21世纪网络平台大学计算机系列教材

C/C++/Visual C++ 程序设计

方辉云 主编

科学出版社
www.sciencep.com

•21 世纪网络平台大学计算机系列教材

C/C++/Visual C++程序设计

方辉云　主编

科学出版社

北京

内 容 简 介

本书以属于标准C、C++语言版的Visual C++ 6.0系统为平台，全面介绍C、C++语言的基础理论和编程技巧。全书共分12章，包括概论，Visual C++ 6.0，程序设计基础，常量、变量与表达式，库函数与输入输出，程序的基本结构和语句，数组，函数和作用域，指针，构造数据类型，文件，带参数的主函数。书末还有5个附录。本书配有实践教材和一张多媒体课件及教案素材光盘，所列例题、习题均在Visual C++ 6.0系统和Turbo C系统下运行通过并有正确结果。

全书从基础到应用，突出重点，面向应用，叙述语言通俗易懂，循序渐进地讲授程序设计的方法和技术，适合高校理、工、经、管、农及相关专业本科生、部分研究生作为学习C语言的教材，也可作为网络学院、成教学院、高职高专院校相应的课程教材，还可供计算机水平考试、等级考试应试者及IT从业人员作培训和自学之用。

图书在版编目（CIP）数据

C/C++/Visual C++程序设计/方辉云主编. -北京：科学出版社，2005
（21世纪网络平台大学计算机系列教材）
ISBN 7-03-014833-9

Ⅰ.C… Ⅱ.方… Ⅲ.C语言-程序设计-高等学校-教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2004）第143034号

责任编辑：冯贵层 王雨舸
责任印制：高 嵘/封面设计：张 琴

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
武汉大学出版社印刷总厂印刷
科学出版社发行 各地新华书店经销
*
2005年1月第 一 版 开本：787×1092 1/16
2005年1月第一次印刷 印张：19
印数：1~8000 字数：472 000
定价：33.80元（含光盘）
（如有印装质量问题，我社负责调换）

前　言

C 语言是国内外广泛使用的计算机程序设计语言之一，也是国内外大学的重要基础课“高级语言程序设计”的首选授课内容。C 及 C 的发展产品包括 C++、Visual C++等，得到广泛的应用，是计算机应用人员应该掌握的一种程序设计语言工具。

C、C++和 VC++语言的特点，决定了学习和掌握该语言及其程序设计方法技术，是计算机学科各专业，以及所涉及到的各专业，特别是计算机科学与技术专业、信息管理与信息系统专业、信息与计算科学专业、电子商务专业以及统计专业、计量经济学专业等理工科专业的学生所必须掌握的专业基础知识技能。本书的内容，实际上是这些专业的专业基础课程。本书是在对上述各专业多次具体教学实用讲稿的基础上，不断修改完善完成的。

本书面向各高等院校本科生及研究生，也可作为管理类、经济类及其相关专业相应课程的教材和教学参考书，还可以供计算机工程技术人员、程序设计人员、IT 行业爱好者参考，亦可作为继续教育、网络教育、成人自修等相应专业或培训班的程序设计课程教材。本教材既适合初学者，也适合已经初步掌握了这方面知识的人员。

本书系统地、循序渐进地介绍了 C 及 C++的数据类型和运算符、语句格式和功能、结构化程序设计思想和方法。全书共分 12 章和 5 个附录。第一章是 C 和 C++概述。第二章介绍 Visual C++ V6.0 系统，这是学习 C 和 C++的平台，后面各章的全部例题、习题和上机实验等，均要借此完成。第三章介绍了结构化、模块化和面向对象、可视化等程序设计基础的概念。第四章介绍 C 语言的基础知识，包括基本字符集、基本词类、基本句类和基本的程序结构，以及基本数据类型、运算符、常量、变量和表达式。第五章介绍库函数及基本输入输出。在此基础上介绍程序设计方法和用 C 语言解决实际问题的算法，包括第六章的三大结构和语句，第七章关于数组和字符串知识，第八章函数编程和作用域，第九章指针的概念和使用指针的程序设计方法。第十章实际上是对数据类型的延续介绍，在已介绍的基本数据类型基础上，再介绍结构、联合、枚举等构造型数据类型，以及它们的特点和在程序设计中的使用。第十一章介绍在 C 语言下的文件处理操作和编程方法。第十二章介绍带参主函数的程序设计和编译预处理知识。

