21世纪高等院校计算机基础系列教材

C++语言程序设计教程

C++ Program Design

宋斌 曾春平 朱小谷 等 编写

科学出版社
www.sciencep.com


C++语言
教程

21世纪高等院校计算机基础系列教材

程序设计教程

C++语言

C++ Program Design

宋 斌 曾春平 朱小谷 等 编 写

科学出版社
www.sciencep.com

内 容 简 介

Visual C++是基于 Windows 操作系统的编程工具。它将 Windows 的编程复杂性封装起来，使编程者可以比较轻松地进行 Windows 应用程序的设计。

本书共分为 10 章。第 1 章介绍了 C++的词法、语法规则和最简单的 C++程序以及如何用 Visual C++来进行开发。第 2 章介绍了 C++语言的基础（数据类型、程序流程控制、运算符和表达式以及函数的格式和调用方式）。第 3、4 章介绍了类和对象的基本概念及特性，包括对象的创建和销毁的机制、友元函数、友元类、嵌套类、对象数组、指针和引用等。第 5 章讨论了类的派生和继承性，继承增强了软件的可扩充性，并为代码重用提供了强有力的手段。第 6 章介绍了函数重载和运算符重载。第 7 章讨论了虚函数和多态性。第 8、9 章介绍了模板和错误处理。第 10 章介绍 C++系统的输出/输出流库，它使得程序员可以很容易的设计执行标准 I/O 和文件 I/O 的程序。书中所有的例子都在 Visual C++ 6.0 下编译运行通过。

本书适合于 C++语言的初学者和有一定编程经验的 C++程序员。

需要本书或技术支持的读者，请与北京中关村 083 信箱（邮编 100080）发行部联系，电话：010-82702660 010-82702658 010-62978181 转 103 或者 238，传真 010-82702698，E-mail：tbd@bhp.com.cn。

图书在版编目（CIP）数据

C++语言设计程序教程 / 宋斌编写. —北京：科学出版社，2005.6

ISBN 7-03-015294-8

Ⅰ. C… Ⅱ. 宋… 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 026903 号

责任编辑：刘 芯 / 责任校对：孙 红

责任印刷：东 升 / 封面设计：梁运丽

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

北京市东升印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005 年 6 月第 一 版 开本：787×1092 1/16

2005 年 6 月第一次印刷 印张：23 1/2

印数：1—4 000 字数：557 000

定价：34.00 元

21世纪高等院校计算机教材编委会名单

（排名不分先后）

主　任： 陈火旺 院士

副主任： 李仁发 教授　金茂忠 教授　陈　忠 教授　陆卫民 高工

委　员：

晏海华	教授	北京航空航天大学
邵秀丽	副教授	南开大学
刘振安	教授	中国科技大学
董玉德	副教授	合肥工业大学
倪志伟	教授	合肥工业大学
吕英华	教授	东北师范大学
杨喜权	副教授	东北师范大学
朱诗兵	副教授	装备指挥学院
樊秀梅	副教授	北京理工大学
徐　安	教授	上海同济大学
赵　欢	副教授	湖南大学
胡学钢	教授	合肥工业大学
林福宗	教授	清华大学
王家昕	教授	清华大学
郑　莉	教授	清华大学
朱淼良	教授	浙江大学
刁成嘉	副教授	南开大学
林和平	教授	东北师范大学
孙铁利	教授	东北师范大学
温子梅	讲师	广东教育学院
吕国英	副教授	山西大学
张广渊	讲师	沈阳大学
何新华	教授	装甲学院
邱仲潘	副教授	厦门大学
曾春平	副教授	第二航空学院
姬东耀	教授	中科院计算所
赵宏利	教授	装备指挥学院
喻　飞	博士	浙江大学
徐建华	总编	北京希望电子出版社
郑明红	副总编	北京希望电子出版社
韩素华	编室主任	北京希望电子出版社
张　曼	副教授	北京邮电大学
宋　斌	博士	北京邮电大学
朱小谷	高级编辑	北京希望电子出版社

