提要

本书采用互动的教学方式，使用 C++ 语言来教授新手学习程序设计。

全书共 17 章。第 1 章是对编程概念和工具的简单介绍，第 2 章到第 6 章介绍了 C++ 编程中的循环、函数、通信、类型创建和使用等技巧技能。第 7 章总结了编程的思路和技巧。接下来的 6 章（第 8 章到第 13 章）中，每一章都作为一个单元，在前 17 章的基础上添加了一些新的材料和一些新的思路。第 14 到第 16 章的主要目标是巩固所学的知识 and 技巧，并显示如何用所掌握的知识实现一些看上去难度很大的任务。第 17 章再次对常见编程问题进行了分类分析和论述。随书光盘中带有所需软件、附录和术语表。

本书适合于缺乏经验但却对学习计算机编程充满兴趣的读者，尤其适合作为高等院校学生学习编程的入门指南。

作者、译者和审校者简介

作者

Francis Glassborow 是 C++ 方面的权威，有着 30 多年的编程和教学经验，是 C++ 标准委员会的一名重要成员。作为 ACCU 的前任主席，他曾审阅了 3000 多本专业书籍，并且总是能够给出准确犀利、切中肯綮的意见，使无数的读者受益匪浅。正是由于他的这些阅历，他抱着严谨、认真的态度和科学的方法来编写这本书，力图为初学者打造一本切合实际、学以致用编程指南。

在这本书的编写过程中，作者 Francis 可谓用心良苦。他找到一位完全未曾接触过计算机工作的妇女 Roberta Allen 作为目标学生，开始按照图书编写的构想，一章一章地对她进行教授和指导，并汲取 Roberta 的反馈意见，对书稿进行修改和调整。本书正是以这种全新的、科学的编写方式完成的。因此，Francis 对本书的评价是：“如果你从来没有尝试过计算机编程，或者你尝试过但始终难窥堂奥，或者你感觉多少有些不得其法，那么，这本书很适合你。至少我希望是这样，因为这本书正是为你量身定做的。”

译者

徐波，浙江宁波人，熟悉 C、C++ 和 Java 语言。2002 年开始从事计算机技术图书翻译，第一部译作《C 专家编程》博得好评。主要译作还有《More Java Pitfalls 中文版》、《C 和指针》以及《面向对象编程 C++ 和 Java 比较教程》。

审校者

荣耀，江苏南京人，有丰富的 IT 从业经验，现任教于南京师范大学。2003 年与侯捷先生合作翻译了《.NET 大局观》、《C++ Templates 全览》（繁体中文版），引起了较大反响。主要译作还有《Windows Forms 程序设计》。

序

去年冬天，为了给 C++ 爱好者推荐一些参考读物，我撰写了《C++ 程序设计之四书五经》一文。我将文中推荐的书单发给 C++ 语言的创建者 Bjarne Stroustrup 征求建议。Bjarne 告诉我，还缺一本给完完全全的 C++ 初学者看的入门书，并向我推荐 Francis Glassborow 的 *You Can Do It: A Beginner's Introduction to Computer Programming*。

Francis 在 C++ 社群大名鼎鼎。他曾任 ACCU (The Association of C & C++ Users, C 和 C++ 用户协会) 主席，多年来评阅了成百上千本 C++ 书籍。他有着 30 余年的编程和教学经验，是 C++ 标准委员会的一名重要成员。他的阅历决定了这本书在 C++ 社群必然占有不容忽视的地位。

因为机缘，我很早浏览过这本书的英文版。我的感觉是 Francis 的本意似乎并不是想写一本关于 C++ 的入门书，而是要写一本关于如何编程的入门书，为希望体验编程乐趣的人们提供一个入门导引。恰巧 Francis 是一位 C++ 权威，于是，他采用 C++ 作为本书的编程描述语言。这个巧合使得本书既是一本优秀的编程入门书，也是一本优秀的 C++ 入门书，同时还证明了的确可以将 C++ 作为第一门编程语言来教、来学。

