国外计算机科学经典教材

初学编程者的最佳教程

C++

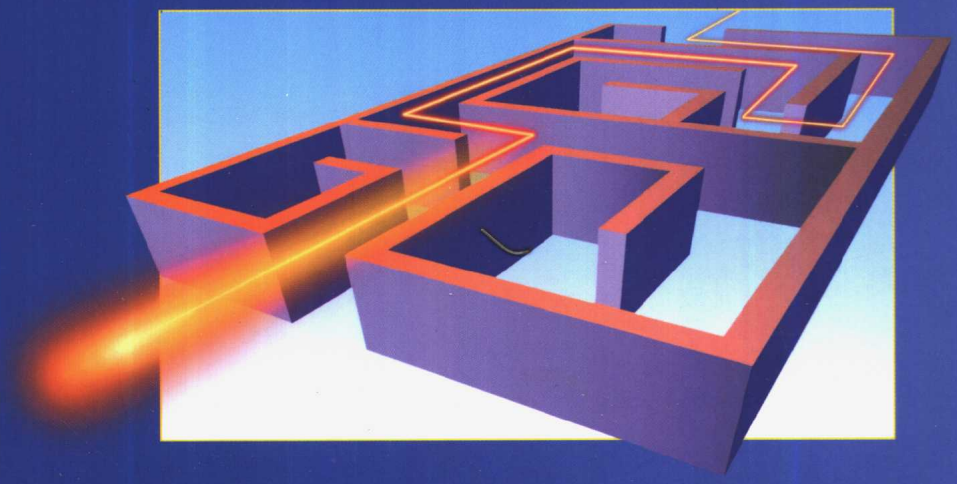
基础教程

创建、编译和运行C++程序

使用流程控制语句、数组、字符串和函数

学习类、对象、继承、重载和I/O

使用所有的C++编译器，包括Visual C++



源代码免费下载

Herbert Schildt 著
张林娣 译

清华大学出版社
http://www.tup.tsinghua.edu.cn
McGraw Hill

C++基础教程

Herbert Schildt 著

张林娣 译

清华大学出版社

(京)新登字 158 号
北京市版权局著作权合同登记号：01-2002-4620

内 容 简 介

C++语言被许多程序员视为开发高性能软件的首选语言。现代很多语言都衍生自 C++，并深受其影响。

本书讲解 C++的基础知识和编写 C++程序的基本技能。内容涵盖 C++所有的核心概念，同时也讲述了一些高级主题。主要内容有：C++基础知识、C++的数据类型和运算符、程序流程控制、数组、字符串和指针、函数、类和对象、继承和多态性、C++ I/O 系统、异常处理、重载和模板等等。本书内容安排合理、结构清晰，在各章和各节的最后都提供有测试题供读者自我测验。每章还提供了大量的示例代码和练习机会。

本书由 C++编程专家编写，内容浅显易懂，适合 C++初学者学习。

图书在版编目(CIP)数据

C++基础教程/(美)舒尔茨著；张林娣译.—北京：清华大学出版社，2002

书名原文：C++：A Beginner's Guide

ISBN 7-302-06099-1

Ⅰ.C… Ⅱ.①舒…②张… Ⅲ.C 语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 091357 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑：夏兆彦
封面设计：康博
版式设计：康博
印 刷 者：北京密云胶印厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：31.5　**字数**：806 千字
版　　次：2002 年 12 月第 1 版　2002 年 12 月第 1 次印刷
书　　号：ISBN 7-302-06099-1/TP・3644
印　　数：0001～4000
定　　价：58.00 元

前　言

C++是开发高性能软件的一门非常优秀的语言。它所定义的现代语法和风格影响了随后出现的所有编程语言。例如，Java 和 C# 都源自 C++。而且 C++是使用非常普遍的编程语言，几乎所有的专业程序员都可以使用这门语言编程。掌握 C++将为您完成当今任何编程任务打下牢固的基础。

本书讲授 C++编程的基础知识。它采用循序渐进的方式，并配以许多示例、自测题目和项目。学习本书不要求您具备编程经验。本书将从基础知识讲起，如怎样编译和运行 C++程序。然后讨论组成 C++语言的关键字、特征和构造。通过学习本书，您将牢牢掌握 C++编程的基本知识。

在开始之前，需要声明一点：本书只是一个起点。C++是一门庞大、复杂的语言，成功的 C++编程不仅仅涉及定义语言的关键字、运算符和语法，它还涉及如何使用范围广泛的类和函数库来帮助开发程序。虽然本书讨论了库的几个元素，但限于篇幅，大多数都没有讨论。要成为一流的 C++程序员，还需掌握庞大的 C++库。在学完本书后，您将具备探索 C++的库和其他领域的必要知识。

本书组织结构

本书以一种均衡的步调组织各个部分，每一部分的内容都构建在前一部分的内容之上。本书共有 12 章，每章讨论 C++的一个方面。本书的独特之处在于，它包括了一些特色段落来强化您所学习的知识和技能。