总　序

21世纪挑战与机遇并存，没有足够的知识储备必将被时代所抛弃。中国IT教育产业竞争日趋激烈，用户需求凸显个性，行业发展更需要理性。未来五年IT行业将以每年18%的速度连续增长，将引发IT产业新的发展高潮。实现信息产业大国的目标，应该依赖教育，要圆信息产业强国的梦想，依然要寄托于教育，IT教育事业任重道远，其产业也正面临着机遇与挑战。

我国的计算机教学长久以来一直重原理、轻应用。高等院校的计算机教学机制和教材对计算机本身的认识都存在误区。要改革高校计算机教学，教材改革是重要方面，用计算机教材的改革促进基础教育的改革势在必行。

一本好书，是人生前进的阶梯；一套好教材，就是教学成功的保证。为缓解计算机技术飞速发展与计算机教材滞后落伍的矛盾，我们通过调查多所院校的师生，并多次研讨，根据读者认识规律，开创出一种全新的方式，打破过去介绍原理——理论推导——举例说明的模式，增加实用操作性，通过上机实验与课上内容结合来增强可读性，用通俗易懂的语言和例子说明复杂的概念。

本套教材的特点，一是“精”，精选教学内容；二是“新”，捕捉最新资讯；三是“特”，配备电子课件，力争达到基础性、先进性、全面性、典型性和可操作性的最大统一。

为保证教材质量，我们同时聘请了一批学术水平较高的知名专家、教授作为本套教材的主审和编委。全套教材包括必修课教材二十多种，选修课教材和学习配套用书十余种，基本上涵盖了目前高等院校（含高等职业技术学院、高等专科学校、成人高等学校）计算机科学与技术专业所必修或选修的内容。各种教材编写时既注意到内容上的连贯性，又保证了教学上的相对独立性。

本套教材在内容的组织上，大胆汲取当今计算机领域最新技术，摒弃了传统教材中陈旧过时的内容。这些变化在各本教材中都得到了不同程度的体现。本套教材编写时既参照了教育部有关计算机科学与技术专业的教学要求，又参考了“程序员考试大纲”和“全国计算机水平等级考试大纲”的内容，因此既适合作为高等学校计算机科学与技术专业教材，也可作为计算机等级考试学习用书。

考虑到各校教学特点和计算机设备条件，我们本着“学以致用”的理念，在本套教材编写中自始至终贯彻“由浅入深，理论联系实际”的原则，以阐明要义为主，辅之以必要的例题、习题和上机实习，能够使学生尽快领悟和掌握。

在本套教材编写过程中，作者们付出了艰辛的劳动，教材编委会的各位专家、教授进行了认真的审定和悉心的指导。书中参考、借鉴了国内外同类教材和专著，在此一并表示感谢。

我们希望更多的优秀教师参与到教材建设中来，真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议。

若有投稿或建议，请发至本丛书出版者电子邮件：textbook@bhp.com.cn

21世纪高等院校计算机教材编委会

前　言

C++是一种高效实用的计算机编程语言。它既可以用于面向过程的应用编程，也可以用于面向对象的系统开发，在计算机普及的现代社会得到了广泛的应用，成为了程序员首选的编程语言。

本书适合于C++语言的初学者和有一定编程经验的C++程序员。作者结合C++语言知识以及丰富的开发经验，通过大量的编程示例，力图使本书的读者通过学习，在掌握C++语言编程基本知识的同时，深刻领会面向对象设计的设计思想和方法。