本书写作风格清新，让人眼前一亮。它不是从“C++ 中的 C”讲起，而是迅速展示类、名字空间、异常处理等 C++ 核心语言特性。代码普遍使用 `vector`、`string`、`set`、`map` 等标准库组件。全书以有趣的实例拉动语言和标准库的讲解，而不像有些 C++ 教材那样，例子纯属虚构，仅为演示语言特性和库的用法，很难给读者留下深刻的印象，因此教学效果也就大打折扣。

为了确保本书能够为完完全全的外行所看懂，Francis 特意邀请了一位家庭妇女 Roberta Allen 作为目标学生，进行互动写作。除了正文外，作者还安排有任务、练习、Roberta 的疑问、Francis 的解释、Roberta 的评论、任务解决方案、练习解决方案等版块。此外，每章后的总结还列出了关键的编程概念、C++ 要点等。读者应该尽可能完成 Francis 布置的任务和练习，因为要想真正掌握编程技术，你必须亲自动手写程序。对于作者提供的解决方案和宝贵的建议则要用心体会，它们有助于你提高编程技术。

我认为，如果你认真阅读了这本书，并动手实践了所有精心设定的练习，你应该可以初步掌握现代风格的 C++ 编程。不过，如果你希望跻身于职业 C++ 程序员之列，阅读这本书只能算是一个良好的开始，你还有很多东西要学，比如面向对象和泛型编程。

说到这儿，我不由得想起国内的 C++ 教材及教育现状。众所周知，许多大学 C++ 课程仍然在主讲 C，学生写出来的大多是伪 C++ 程序。作为一名 C++ 讲师，我个人认为责任并不全在教育者，我的教学经验证明一上来就讲面向对象、标准库，为数不少的学生接受起来非常吃力，花一定的时间讲述 C++ 中的 C 纯粹情非得已。尽管如此，我还是认为 Francis 的这本书可以为包括我本人在内的中国教师，提供关于如何进行新式 C++ 教学的有益启发。

当然，我并不认为这本书就适合用作正规的 C++ 大学教材，这种例子驱动的写作方式注定了本书的理论体系不够完整，概念阐述不够全面深入。它虽然很棒，但毕竟浅显了一些。我认为它最大的作用在于可以培养你编写程序、学习 C++ 的兴趣，而这正是国内 C++ 教材和 C++ 教育最最欠缺的！

因此，我认为，如果你是一名完完全全的 C++ 初学者，此前对编程如雾里看花，虽然正忍受着糟糕的 C++ 教材和教育的折磨，但对编程的热情火焰尚未完全熄灭，不妨读一读本书并动手实践，希望它可以重新唤起你对 C++ 编程的热爱。

退一步来说，即使你不是一名学生，比如说，是中关村或珠江路上摆弄电脑硬件的新青年，但对编程充满了渴望，或是一名 Web 网站设计者，希望体验 C++ 编程之乐趣，这本书都可以让你心想事成。

编程对于初学者来说是一件神秘的事情，而当程序能够如愿运行时，那种快乐无与伦比。我还记得大学时使用 SUN 工作站编写第一个 Basic 程序时（那时 C++ 尚处于发育期）从激动到沮丧再到兴奋的体验。我折腾了好久也没能让计算 1 加 1 的代码算出 2 来，在克服“重重困难”运行出结果后，我几乎欢呼出来！

编程不仅可以成为一种兴趣，让我们乐在其中；编程也可以成为一种职业，我们靠它养家糊口。可以肯定的是，几乎所有程序员都会从编程本身获得快乐，尤其当我们的工作为别人所欣赏时。

本书译者徐波先生文风轻松流畅，译文读来甚为舒坦。作为一名 C++ 老手和讲师，我饶有兴味地读完了这本入门读物，相信作为完完全全的编程新手的你，定会从中获得更大的快乐。我甚至幻想我的爱人有朝一日也能阅读本书并动手体验编程之乐趣，体会我为何心甘情愿年复一年、日复一日佝偻于手提电脑前，写程序，写关于 C++ 的一切。