全书根据教学特点精心安排了例题和习题。每章都精选了丰富的例题来验证语法，描述算法，说明程序设计方法。这些例题全部在 Turbo C 和 Visual C++系统上运行通过并有正确的结果。各章后面都配有相应的习题，这些习题在本书配套的实践教程中给出参考答案，供读者在学习中作自我测验用。答案涉及程序时，无论是编程题，还是阅读程序写结果或程序填空等，均在机器上具体实现过。

书尾的附录包含了丰富的与本语言有关的学习中需要的内容，供读者在学习过程中查阅。

全书的知识结构如图 1 所示。

为方便教学和阅读，本书还配套有电子版(课件素材)和实践教材。

本书由方辉云主编，何友鸣撰写了其中的第二章和部分附录。在编写和出版过程中，作者在对内容的取舍、章节的安排、结构的设计以及诸如表达式之类的重要基础知识方面，都多方听取了意见，并进行了反复的修改。这里，我们要衷心感谢中南财经政法大学信息学院

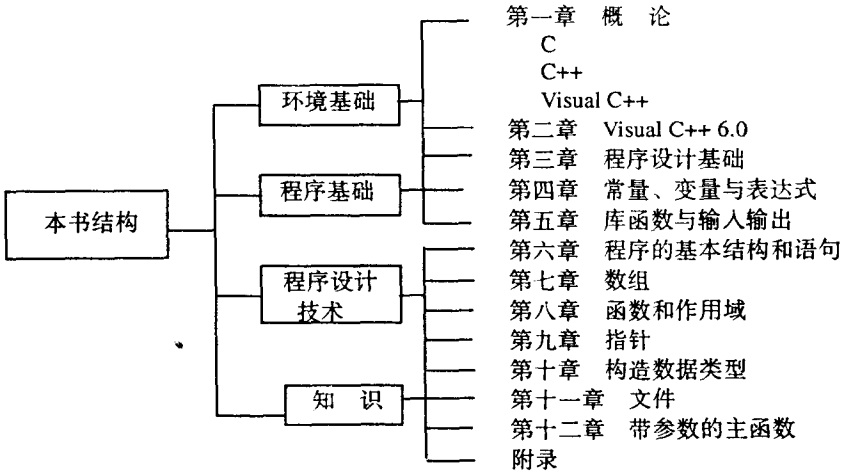

图 1 本书的知识结构

院长、博士生导师杨云彦教授的深切教诲，副院长刘腾红教授对本书自始至终的关怀，以及信息管理系主任宋克振教授的指导。衷心感谢武汉大学商学院副院长、博士生导师童光荣教授和武汉理工大学信息工程学院副院长、博士生导师钟珞教授对本书的审阅！衷心感谢中国地质大学(武汉)金连昆教授、武汉大学数学与统计学院梅献民主任、湖北大学数学与计算机学院李跃新教授、武汉科技大学文法与经济学院黄春荣教授以及武汉化工学院(武汉工程大学)、湖北工业大学、华中理工大学、空军雷达学院、武汉工业学院、武汉科技学院等院校的领导、学者和同仁们对本书的支持、肯定，以及直接的帮助和中肯的、建设性的意见。何苗、郭凯红、夏鹏、明喆等做了很多工作，一并表示感谢！写书是很容易“一叶蔽目，不见泰山”的，那些站在幕后评读全稿的人的指导的重要性，是无法估价的！

由于作者水平有限，时间仓促，不足和疏漏之处在所难免。在此衷心希望采用本书作教材的教师、学生和读者提出意见和建议；竭诚希望得到计算机教育界、程序设计方面的同仁批评指正；祈望得到专家们的不吝赐教，以便我们在再版时进行修订和补充。

方辉云

2005 年 1 月

目　录

第一章 概 论

1.1 C 语言的发展

20 世纪 60 年代，随着计算机科学的迅猛发展，高级程序设计语言得到了广泛的应用，然而，当时尚没有一种用于书写系统软件如操作系统、编译系统等的高级语言。人们对任何系统软件，都不得不用汇编语言来编写。

用汇编语言编写程序存在可读性差、不可移植等缺点，描述问题的效率不如高级语言高，编写程序有很多不便，于是人们开始开发能用于设计系统软件的高级语言。

当时流行的高级语言之一的 Algol 60，结构严谨，注重语法和分程序结构，因此它对后来许多重要的程序设计语言如 Pascal、PL/1、SIMULA 69 等，都产生过重要的影响。但由于它是面向过程的语言，与计算机硬件相距甚远，所以不适合编写系统软件。