全书共分为10章。第1章回顾了C++语言的发展历史，介绍了C++的词法、语法规则和最简单的C++程序以及如何用Visual C++来进行开发。第2章介绍了C++语言的基础，包括数据类型、程序流程控制、运算符和表达式以及函数的格式和调用方式等内容。第3章讨论了类和对象的基本概念，介绍了对象的创建和销毁的机制、友元函数、友元类和嵌套类等内容。第4章进一步介绍了类和对象的一些特性，包括对象数组、指针和引用等概念。第5章讨论了类的派生和继承性，继承增强了软件的可扩充性，并为代码重用提供了强有力的手段。第6章介绍了函数重载和运算符重载。第7章讨论了虚函数和多态性，多态性使程序员在设计程序时可以对问题进行更好的抽象，以设计出重用性和维护性俱佳的程序。第8章和第9章介绍了模板和错误处理。第10章介绍C++系统的输出/输出流库，它使得程序员可以很容易的设计执行标准I/O和文件I/O的程序。本书每章都有较多的程序示例，希望读者可以通过这些实例更好的理解书中的内容。书中所有的例子都在Visual C++ 6.0下编译运行通过。

本书作者以极大的热诚和细致，阐述了C++编程基础知识的方方面面，避免出现生涩技术词汇和知识断层，目的就是为了使得每一个有志于程序开发的初学者，或者正在学习C++编程的程序员能够系统的，最大限度的领会和掌握C++编程思想和工具。成为一个真正合格的C++编程高手，本书愿意是您的一本“开卷有益”的技术参考书！

在本书编写过程中，查阅了大量有关的中外文资料，现谨对这些书的作者提供的帮助表示最衷心的感谢。由于时间仓促，加之水平有限，对于书中所存在的错误和不足之处，敬请广大读者批评指正。

编　者

目　录

第1章 绪 论

目标

- C++语言的前身-C 语言
- C++语言的产生和发展
- 面向对象程序设计
- C++面向对象程序设计语言
- C++程序的编辑、编译和运行
- C++的词法及词法规则
- C++的程序结构
- 用 Visual C++创建控制台应用程序

C++语言是一种优秀的面向对象的程序设计语言，它是在 C 语言的基础上发展起来的一种新型的编程语言。作为 C 语言的超集，C++成功的实现了与 C 语言的全面兼容。与传统的 C 语言相比，C++不仅丰富和完善了 C 语言的功能，同时将面向对象的思想引入了程序设计，使其真正成为一种面向对象的程序设计（OOP）语言。C++语言通过类结构、数据封装、继承、重载和多态性等面向对象的特征，实现了软件的重用和程序的自动生成，从而为开发和维护大型复杂的应用程序提供了一种更有效更简洁的技术。

通过本章的学习，应该了解 C++语言的历史和概念，了解 C 语言与 C++语言之间的关系，了解简单的 C++程序的结构，能够开发最简单的 C++程序，同时要了解面向对象程序设计的一些基本概念。

1.1 C++语言的前身-C 语言

C 语言是在 1972 由 AT&T 贝尔实验室（Bell Laboratories）的 D.M.Ritchie 设计的一种程序设计语言。其前身是 B 语言，而 B 语言又是在 BCPL（Basic Combined Programming Language）语言的基础上发展出来的。C 语言既保持了 BCPL 和 B 语言精练、接近硬件的优点，同时也克服了它们过于简单、数据无类型等缺点。

最初的 C 语言只是为了描述和实现 UNIX 操作系统而提供的一种工作语言而设计的。在 1973 年，K.Thompson 和 D.M.Ritchie 两人合作将 UNIX 的 90%以上用 C 语言进行了改写（之前的 UNIX 是用汇编语言编写的）。1975 年，UNIX 第 6 版的公布，C 语言的突出优点引起了人们的普遍注意。随着 UNIX 的成功和广泛使用，也使 C 语言得到了迅速的推广，成为了一种普遍使用的程序设计语言。目前，在各种机型和各种操作系统上都有相应的 C 编译器。

1.1.1 C 语言是中级语言

C 语言通常被称为中级计算机语言。这并不意味着它的功能差和难以使用，或者比 BASIC、Pascal 那样的高级语言原始，也不意味着它象汇编语言那样，会给使用者带来类

似的麻烦。C 语言之所以被称为中级语言，是因为它把高级语言的成分同汇编语言的功能结合起来了。

表 1-1 列出了 C 语言在计算机语言中所处的地位：

表1-1 C语言在计算机语言中所处的位置

语言级别	语言
高级语言	Ada Modula-2 Pascal COBOL FORTRAN BASIC
中级语言	C++ C FORTH
低级语言	Macro-assembler Assembler