荣 耀

2004 年 12 月

于南京师范大学

www.royaloo.com

译者序

初次浏览这本书的时候，我心里颇有几分不以为然。虽然我并不认为只有天赋过人者才能学会编程，但编程怎么也应该算是一种高层次的智力活动。可是在本书中，作者所面向的读者居然可以是任何层次的人，读者甚至不必有多少计算机和数学基础，这也太小看编程了吧。更何况，作者使用了公认的最为难学的一种语言（C++）来讲述编程。

留意到本书的作者，我才知道他是大名鼎鼎的 ACCU 组织的前主席。ACCU 是 Association of C and C++ Users（C 和 C++ 用户协会，www.accu.org）的缩写，是一个著名的编程技术组织，经常召开世界范围的编程技术大会和研讨，组织网上社群交流，广泛评阅数以千计的技术书籍并共享阅读经验。这个组织在技术社群有着很大的影响力，它拥有许多重量级的高手，作为前主席，Francis Glassborow 更是个中翘楚。

作为一名具有多年编程教学经验并且经常在 ACCU 的技术刊物上撰写技术文章的学者，Francis Glassborow 在 C++ 编程的教学方面具有非常独到的经验。他以这种互动的新型方式推出这本编程入门教程，肯定有其独具匠心之处。在仔细阅读和翻译本书的过程中，我越来越领会到作者的良苦用心，对本书的态度也变得越来越欣赏。

这本书的最大特点就是“互动性”，作者邀请了一位年过 50 且以前从来没有编程经验的妇女（Roberta）参与本书的写作过程。中国唐代诗人白居易在写完诗后，常常先读给老妪听，若对方不懂，则再行修改，直至“老妪能解”。估计 Francis Glassborow 不知道这个典故，但他所采用的方式却是惊人地相似。这或许就是他相信任何人都可以用这本书学会编程的信心所在吧。

诚如作者在书中所言，要想学会编程，至少要具备两个条件，一是对编程感兴趣，二是愿意花时间和精力学习编程。我也相信，任何人只要具备了这两个条件，学会编程绝对不在话下。当然，一本好的指导书可以帮助读者有的放矢地学习编程的各个方面，达到事半功倍的效果。本书就是这样一本可以帮助读者正确学习编程的好书。

C++ 是一种非常复杂的语言，所涵盖的内容非常广泛。本书并没有完整地讲述 C++，有许多重要和常用的 C++ 特性都没有出现在书中。站在作者的角度，他认为当务之急是帮助读者掌握如何利用已经学会的工具来解决问题，也就是编程的方法论，而不是完整地学会一门编程语言。许多读者（包括当年初学编程的我）似乎有这样一种想法，学会了一门编程语言就相当于学会了编

程。其实不然：语言只是工具，重要的是如何用这个工具解决你需要处理的问题，而不是把主要精力放在如何全面仔细地对工具进行研究。

本书之所以使用相对复杂的 C++ 来描述编程，很大程度上是因为 C++ 是当前的主流编程语言之一，并且它与另外几种主流语言 C、Java、C# 在语法上极为相似。C/C++/Java/C# 合起来称为 C 族语言，在当前占据着主导地位。此外，在教学方面，C 族语言也逐渐取代了从前的伪 Pascal 语言。

本书的译稿得到了南京师范大学荣耀老师的审阅，他提出了许多宝贵的修改意见，并欣然为本书作序。在序言中，他从一个 C++ 一线教师的角度对本书给出了客观的评价和积极的阅读建议，相信对读者会很有帮助和启发。译者在这里向荣耀老师表示诚挚的谢意！

虽然译者想尽力译好这本书，但疏漏之处在所难免，欢迎读者批评指正。译者的 E-mail 地址是 parassite@sina.com，读者如有问题，可随时与译者联系。

徐 波

前 言

购书之前

如果你正在决定是否购买本书，请仔细阅读本节的内容，以便对本书有一个充分的认识。我将尽自己最大的努力帮助你作出正确的结论。虽然我希望自己的书卖得越多越好，但我并不希望你浪费时间和金钱购买以后会令你后悔的东西。