本章学习目标

每章的开始列出本章要达到的目标。

本章自测

每章章末有一个自测，检验您所学的知识。其答案在附录 A 中。

一分钟练习

在讲解完每个主要部分时都附有一个小练习，测试您对前面所学关键点的理解。答案就在问题的下面。

专家答疑

遍布全书的“专家答疑”包含关于某个主题的补充信息和有趣的评论。

项目

每章都包含一个或几个项目，说明如何应用所学的知识。它们都是真实的示例，您可用作自己程序的起点。

不要求编程经验

本书不要求读者有编程经验。即使以前从没有编写过程序，也可以学习本书。自然，现在大多数读者至少都有一点编程的经验。其中很多人的经验都是从 Java 或 C# 获得的。您将会了解到，C++就是这两种语言的起源。因此，如果您对 Java 或 C# 有所了解，就能够很轻松地学习 C++。

需要的软件

要编译和运行本书的程序，需要安装一个现代的 C++编译器，如 Microsoft 的 Visual C++ 或 Borland 的 C++ Builder 的最新版本。

免费代码下载

记住，本书所有示例和项目的源代码都可以从 www.osborne.com 免费下载。

目　　录

第1章 C++ 基 础

本章学习目标

- 了解 C++的发展史
- 学习 C++与 C、C#及 Java 的关系
- 学习面向对象程序设计的 3 条准则
- 创建、编译及运行 C++程序
- 使用变量
- 使用运算符
- 处理 if 和 for 语句
- 应用代码块
- 了解 C++关键字
- 理解标识符
- 使用函数

当今时代，如果说有哪种语言能够表达编程本质的话，那么它就是 C++。C++是开发高性能软件的优秀编程语言。它的语法已经成为专业编程语言的标准，它的设计原理贯穿在整个计算领域。C++也是 Java 和 C#的始祖。简单地说，要想成为一名专业的编程人员，那就必须精通 C++。它是通往现代编程领域的入口。

本章简要介绍 C++，包括它的发展历史，它的设计原理和几个最重要的特性。到目前为止，学习一门编程语言最困难的事情就是没有哪个组成部分是孤立存在的。相反，语言的所有组成部分都是共同发挥作用的。这种相关性使得很难单独介绍 C++的一个方面而不涉及其他内容。为了帮助克服这个问题，本章将简要介绍 C++的几个特性，包括 C++程序的通用形式，一些基本的控制语句和运算符。本章不讲述太详细的内容，而是集中阐述所有 C++程序共有的一般性概念。

1.1 C++简史

C++的发展历史起始于 C 语言。这非常易于理解：C++建立在 C 的基础之上。因此，C++是 C 的一个超集。C++扩展并增强了 C 语言的功能，以支持面向对象编程(我们将在本章后面讲述面向对象编程)。同时 C++还对 C 语言的其他几个方面进行了增补，包括库例程的扩展。然而，C++的思想和风格还是直接继承于 C。因此，要想完全理解和掌握 C++，就需要对 C 语言有基本理解。

1.1.1 C 语言：现代编程的开端

C 语言的发明标志着一个崭新的编程时代的开端。它的影响决不可低估，因为它从根本上改变了编程设计和思考问题的方式。它的设计原理和语法对以后出现的所有主要语言都产生了影响。在计算领域，C 曾是主要的、具有革命性的新语言。

C 是由 Dennis Ritchie 在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上发明并首次实现的。C 是一门更“古老”的语言 BCPL 发展演进的结果。BCPL 语言是由 Martin Richards 开发的。BCPL 语言对 B 语言产生了重要影响，B 语言由 Ken Thompson 开发，它导致了 20 世纪 70 年代 C 语言的出现。

在 C 语言发明之前，计算机编程语言的设计通常被视为一门学术活动，或者说它是由委员会的学究们开发的。C 不同。它由真正的编程人员设计、实现和开发，反映了他们处理编程任务的方式。它的各种特性都由那些真正使用语言的人经过琢磨、测试和再三思索过。因此，C 吸引了大量支持者，并迅速成为软件界人员的首选编程语言。

C 发源于 20 世纪 60 年代的结构化编程革命。在结构化编程出现之前，大型程序非常难于编写，因为程序逻辑总是倾向退化为众所周知的“意大利面条式的代码”(spaghetti code)，杂乱无序的跳转、调用和返回使得人们难于理清其中的头绪。结构化编程语言通过添加定义清晰的控制语句、带有局部变量的子例程和其他改进措施解决了这个问题。使用结构化的编程语言，才可能编写非常大型的程序。