1963 年，英国剑桥大学在 Algol 60 的基础上，推出能接近硬件的 CPL 语言，但 CPL 语言太复杂，难于实现。

1967 年，英国剑桥大学的 Martin Richards 对 CPL 语言作了简化，设计并实现了 BCPL(basic combined programming language)语言。

1970 年，美国贝尔实验室 Ken Thompson 以 BCPL 语言为基础，设计了更简单也更接近硬件的一种语言，该语言取 BCPL 的第一个字母 B，即 B 语言。

1971 年，Ken Thompson 在 PDP-11/20 计算机上实现了 B 语言，并用该语言书写了系统软件——Unix 操作系统和一些应用程序。B 语言是一种解释性语言，功能也不够强大。比如说，由于 B 语言是一种无类型语言，因而无法支持多种数据类型，这在描述许多事物时，将会遇到困难。

为了很好地适应系统程序设计的要求，在 1972~1973 年间，贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上再发展，重新设计了一种新的通用程序设计语言，取 BCPL 的第二个字母 C，称为 C 语言。现在人们一般都认为 C 语言是美国贝尔实验室的 D. M. Ritchie 在 1972 年设计实现的。C 语言既保持了 BCPL 和 B 语言的精练及接近硬件等优点，又克服了它们过于简单、数据无类型等缺点，是一种较完好的计算机语言。

1973 年，贝尔实验室的 Ken Thompson 和 Dennis M. Ritchie 用 C 语言重写了 Unix 操作系统，并在 PDP-11 机上实现，这就是著名的 Unix Version 5，是当年世界上最著名的分时操作系统之一。该版本的 Unix 加进了多道程序设计功能，为 Unix 的发展奠定了基础。随后出现的 V6、V7 以及 SYSTEM III、IV 等，使 Unix 逐渐成为最重要的操作系统之一。

当初设计 C 语言的目的是为描述和实现 Unix 操作系统提供一种高级语言工具，但由于 C 语言本身并没有被束缚在任何特定的操作系统和硬件之上，所以它具有良好的可移植性。

1977 年出现了不依赖于具体机器的 C 语言编译文本，各种机器间的移植变得更加简单。这时，用 C 语言编写的程序，可以不加修改或稍加修改，就可从一个环境搬到另一个环境中使用。C 语言的这种特性推动了其在 Unix 的广泛实现，而随着 Unix 的日益普及，又带动了

C 语言的迅速推广。今天，各种大中小型计算机及微型计算机上都可以装配 C 语言系统，无论是装配 Unix 操作系统还是装配非 Unix 操作系统的机器，都能实现 C 编译程序。

1978 年，贝尔实验室的 Brian W. Kernighan 和 Dennis. M. Ritchie(即后来人们所说的 K&R)合写了一部名为 *The C Programming Language* 的书，并在该书的附录中列出了 C 语言参考手册。之后，这本书成为人们广泛使用 C 语言的基础，被称为非官方的 C 语言标准。

1983 年，美国国家标准化学会 ANSI(America National Standards Institute)制定了 ANSI C 国家标准。

1987 年，ANSI 又公布了新的一个标准，称为 87 ANSI C。

1990 年，87 ANSI C 获国际标准化组织 ISO(International Standard Organization)通过，成为 ISO 标准语言，即 ISO 9899-1990。

目前流行的 ANSI C99 是一个向下兼容的 C 语言新标准。

20 世纪 80 年代以后，微机的普及使 C 语言更加普及。C 语言的使用几乎覆盖了计算机的所有领域，包括操作系统、编译系统、数据库系统、CAD(computer-aided design，计算机辅助设计)、过程控制、图形图像处理等，程序员、非程序员计算机工作者以及非计算机专业人员都喜欢它。

现在广泛流行的各种版本 C 语言编译系统都以 ANSI C 为基础，其基本部分是相同的，也存在一些不同的地方。微机上使用的如 MC(Microsoft C)、TC(Turbo C)、QC(Quick C)、BC(Borland C)及 VC(Visual C)等等，都各有其特点，后面我们将介绍。

在我国，从 20 世纪 80 年代开始流行使用 C 语言。开始是 TC 和 BC 相应普及一些，现在是 VC++。今天，C 语言已经成为我国最热门的程序设计语言之一。