在计算机编程领域，这是一本非常独特的书，因为它是为普通人编写的。任何人只要拥有一台计算机（在目前这个阶段，它还需要运行某个版本的 Microsoft Windows），想知道编程是怎么回事，而且打算花点时间通过学习编程来满足这方面的好奇心，我相信，这本书肯定能够帮助他轻松地学会编程。

本书是我和 Roberta 的合作成果。在本书中，我的角色是一位知识广博、经验丰富的教师，而 Roberta 的条件应该和本书的读者相似。在本书的技术内容方面，Roberta 的贡献并不大，但她仍然在本书的写作过程中发挥了极大的作用，帮助我精心安排难度和进度，使那些计算机水平只不过能够启动程序、使用字处理程序、收发电子邮件以及在 Internet 上进行冲浪的人也能够跟得上学习的进度。当她以学生身份半途加入到本书的写作过程时，虽然她使用计算机已有 10 年之久，但她仍然没有掌握诸如目录结构之类的概念。16 岁以后，她就再也没有接触过数学课程。她的数学水平也就是会一些简单的算术，顶多只能在公司里干一些简单的财务工作。

不过，她也有两个优点。其一，她希望了解编程；其二，她相信我能够教会她编程。这两点非常重要。如果你希望使用本书，你也需要做到这两点。你必须愿意投入时间和精力了解编程的基础知识，并且信任我们（Roberta 和我）能够帮助你实现这个目标。只要你能够满足这两个条件，我们可以承诺你肯定能够学会编程。并且，你根本用不着学完全书就能够具备自行编写程序的能力。只要稍微运用一些想象力，你肯定能设计出一些能够展现自己独特风采的程序。Roberta 在学完第 6 章之前就完成了她的第一个完整的源程序。当她完成学业时，她已经为她的孙子孙女们编写了好几个程序，其中至少有一个程序是完全按照她自己的兴趣所编写的。

我之所以告诉你这些，是因为我确信任何人只要有兴趣都可以学会简单的编程。我还相信，

很多人会认为学会编程将在多个方面使他们受益。当一个程序最终能够顺利运行且能够实现设计目标时，所有程序员都会为之振奋，并有一种成功的感觉。

如果你粗略浏览本书，你可能会疑惑你是否能够看懂书中那些稀奇古怪的文本¹。不必担心，你很快就会发现，所有这类文本只不过是计算机用来表达意图的一种方式，并没有任何稀奇古怪之处。它并不是普通的语言，尽管其中确实有一些英文单词。这是一种称为 C++(读作 **cee plus plus**) 的计算机语言。如果你周围有一些对编程略知一二的朋友、亲戚或同事，当他们听说本书是用 C++ 讲述编程时，他们可能会向你反馈一些可怕的警告。相信我，不要被他们的话吓倒，其实他们都受到了误导。

为什么要用 C++

让我问一个不同的问题：“为什么要用英语？”你的回答大概是：“因为这是我讲话所用的语言。”问得再深一点，哪种语言是人类使用最为广泛的语言呢？以中文为母语的人数多于任何其他语言，但我为什么不用中文写作呢？另一方面，像西班牙语或斯瓦里希语之类的语言远比英语易学，但我为什么不用它们来写作呢？

我们知道，英语不仅是几亿人的母语，而且以它为第二语言的人更是数量惊人。如果你是个初次访问地球的外星人，毫无疑问，你会选择英语与地球上的人进行交流。尽管英语非常复杂，但是如果人们必须与他们社区之外的人进行交流，绝大多数人都会把英语作为第一选择。

C++在计算机社区中的角色与英语在人类社会中的角色非常相似。它是一种功能强大、语法复杂的语言。当然，对于新手而言，它确实存在着许多难解的地方，可以说处处隐藏着陷阱。但是，它确实又是应用最为广泛的通用目的的计算机编程语言。就像没有人敢声称自己完全掌握英语一样，彻底精通 C++的人也可以说是凤毛麟角。但是，就像我们并不需要掌握英语的全部内容一样，我们也不需要精通 C++的所有方面。本书并不是一本讲述 C++的书，即使你学完了本书，你也称不上是位 C++程序员。但是，你将成为一名能够使用 C++来表达问题的解决方案并且能够编写程序来满足现实需要的程序员。