尽管当时还出现了其他的结构化语言，例如 Pascal，但是 C 率先实现了集功能强大、简洁易用和表达式丰富等众多特征于一身。它的简洁、但却易于使用的语法与其基本设计理念直接相关，其基本的设计理念是“程序是由程序员来管理的，而非由语言控制的”。这让很多人改投到 C 语言门下。这种轰动效应在今天看来的确有些难于理解，但是 C 却带来了程序员们期待已久的一股清新气息。因此，C 迅速成为 20 世纪 80 年代使用最为广泛的结构化编程语言。

1.1.2 对 C++的需求

在讨论了上述问题之后，您也许会不解，为什么要发明 C++。既然 C 是一种成功的计算机编程语言，那为什么还需要其他语言？答案在于复杂性。纵览编程技术的历史，程序逐渐增加的复杂性驱使出现更好的方法来解决这些复杂性。C++就是应运而生的产物。为了更好地理解程序逐渐增加的复杂性和计算机语言发展之间的相关性，我们来考虑下列问题。

在计算机发明之后，编程方法发生了巨大的变化。例如，当计算机刚发明时，编程是通过在计算机面板上输入二进制的机器指令来进行的。当时程序只有几百个指令那么长，这种方法还可以奏效。随着程序的增长，出现了汇编语言，因此使用机器指令的符号表示法，程序员就可以处理更大型的、日益复杂的程序。当程序进一步增长时，高级语言就出现了，它们给程序员提供了更多的工具来处理这些复杂性问题。

当然，第一个广泛使用的计算机语言是 FORTRAN。虽然 FORTRAN 给人留下了深刻的印象，但它说不上是一种实现了清晰、易于理解的程序的语言。在 20 世纪 60 年代诞生了结构化的编程方法，它是 C 之类的语言所提倡的编程方法。结构化编程语言的出现才可能实现轻松地编写适度复杂的程序。然而，即使是使用结构化的编程方法，一旦项目达到一定的规模，其复

杂性仍然超出了程序员的管理能力。到20世纪70年代末期，许多项目已经接近或者达到了这种规模。

为了解决这种问题，一种新的编程方法出现了，这就是面向对象编程(object-oriented programming，OOP)。使用OOP，程序员可以处理更大型、更为复杂的程序。而问题是C语言并不支持面向对象编程。对C语言支持面向对象编程版本的期待最终导致了C++的出现。

归根到底，虽然C语言是业界最受欢迎、使用最广泛的专业编程语言，但是现在它却面临着其处理复杂性的能力小于实际需要的严重挑战。一旦程序达到了一定的规模，它就复杂得难以作为一个整体来管理。C++打破了这种障碍，帮助程序员理解并管理更为大型、更为复杂的程序。

1.1.3 C++的诞生

1979年，Bjarne Stroustrup在新泽西州Murray Hill的贝尔实验室中发明了C++。最初他将其称为“带类的C”。但在1983年，这种语言的名称改为C++。

Stroustrup将C++建立在C的基础上，C++包括C的所有功能、属性和优点。他仍然坚持“程序是由程序员来管理的，而非由语言控制的”这个C的基本设计理念。这里要理解的关键一点是，Stroustrup并没有创建一门全新的编程语言，而仅仅是改进了一门已经非常成功的语言。

向C中添加的大多数功能都用于支持面向对象编程。从本质上讲，C++是C的面向对象版本。Stroustrup将C++建立在C的基础之上，使得C向OOP的过渡变得非常平滑。无需学习全新的语言，C程序员只需要再掌握少量的功能即可享受面向对象编程带来的好处。

在创建C++之初，Stroustrup就知道保持C的原始风格是很重要的，包括它的高效、灵活性及其设计理念，在此同时再添加对面向对象的支持。令人高兴的是，他的目标完全实现了。C++除了支持面向对象的功能外，依然向程序员提供了C的灵活性与控制功能。

尽管C++最初的设计目标是帮助管理超大型的程序，但是它的作用并不局限于此。事实上，C++的面向对象功能实际上可以高效地运用在所有编程任务上。C++在诸如编辑器、数据库、个人文件系统、联网工具和通信程序这样的项目中的应用都不鲜见。因为C++共享了C的高效性，所以许多大型高性能系统软件都使用C++创建。同时，C++也是编写Windows程序时最常使用的语言。

1.2 C++的演化

自从C++发明之后，它经历了3次主要的修订，每一次修订都会对这种语言进行一些添加和修改。第一次修订在1985年，第二次在1990年。第三次修订发生在C++的标准化过程中。几年前，人们开始对C++进行标准化工作。为了实现这一目标，成立了由ANSI(American National Standards Institute，美国国家标准化协会)和ISO(International Standards Organization，国际标准化组织)参加的联合标准化委员会。推荐的第一稿标准产生于1994年1月25日。在这个草案中，ANSI/ISO C++委员会(作者也曾是其委员)保持了Stroustrup最初定义的功能，并添加了一些新的内容。但是，总的来说，这个最初的草案反映了当时C++的状况。

在C++标准的第一个草案完成不久，发生了一件事情，它导致了C++标准的大大扩展：这