1.2 C 语言的特点

C 语言是高级程序设计语言中的一种“低级”语言，它与系统和机器的紧密程度使其几乎可以替代汇编语言。通常只有汇编语言才具备的功能，C 语言中都有，如位操作、直接访问物理地址等，这使得 C 语言在进行系统操作程序设计时非常有效。而在这之前，系统软件只能用汇编语言来编写。C 语言的出现和日趋成熟，使汇编语言的许多传统领地被 C 语言所取代，使程序员的负担得以减轻，效率得以提高。

C 语言的这种既具有高级语言的功能、又具有低级语言诸多功能的双重性，使它既是成功的系统描述语言，又是通用的程序设计语言！这就是人们常说的“高级语言中的低级语言”或“中级语言”的意义之所在。在系统软件方面，除 Unix 外，高性能的关系数据库管理系统 ORACLE 也是用 C 语言编写的。

1. C 语言简洁、灵活

与其他高级语言不同的是，C 语言还可考虑背景机的特点，故编写的程序及生成的目标代码质量高，甚至可以与汇编语言相媲美。其他高级语言相对于汇编语言的代码效率而言，要低得多。就 IBM-PC 机上的 C 语言系统 C86 而言，其代码效率只比汇编语言低 10%~20%，但 C 语言在描述问题时表现出来的编程迅速、表达能力强等优点，又是汇编语言无法比拟的。

另一方面，C 语言优秀的控制流和数据结构、丰富的数据类型和运算符，又使它在数值计算方面也显示出相当的优势。C 语言共有 34 种运算符(见附录 E)，它把括号、赋值、强制类型转换等都作为运算符来处理，从而使 C 的运算类型极其丰富，表达式类型多样化。

此外，C 语言书写风格独特，表达形式也比较精练，它可以直接处理字符、数字、地址等，可以完成通常要用硬件才能实现的普通的算术及逻辑运算，且可以表达数值处理、字处理功能；对语法限制也比较宽松，例如，C 语言对数组下标越界不作检查，不专门设置逻辑型数据而以整型代替，整型和字符型数据可以通用等等。

我们知道，系统较少的限制可以给程序员带来较大的自由，但在编程时必须先考虑好，明确自己在做什么，而不要把检查错误的工作仅仅寄托于编译程序。

C 语言是一种模块化的程序设计语言，完整的 C 语言程序通常由若干个函数组成，其结构严谨，清晰紧凑。一个大程序可以由若干个较小的程序模块组成。

可见，C 语言较好地处理了简洁性与实用性、可移植性与高效性之间的矛盾。如果说 Fortran 语言是面向工程师的，Cobol 语言是面向商务管理的，Pascal 语言是面向学生的，Basic 是面向非程序员的，那么 C 语言则是面向专业程序设计的。C 语言的设计者，美国贝尔实验室的 Dennis M. Ritchie 就是一个专业程序员，而他最初是为了自己写 Unix 操作系统而设计 C 语言的。C 语言实现了程序员的期望：很少限制、很少强求，程序设计自由度大，方便的控制结构，独立的函数，紧凑的关键字集合。这些特点使 C 语言有较高的执行效率，受到专业程序员的青睐，这可能是 C 语言被广泛应用的主要和直接原因。实际上，现在各行各业的专业程序员都普遍使用 C 语言编程。

2. C 语言是典型的结构化语言

结构化语言编写结构化程序，一个结构化程序就是用高级语言表示的结构化算法。结构化算法用三种基本结构完成，即我们在后面第三章和第六章要讲到的顺序结构、选择结构和循环结构。用三种基本结构组成的程序必然是结构化的程序，这种程序便于编写、阅读、修改和维护，能提高程序的可靠性，保证程序的质量。

结构化程序设计强调程序设计风格和程序结构的规范化。结构化程序设计方法的基本思路是，把一个复杂问题的求解过程分阶段进行，每个阶段处理的问题都控制在人们容易理解和处理的范围内。其具体做法是：自顶向下，逐步求精，模块化设计和结构化编码。

世界上第一个结构化语言是 Pascal 语言，因而大量用于教学之中。但 Pascal 语言难以推广到各种实际应用领域，到目前为止基本上只是一种教学语言。C 语言是理想的结构化语言，且描述能力强，同样适宜于教学，而且有广泛的应用领域。Pascal 等高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性，C 语言则强调灵活性，使程序设计人员有较大的自由度，以适应宽广的应用面。