为什么要用 C++？因为它并不会妨碍我向你展示如何进行编程。其他编程语言可能更简单一些，但我常常觉得它们没办法让我用一种非常简洁的方式向你展示编程问题的答案，这常常令我有受挫的感觉。使用 C++，我可以从它的丰富功能中撷取一些精华，向读者提供功能强大的工具，允许他们更为方便地在程序中体现自己极富创意的思路。

C++存在一个小小的缺点。作为一种基本语言，它缺乏进行图形开发的工具。不过这个问题很容易得到解决，因为我可以 C++编写这些工具，供读者使用。我需要一些专家帮助我开发这些工具，这是由于计算机的复杂特性所决定的：不同的计算机可能使用不同的图形工具，不同计算机屏幕所使用的颜色数量也不一定相同。使用 C++，我可以清楚地说明我需要的是什么，这样

¹ 译注：指书中的源代码。

我的一位同事 (Garry Lancaster) 就可以把我所写的规范转换为能够在所有 MS Windows 机器上运行的工具。最终 (甚至可能在你看到本书之前), 我会请另外的人改写这些工具, 这样其他机器也能够运行它们。但是, 在此之前, 我们不得不接受一个人为的限制, 也就是说如果你在程序中使用了我所提供的工具, 那么你的程序就只能在某种版本的 MS Windows 上运行。

最后, 我还需要一些简单的工具, 你可以用它们编写程序, 并且管理一些技术细节, 使你所写的东西能够为计算机所理解。这些工具是由另一位作者 Al Stevens 提供的, 他允许我发布他所开发的 Quincy 工具集, 帮助新手学习编程。

Garry Lancaster 和 Al Stevens 等人所做的工作说明了计算机社区的最大优点: 优秀的作品可自由共享。其结果是, 你可以得到学习程序所需要的任何东西 (除了计算机以及时间和精力)。

选择权完全在你的手中。如果你希望学习编程, 并进而学习别人的计算机程序是如何运行的, 你肯定能够实现自己的目标。Roberta 和我足足花了 9 个月时间编写这本书 (但这并不意味着你需要花 9 个月时间来读完它; Roberta 在此期间还有一个非常重要的任务, 就是说服我改进本书的文字, 使其他人更容易读懂)。你第一次面临一个真正的选择, 是否要在学习编程的道路上走下去。我们已经完成了我们所能做的一切, 剩下的就靠你自己了。

最大限度地利用本书

既然你已经决定购买本书, 我将向你提供一些建议, 以便最大限度地利用本书。

在理想情况下, 你不应该独自学习。注意, 我用了“在理想情况下”这个词。在现实中, 你很可能发现自己别无选择, 只能依靠自学。但是, 只要有可能, 你要尽量避免这种情况。脑子里有了这个想法之后, 你可以把我接下来所说的内容当成是一种有益的建议, 而不是学习本书的必需条件。

有两种人可以帮助你: 学习伙伴和指导老师。学习伙伴应该是那种与你能力相近, 并且你乐意与之共同学习的人, 他愿意与你共同分享成功和失败。学习的过程也包括了犯错误。犯错误并没有什么值得羞愧的。有时候, 我们可能会为自己的愚蠢感到可笑; 有时候, 我们可能会对学习伙伴的洞察力印象深刻。在任何时候, 我们都不应该嘲笑别人。错误可能是令人发笑的原因, 但你却可以从别人所犯的错误中吸取教训。从这个意义上讲, 犯错误的人也是值得你尊敬的。

如果在认识的人里面, 你找不到能够与你一起学习的人, 你可以试试是否能够通过 Internet 找到学习伙伴。你可以通过 CD 上名为“Link to Website.html”的文件 (当你安装 CD 时这个文件会拷贝到你的硬盘上) 访问本书的网站。为了更好地共同学习, 你要尽量拓宽与学习伙伴的交流面, 不要仅仅局限于编程方面的技术问题。即使只是通过电子邮件, 正常的人际交往比绝大多数人想象的更为重要。你需要把学习伙伴当成朋友看待, 而不是仅仅把他当成学习伙伴。

