国外计算机科学经典教材

Mc Graw Hill Education

# C++: A Beginner's Guide

## Second Edition

# C++ 基础教程

## (第2版)

(美) Herbert Schildt 著

王 军 译

清华大学出版社

国外计算机科学经典教材

# C++基础教程
# (第2版)

(美) Herbert Schildt 著

王 军 译

清华大学出版社

北 京

Herbert Schildt C++:A Beginner's Guide, Second Edition

EISBN：0-07-223215-3



北京市版权局著作权合同登记号　图字：01-2003-8454

**本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。**

**图书在版编目(CIP)数据**

C++基础教程：第 2 版/(美)斯柴德(Schildt, H.)著；王军译.—北京：清华大学出版社，2004

书名原文：C++：A Beginner's Guide, Second Edition

(国外计算机科学经典教材)

ISBN 7-302-08264-2

Ⅰ.C…　Ⅱ.①斯…②王…　Ⅲ.C 语言—程序设计—教材　Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2004)第 018509 号

**出 版 者**：清华大学出版社　　**地　　址**：北京清华大学学研大厦

http://www.tup.com.cn　　**邮　　编**：100084

**社 总 机**：010-62770175　　**客户服务**：010-62776969

**组稿编辑**：曹　康

**文稿编辑**：崔　伟

**封面设计**：康　博

**版式设计**：康　博

**印 刷 者**：北京市清华园胶印厂

**装 订 者**：三河市新茂装订有限公司

**发 行 者**：新华书店总店北京发行所

**开　　本**：185×260　**印张**：32　**字数**：806 千字

**版　　次**：2004 年 5 月第 1 版　2004 年 5 月第 1 次印刷

**书　　号**：ISBN 7-302-08264-2/TP·5960

**印　　数**：1～5000

**定　　价**：59.00 元

# 国外计算机科学经典教材

## 编审委员会

# 前　言

C++是一门非常优秀的编程语言，掌握它，可以开发高性能的软件。它所定义的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++是目前普遍使用的编程语言，几乎所有的专业程序员都可以使用这门语言编程。掌握 C++将为您完成编程任务打下坚实的基础。

本书讲授 C++编程的基础知识。它采用循序渐进的方式组织内容，并配以许多示例、自测题目和项目实例。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、特征和结构。通过学习本书，您将牢牢掌握 C++编程的基础知识。

在学习本书之前，首先对读者声明：本书介绍的只是利用 C++编程的入门知识。C++是一门庞大、复杂的语言，C++编程不仅仅涉及关键字、运算符和定义语言的语法，它还涉及如何使用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 C++库。在学完本书后，您将具备探索 C++库和其他领域的必要知识。

## 本书组织结构

本书的结构严谨，并且循序渐进地安排内容，每一部分的内容都以前面部分的内容为基础。全书共有 12 章，每章讨论 C++语言的一个方面。本书的独特之处在于，它使用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

每章的开头都列出本章要完成的学习目标。

### 本章自测

每章章末都有一个自测部分，检验您对知识的掌握程度。全部答案附在附录 C 中。

### 一分钟练习

在讲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

贯穿全书的“专家答疑”，是书中各相应主题的补充信息和重要评论。

### 项目

每章都包含一个或多个项目实例，演示如何应用所学的知识。您也可以将其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。不过大多数读者多少都有一点编程的经验。其中很多人都是从 Java 或 C#获得的。您将了解到，C++就是这两种语言的起源。因此，如果您对 Java 或 C#有所了解，就很容易学习 C++了。

## 需要的软件

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

## 免费代码下载

最后，告诉大家一个好消息：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
- 从键盘上读取输入
- 处理 if 和 for 语句
- 应用代码块
- 使用函数
- 了解 C++关键字
- 理解 C++标识符的规则

当前，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++是一种优秀的、可以开发高性能软件的编程语言。它的语法已经成为专业编程语言的标准，它的设计原理贯穿于整个计算领域。C++也是 Java 和 C#的始祖。总之，要想成为一名专业的编程人员，就必须精通 C++。它是通向现代编程领域的大门。