C 语言可以直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此 C 既具有高级语言的功能，又具有低级语言的许多功能，可以用来写系统软件。C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。因此，我们将 C 语言称为“中级语言”。

1.1.2 C 语言是结构化语言

C 语言是结构化程序设计语言。结构化程序设计要求程序的逻辑结构由顺序、选择和循环三种基本结构组成，C 语言提供了结构化控制语句（如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句等等）实现了这些功能。

结构化程序设计语言的另一个显著特征是程序和数据的分离。这种语言能够把执行某个特殊任务的指令和所有数据信息与程序的其他部分分离开并隐藏起来。获得隔离的一个方法是调用使用局部变量的子程序，通过使用局部变量，我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的 C 程序变得十分简单。如果开发了一些分离得很好的函数，在引用时仅需知道函数做什么，而不必知道它如何去做。切记，过度使用全局变量（可以被全部程序访问的变量）会由于意外的副作用而在程序中引入错误。使用过 BASIC 进行程序设计的人对这个问题都深有体会。

下面是一些结构化语言和非结构化语言的例子：

非结构化程序设计语言	结构化程序设计语言
FORTRAN	Pascal
BASLE	Ada
COBOL	C
	Modula-2

随着结构化程序设计语言的出现，人们发现它要比非结构化程序语言更易于设计和维护。C 程序由众多的函数组成，函数是进行模块化程序设计的基本单位。函数允许一个程序的不同任务被分别定义和编码，使程序模块化，具有良好设计风格的函数能保证它正确工作且不会对程序的其他部分产生副作用。能否在大型项目中设计出独立的函数是至关重

要的。在这种项目中，一个程序员的代码不应意外地影响其他人的程序。

在C语言中，实现结构化和代码隔离的另一种方法是使用复合语句。复合语句是程序单位，是一组在逻辑上相互关联的语句．在C语言中，复合语句就是处在左右花括号之间的一串语句。

1.1.3 C语言的特点

除了上面提到的C语言的两个特点之外，C语言还具有如下特点：

（1）语言简洁、紧凑，使用方便、灵活。C语言一共只有32个关键字，9种控制语句，程序书写形式自由，主要用小写字母表示。

（2）C语言提供了丰富的数据类型，包括了整形、实型、字符型、数组类型、指针类型、结构体类型、联合体类型等等。可以用来实现各种复杂的数据结构（如链表、树、栈等）的运算。

（3）C语言提供了丰富的运算符。C的运算符包含的范围非常广泛，共有34种运算符。C语言把括号、赋值、强制类型转换等都作为运算符进行处理，从而使C语言的运算符类型极其丰富，表达式类型多样化。

（4）语法限制不太严格，程序设计自由度大。

（5）生成目标代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低10~20%。

（6）用C语言写的程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

1.1.4 C语言的缺陷

当然，C语言也存在着很多缺陷：

（1）C语言的类型检查机制相对较弱，这使得程序中的一些错误不能在编译阶段由编译器检查出来，而这些错误遗留到程序的运行阶段由程序员来检查，将是非常困难的。

（2）C语言本身基本没有支持代码重用的语言结构，因此一个程序员即使有很高的程序设计技巧，并且严格遵循模块化程序设计方法，为一个应用程序所编写的程序代码也很难重用于另一个程序中。

（3）C语言不适合于开发大型程序，当程序的规模到达一定的程度时，程序员就很难控制程序的复杂性。

1.2 C++语言的产生和发展

C++语言是在基于C语言的语法基础上容入了Simula67、Algol68和Ada等语言中许多独特的语法特点之后而形成的全新的语言体系。本节主要讨论C++语言的产生和发展，以及实现标准化的过程。

1.2.1 带类的C

C++语言和C语言一样，都是诞生于著名的贝尔实验室。二十世纪八十年代初，为了

解决C语言中存在的问题，同时又要保持C语言的简洁、高效和接近汇编程序的特点，贝尔实验室的Bjarne Stroutstrup博士及其同事开始对C语言进行改进和扩充，他们称这种由C语言发展而来语言为“带类的C”。