另外, 我还向你提供了一位虚拟的学习伙伴, 也就是我的合作者 Roberta。她是第一位见到本书内容的人, 初稿中有些内容她认为对学习帮助不大, 因此我接受她的建议把它们给删掉了。书

中还包括了她对每一章内容的评价以及她的学习经验。有时候，我还在书中包括了她所提出的问题以及我的回答。在前言的最后，她对自己作了一番简单的介绍。在接下来的内容里，她就像是我的另一双敏锐的眼睛。每位撰写科技读物的作家都应该有这样一位助手。如果你觉得这本书非常容易读懂，其中就有她有功劳；如果你仍然觉得本书有些地方难以读懂，你仍然应该感谢她在提高本书的易读性方面下了许多功夫。

第二种能够帮助你的人就是指导老师。他应该是一位专家，当你迷失方向时能够把你拉回正路，支持和鼓励你走向成功。优秀的指导老师是无价之宝，但是差劲的指导老师却可能给你带来灾难。

你应该很容易就能够分辨出什么样的指导老师是比较差劲的，因为他们想告诉你本书并不包括的所有类型的东西。水平不够的指导老师希望你从头开始，沿着他们曾经走过的路重新学习一遍。指导老师如果不能在你的学习过程中向你提供有益的纠错意见，让你感觉踏实，那纯粹是浪费你的时间。优秀指导老师的品质就是他在指导你的过程中也愿意向你学习。换句话说，他们是真正的专家，渴望学习新思路、新观点和新见识。我之所以热爱教学，是因为我不仅能够向学生传授知识，而且能够从学生那里吸收有益的东西。

如果你无法找到一位合适的指导老师，你可以再试试本书的网站。你也可以不请指导老师（这比请一位差劲的指导老师强），并使用网站上的资源作为模板答案（书中有所提供，本书的网页中也有所提供），或者你也可以访问新闻组如 `alt.comp.lang.learn.c-c++`。但是，你必须学会小心甄别，因为新闻组里虽有很多好东西，但垃圾也不少。

许多现代书籍的写作目的似乎就是让读者立即感到满意，并且认为读者只会阅读一遍。这意味着每页书所承载的信息量非常有限。作者们试图用多种方法描述同一样东西，因为他们期望读者对每页书只读一遍就能读懂。我不是这样做的。我期待你进行学习而不仅仅是阅读，也就是说要经常根据需要重新返回来阅读，以便更好地理解以后的内容。这需要花费时间。我花了9个月写作本书，同时 Roberta 也花了9个月理解这本书。也许是由于 Roberta 帮助我对本书作了改进，你很可能只要花6个月就可以完成本书的学习了。可你也不要指望能够花更短的时间掌握本书的内容。但是，在你完成本书之前，你早已具备了编程的能力。

我希望你认真学习每一章，并且随着理解的加深，重新复习以前的章节。与普通的训练课程相比，这是本书最大的优点之一。本书的第二个优点是你可以根据自己的特点设置学习进度。有些东西你可能很轻松地掌握，有些东西可能要花一点时间。有些你能够快速理解的东西在别人那里可能要费一番心思才能领会。

我希望你能够在计算机前面完成本书绝大部分的学习。仅仅依靠阅读是不够的，你还需要进行实际操作。我假定你将会输入书中所有的代码，即使我并没有催促你这样做。Roberta 所总结的一个教训就是她在学习本书时跳过了一些代码，从而造成了学习上的麻烦。相信她的话，多花点时间进行练习和编码绝不会没有价值。

那些稍通编程的人如果偶尔浏览一下本书，他们可能会对前面章节的一些材料感到惊奇。他们已经习惯了冗长、乏味和重复的大部头书，并且学习的步伐相当缓慢。

我希望你能够觉得这本书与众不同。本书的目标是解释编程，并要求你使用 C++ 工具的一个有限子集对编程进行挑战。当你进一步学习时，你将学会使用越来越多的工具，但是对编程的挑战却是一个永恒的主题。