本章简要介绍 C++，包括它的发展历史，设计原理和一些最重要的特性。到目前为止，学习编程语言最困难的是：所有的组成部分都不是孤立存在的。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。为了克服这个困难，本章将简要介绍 C++的几个特性，包括 C++程序的一般格式，一些基本的控制语句和运算符。本章不深入介绍细节内容，而主要阐述所有 C++程序共有的一般性概念。

## 1.1 C++简史

C++的发展源于 C 语言，原因非常容易理解：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，支持面向对象编程(将在本章后面讲述)。同时 C++还对 C 语言的其他方面作了多处改进，包括对库例程的扩展。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 语言有较为深入的理解。

### 1.1.1　C 语言：现代编程的开端

C 语言的诞生开辟了一个崭新的编程时代。它的影响绝对不可低估，因为它从根本上改变了程序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流编程语言都产生了影响。在计算机领域，C 曾是主要的、具有革命性的语言。

C 是由 Dennis Ritchie 在一台使用 UNIX 操作系统的 DEC PDP-11 计算机上创造并首次实现的。C 是由一门更“古老”的语言 BCPL 发展演化而来的。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，它推动了 20 世纪 70 年代 C 语言的发展。

在 C 语言诞生之前，计算机编程语言的设计通常被视为一个学术活动，或者说它是由委员会的学究们开发的。而 C 不同，它是由真正的编程人员设计、实现和开发的，反映了他们处理编程任务的方式。C 的各种特性都由那些真正使用的人琢磨、测试和再三思考过。因此，C 吸引了大量支持者，并迅速成为软件界人员首选的编程语言。

C 语言诞生的源动力是 20 世纪 60 年代的结构化编程革命。在结构化编程出现之前，大型程序很难编写，因为程序逻辑总会退化为“意大利面条式的代码”，杂乱无序的跳转、调用和返回使人们很难理清其中的头绪。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，才可能编写比较大型的程序。

尽管当时还出现了其他的结构化语言，例如 Pascal，但是 C 首先成功实现了集功能强大、简洁易用和表达丰富等众多特征于一身。它的简洁、易于使用的语法与其基本设计理念直接相关——“程序是由程序员来管理的，而不是由语言控制的”。这让很多人迅速转向使用 C 语言。这种轰动效应在今天看来有些难于理解，但是 C 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

看了前面的叙述之后，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，程序的日益复杂要求人们用更好的方法来解决这些复杂性，于是，C++就应运而生了。为了更好地理解程序的日益复杂和计算机语言发展之间的相关性，可以考虑下列问题。

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的前面板上手动操作二进制机器指令来进行的。当时程序只有几百个指令，这种方法还可以奏效。随着程序的增长，出现了汇编语言，程序员使用机器指令的符号表示法，就可以处理更大型的、更复杂的程序。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问题。

实际上，第一种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，但它并不能编写出清晰、易于理解的程序。在 20 世纪 60 年代诞生了结构化的编程方法，它是类 C 语言所提倡的编程方法。结构化编程语言的出现，第一次实现了轻松编写中等复杂程度的程序。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。

为了解决这个问题，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++的诞生。

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，它就非常复杂，因而难以作为一个整体来管理。C++扫清了这种障碍，帮助程序员理解并管理更大型、更复杂的程序。

### 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在C的基础上创建了C++语言，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，而不是由语言控制的”这个C的基本设计理念。这对理解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，是至关重要的。

Stroustrup向C中添加的绝大多数功能都用于支持面向对象编程的。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。

在最初创建C++时，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，与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、网络工具和通信程序等项目中的应用很广泛。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程序最常用的语言。

## 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。推荐的第一个标准草案产生于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++第一个标准草案完成不久，发生了一件事情，它导致了C++标准的大大扩展：这就是Alexander Stepanov创建了标准模板库(Standard Template Library，STL)。STL是一组通用的