带类的C顾名思义就是在C语言的基础之上扩充了类的机制。类是用户定义复杂数据对象的样板，是封装了数据和对数据操作的一种特殊的“数据类型”，可以按照用数据类型定义变量一样生成很多的对象。但是类毕竟不是一般的数据类型，它要复杂的多，因此在生成对象的时候就要有复杂的初始化工作。因此，在类中提供了用于构造对象的构造函数；当不再需要某个对象时，提供了析构函数来销毁它，复杂的对象对程序而言是它的行为而不是行为如何一步步实现的动作。因此，类样板设有控制可见性的接口，不在接口上的数据和操作（调用这些操作实现对象行为）对其他对象是不可见的。这就体现了封装性和局部性，为此扩充了公有私有控制，以及缓和封装友员类和减少冗余的派生类。当然派生类继承父类的机制完成了面向对象最重要的特征。为了充分体现抽象数据类型把“做什么”和“怎么做”分开，允许程序员在类中只写函数声明，函数体可以另写于类外。但为了运行效率，设置了内联函数，把这些分开写的函数定义和函数声明又连接起来。此外，类型检查和函数参数的类型转换、赋值运行符的重载都是在这一时期扩充的十种机制。

带类的C最初在DEC PDP/11机上实现使用，以后又移植到DEC VAX系统和Motorola 68000系列的机器上。它已应用于网络模拟器的研究之中。

1.2.2 从带类的C到C++

带类的C的原理探索取得非常好的效果，各种有C语言环境的用户竞相扩充带类的C，导致了这些软件无法相互移植。因此，AT&T公司就设想将带类的C作为一个新的标准语言，做一个工业化的C++版本，提供最低限额的标准的支持工具，并命名为C84语言。

C84编译器的前端称为Cfront，它对语言的语法和语义作完全的检查，建立输入的内部表示，产生代码所要的输出。在完成Cfront前端的工作中，全部吸收了带类的C的十种特征，同时又扩充了以下6种特征：

- 虚函数
- 函数名和运算符重载
- 引用机制/常量const
- 用户对自由存储的控制
- 改进了的类型检查
- 注释表示的多样化

由于C++和C各自目标不同，相互100%的兼容显然是不可能的，但商业和软件继承的需要只能采取折衷。由于C有几十个版本，ANSI标准化组织对其进行了标准化产生了ANSI C。C++和ANSI C做到基本兼容，ANSI C使C向C++更加接近。例如，函数调用一律做类型检查，因而设参数原型和缺省参数等机制。当然，C++和ANSI C依然存在“合理”的不兼容性。

1985年随着《C++程序设计语言（The C++ Programming Language）》一书的出版，C++1.0版定型。它不象其他语言先有语言规格说明，最后陆续验收合乎该规格说明的编译器，而是在已有编译器的原型上写出语言的定义。

1.2.3 C++2.0版

到1986年，C++的用户已经增加至两千人，作为一个原型语言它是成功的，于是如何开发一个更加高效的全新的编译器来代替Cfront1.0，以及如何完善支持面向对象的各种辅助机制，如流类库、各种部件库等等。这两个方面的工作同时在进行。

1989年AT&T推出了Cfront2.0，其新特征是：

多继承/抽象类/静态成员函数/const成员函数/protected成员函数/运算符的重载/成员函数指针/赋值和初始化的递归定义。

同时，对一些原有的特征又作了改进：

重载分辨/类型安全连接/用户自定义的内存管理机制。

Cfront2.0的效率和安全性都要比1.0版有显著的改进。C++的用户每7.5个月翻一倍。到1989年估记已经超过五万。但此时仍着眼于功能完善和安全可靠，开发各种机型上的C++编译器。1988年，Zortech在微机上推出C++编译器，次年出现了TURBO C++ 2.0编译器。紧跟着Borland公司在1990年5月推出了Borland C++系统。相比之下Microsoft的动作迟缓了一些，直到1992年才推出了基于DOS平台的MS—C/C++ 7.0系统。但由于其把握着PC操作系统的命脉，所以发展势头强劲，很快于1993年推出了面向Windows的Visual C++ 1.0系统，1994年推出了Visual C++ 1.5和可用于Windows 95和Windows NT系统平台的Visual C++ 2.0直至1998年推出Visual C++ 6.0。