学习目标

你能够实现的所有目标都是由简单的工具（标准 C++ 和我的程序库）所实现的。我想你可能会惊叹这些简单的资源所能够实现的目标。我花了不到一天时间摆弄 Playpen（一个绘图库函数，后面会用到），就觉得自己已经掌握了它的所有内容。你需要领会非常关键的一点：在编程中你能够实现的目标仅仅受限于你的想象力。例如，编写程序让计算机下棋并不困难，但是要让计算机的下棋水平达到很高却是非常困难的，这也是为什么必须要由专家来编写对弈程序的道理。

你还将学习一些技巧，完成一些有用的任务，例如生成简单的动画、基本的数据处理以及数值计算等，这将充分考验你的编程基础知识。本书的主题就是让你具备编写简单程序的能力，并且认识到编程并不像有些专家所说的那么困难。是的，C++ 确实存在一些阴暗的角落、奇怪的陷阱以及反常的特性，但是你在完成令自己骄傲的编程任务时并不一定需要靠近它们。

学习编程应该是一种非常有价值的经历。如果你并没有这样的感觉，那么或者是不适合编程，或者是你的学习方式不对。我希望，通过钻研这本书，你能够发现自己不仅能够进行编程，而且喜欢进行编程。

对学生的建议

绝对不要放弃，但是要学会寻求帮助。你应该足够聪明，分得清喜欢求助与懒惰之间的区别。如果你不理解一个问题，你应该寻求帮助，获得更多的信息。当然，这些信息应该只是帮助你找到正确的方向，而不是直接告诉你问题的答案。

如果你曾经由于时间匆促，请别人帮你编写过程序，你应该坦率地承认这个事实，并且至少要理解别人所编写的程序。由于程序不是你亲自完成的，你失去了大好的锻炼机会。但是，如果你能够理解别人所写的程序，你至少和那些连程序都不看的人还是存在巨大的区别。

对教师的建议

本书基于作者多年的课堂教学经验以及超过 30 年的编程经验。你应该把精力集中在学生的要求上，并且向学生展示自己优秀的教学素质。其中包括乐于倾听学生的声音，渴望理解学生的要求。他们并不期望你知道所有的答案，但他们希望你是一位真诚的教师。如果你并不知道学生的某个问题的答案，你就实话实说，并且能够花点时间寻找答案。

如果你把本书作为教科书使用，请不要违背它的精神。本书的设计目标是以一种探索旅程的方式，介绍基本的编程知识。读者应该使用已经掌握的工具探索新的目标，而不是不断地寻找新的工具。

你应该理解，用简单的方式完成任务需要技巧和洞察力，并且你应该赞赏学生最初所递交的复杂解决方案。但是，你要鼓励他们寻找更为简单的解决方案。程序能够运行并且能够获得正确的结果是不够的，它还应该清楚地表达问题的解决方案。

不要向学生提出并非必需的要求，以免加重他们的负担。像注释这样的东西应该适当地使用，你不应该把它作为评判学生作业优劣的标准。如果你觉得学生在程序中所作的注释太少，那你就过上几个星期，然后再要求他们解释自己的程序。如果他们能够清楚地进行解释，那么他们原先所作的注释已经足够。代码本身就是最好的注释，如果它需要进行注释的地方越多，你就越应该怀疑它的质量。

作者介绍

第一作者 Francis Glassborow

我们将一起度过许多时间，因此请容许我介绍一下自己。如果你不感兴趣，你也可以跳过这段内容。但是，我觉得我的自我介绍或许可以满足你的好奇心。

我生于1942年（生日是6月3日，这是为了满足那些喜欢刨根问底的人），是6个孩子中的老大。1949年，我父亲去苏丹工作。他把所有的孩子都送到当地的一所教会学校，因为他觉得生活在一种外国文化氛围中要强于接受任何英语小学教育（我认为这是对的）。我所就读的这所学校共有2000名学生，使用3种教学语言，第一语言为英语的学生最多也不会超过6个。我不得不学习阿拉伯语和意大利语。当我9岁的时候，我准备转到一所英语预科学校。因此，我还通过一位私人教师学习拉丁文，并从我的母亲那里学习法语（她讲一口流利的法语，曾经在一所法国教会学校接受教育）。

1953年9月，我开始在一所英语预科学校学习。在那里，我还学习了古希腊语。1956年4月，我进入Downside学习，这是一所天主教公立学校。1960年10月，我进入牛津的默顿学院。在那里，我攻读数学专业，并获得了一个三级荣誉学位。正如我的老师在数年后所说的那样，这个学位是比较令人失望的。但是，重要的是我在牛津得到了许多课堂之外的收获。例如，在1961、1962和1963年，我代表牛津参加了与剑桥的柔道对抗赛。在1962年，我是牛津大学柔道俱乐部的主席。

早年我从事数学教学。在牛津的克伦威尔学校负责计算机教学之前担任数学系的主任。随后，我开始自学编程，并编写了一些程序让我的学生使用。1982年，我实现了Forth编程语言，为我的学生提供了一种可移植的编程资源。这样，他们不仅可以在他们自己的Sinclair ZX Spectrum机器上编写程序，而且可以在学校的Research Machines 380Z上进行编程。我的一位同事设计并制

造了一种硬件，将一台 Spcetrum 连接到一台 380Z 上。我设计了一种协议，让这两台机器可以通过连接交换信息。

1988 年，我从教师岗位上退了下来，因为向同事们提供计算机方面的支持所产生的压力损害了我的健康。同年，我加入了英国的 C Users Group (C 用户组)，也就是 ACCU 的前身。在上世纪 90 年代的绝大多数时间里，我是这个组织的主席。并且从 1990 年 8 月直到 2001 年 12 月，我一直是这个组织的主要刊物的编辑。

1990 年，我加入了 BSI 小组，参与 C 和 C++ 的标准化。从此，我便代表英国参加了 ISO/IEC SC22/WG14 (C) 和 WG21 (C++) 委员会。在最近几年里，我是这些工作组中英国代表的负责人。

我还是 RYA (皇家游艇协会) 的一位资深游艇教练，并且喜欢玩定约式桥牌。我有两个孩子和一位美丽的有残疾的孙女 (出生于 2001 年 5 月)。

对于你，我只知道你希望进入编程世界。欢迎你！我希望这可以丰富你的人生，因为对于我而言确实是这样，至少让我遇到了许多聪明和有趣的人。没有这些人，我想也不会有本书的面世。

学生作者 Roberta Allan

我在牛津出生、成长并接受教育，至今我仍然生活在那里。16 岁那年我就离开了学校，很早就结婚，并且有一个儿子和一个女儿。早年，我尝试过许多业余工作以贴补家用。我干过电话接线员，当过酒吧侍者，做过体育场引路员，甚至曾经在牛津的繁华市场中担任一位屠夫的售肉服务员。

当我的孩子都进入学校之后，我决定重新学点什么，并以成人学生的身份进入威斯敏斯特学院。最初，我想攻读一个教师学位，但后来改为修习神学。当我获得一级荣誉学位时，我兴奋异常。

后来，我加入了我丈夫的塑料注塑模具厂，担任公司主管，负责管理、销售和质量部门，并负责日常管理。正是在这里，我第一次遇到了 Francis，他当时是我们公司的计算机顾问。

当我们建立了一支强有力的管理队伍后，我觉得可以重新继续自己的学业。我进入曼彻斯特学院，获得了一个神学方面的文凭。我主要学习希腊语和希伯来语，以便进行更深入的圣经研究。在接下来的几年里，我继续钻研约翰的福音书，并且撰写了一本书，不过我没有想过要出版它。

在这几年里，Francis 继续帮助我解决计算机问题。严格地说，我并不是一位技术恐慌者，但是对这些可怕的机器仍然心存畏惧。但是，我喜欢 Internet，我发现它在研究和娱乐方面都有无穷的价值。我希望自己最终能够拥有自己的网站。因此，我慢慢开始学习设计网站，目的是发布自己的书籍以及其他一些文章。

随着我对精神上的东西兴趣日益广泛，最近我又喜欢上了一些深奥的主题，包括占星术和犹太教灵学。