1.2.4 C++3.0版

自C++2.0版后，2.1版扩充了迁移模块，以便向更高版本平滑过渡。

为完善类库提供重用支持，C++3.0版扩充了模板（template），即提供抽象类型的数据，可以实例化为所需要的类型，实现了真正嵌套的作用域，放弃了2.1版的扩充的迁移模块。改进了继承，解决了虚基类多继承的命名冲突/完善了构造函数和析构函数，完善的了成员和友元机制/完善了重载运算符/放宽内联限制/const限制/可对函数返回值优化等等。

从以上3.0版的改进可以看出，由于C语言过多依赖实现且不统一，造成了上层C++语法扩充后语义微妙的不同，致使完善和改进拖了很长时间，运用时要求对C++语言标准化。

1.2.5 C++的标准化

1989年后，随着若干独立开发的C++实现产品的出现和广泛应用，对C++实现标准化的需求越来越迫切。1989年由HP公司联合AT&T、DEC、IBM等公司发起建议标准化。为此，美国国家标准局ANSI（American National Standard Institute）成立了C++语言标准化小组X3J16，于1989年12月召开第一次会议。1991年6月国际标准化组织ISO（International Standards Organization）也为ISO C++成立了WG2委员会，第一次会议在瑞典召开，当然标准化组织是不会发布尚处于快速发展、不太成熟的版本。直到1998年，C++语言的国际标准才正式发布。

1.2.6 C++大事年表

在本节的最后，我们借用《C++语言的设计和演化》一书中的C++大事年表1-2，通过这个列表，可以对C++的产生和发展有一个概括的了解。

表1-2 C++语言大事年表

1979	3月	开始带类的C(C with Class)的工作
	10月	第一个带类的C实现投入使用
1980	4月	第一篇关于带类的C的贝尔实验室内部报告[Stroustup，1980]
1982	1月	第一篇关于带类的C的外部论文[Stroustrup，1982]
1983	8月	第一个C++实现投入使用
	12月	C++命名
1984	1月	第一本C++手册
1985	2月	第一次C++外部发布(Release E)
	10月	Cfront Release 1.0(第一个商业发布)
	10月	The C++ Programming Language[Stroustrup，1986]
1986	8月	有关“什么是”的文章[Stroustrup，1986]
	9月	第一次OOPSLA会议(OO的宣传开始时集中在Smalltalk)
	11月	第一个C++的商业移植(Cfront1-1GLockenspiel)
1987	2月	Cfront Release1.2
	11月	第一次USENIX C++会议(圣非，新墨西哥州)
	12月	第一个GNU C++发布(1-13)
1988	1月	第一个Oregon Software C++发布
	10月	第一次USENIX C++实现者工作会议(EstesPark，科罗拉多州)
1989	6月	Cfront Release2.0
	12月	ANSI X3J16组织会议(华盛顿特区)
1990	5月	第一个Borland C++发布
	3月	第一次ANSI X3J16技术会议(Somerset，新泽西州)
	7月	模板被接受(西雅图，华盛顿州)
	11月	异常被接受(PoloAlto，加利福尼亚州)
1991	6月	The C++ Programming Language(第2版)
	6月	第一次ISO WG21会议(Lund，瑞典)
	10月	Cfront Release3.0[包括模板]
1992	2月	第一个DEC C++发布(包括模板和异常)
	3月	第一个MicrosoftC++发布
	5月	第一个IBM C++发布(包括模板和异常)
1993	3月	运行时类型识别被接受(Portland，俄勒冈州)
	7月	名字空间被接受(慕尼黑，德国)