C 语言具有结构化的控制语句，用函数作为程序的模块单位，便于实现程序的模块化。可以说，C 语言是理想的结构化语言，符合现代编程风格的要求。

C 语言也存在一些缺点和不足。例如，它的某些运算符的优先顺序不符合人们的日常习惯；复合语句在多层嵌套时，语句括号匹配情况不够醒目；C 语言的弱类型转换具有潜在的不安全因素等等。这些都警示我们：用 C 语言编写程序，切忌粗枝大叶，必须要有严谨的态度，不然会受到系统的“惩罚”。

1.3 C 和 C++

从 C 语言出现到今天，版本各式各样，但核心部分编程技术和整体结构特点是相同的，不同的是有扩充部分和功能增减部分，有的则仅仅是上机操作步骤不同、用户界面不同等。

可以这样认为：

初始阶段，针对 Unix 操作系统，在中、小型机上应用的 C 语言是一般的 C 语言，对此，我们在这里就不介绍了。

20 世纪 80 年代，微型计算机开始普及应用，其主流操作系统是 DOS(disk operating system)。在这一段时间内，有很多公司为 IBM-PC 机开发和配置 DOS 下的 C 语言系统，版本也很多，如 C-SYSTEMS C、Desmet C、Intel Associates C88、CI-C86、Microsoft C、Super Soft C 等等。其中的典型代表是美国的 CI(Computer Innovations)公司的优化 C 语言系统——CI-C86，它同时有汉化版。CI-C86 的典型特点是，产生的目标程序采用 Microsoft(微软)公司的.OBJ 文件的兼容格式，可直接用标准的 Microsoft 公司的宏汇编 MASM 进行汇编。基于 DOS 的 Turbo C 通常称为 TC，使用的普及性很高。

美国 AT&T 贝尔实验室的 Bjarne Stroustrup 在 20 世纪 80 年代初设计并实现了 C++。它是 C 语言的扩充，其主要扩充功能为：①支持数据抽象；②支持面向对象的设计及编程；③改进了 C 语言中的若干不足之处。

在作扩充和改进时，C++保留了传统的和有效的结构化语言的特征，即 C 的简洁、高效性，同时融合了面向对象的能力。

C++语言从 20 世纪 80 年代初期开始开发，在 20 年左右的时间里，随着人们对有关方面认识的发展以及使用 C++的经验积累，使得本语言也在不断发展。这方面的详情可参看 Stroustrup 的著作 *The Design and Evolution of C++*。

之后，C++越来越受到重视并被广泛使用。随着 C++语言使用的飞速扩张，人们开发出很多独立的 C++语言系统，许多软件公司争相为 C++设计编译系统，如 AT&T、Apple、Sun、Software、Borland 等等，这些系统之间也逐渐出现了一些差异。为了能使 C++保持为一种统一的、通用的并不依赖于具体实现的语言，1990 年开始了对 C++语言的标准化工作，这一工作一直持续到 1998 年 C++的标准化文本通过。在这个发展过程中，人们对 C++语言做了一些整理，也逐渐增加了一些功能。本书尽力使用标准化文本所定义的语言，所有程序尽力基于标准化语言书写。

学习 C++语言及其程序设计时，少不了要有 C++系统的支持。目前可用的 C++系统很多，但有一些老系统是多年之前开发的，缺乏标准 C++语言的许多威力强大的功能。因此，要注意尽可能地选用符合标准 C++语言的系统。在本章后面将要介绍几个好的系统，其中包括 Microsoft 的 VC++ 6.0。

在我国，Borland C++比较流行。随着面向对象技术的日益成熟，面向对象的程序设计语言也越发完善。国际上流行的面向对象语言主要有：Smalltalk、CLOS、Object Pascal、Borland C++等，使用得最多的要算 Smalltalk 及 C++。C++是在 C 语言中引入 Smalltalk 风格，C 则是在 Lisp 语言的基础上开发出来的。Lisp 是一种智能语言。

什么是面向对象？客观世界中任何一个事物都可以看成一个对象，这样，我们可以说，客观世界是由千千万万个对象组成的，它们之间用一定的方法相互联系。比如说一个系是一个对象，一个班级也是一个对象。实际生活中，人们往往在一个对象中进行活动，比如，学生在某个班中。从这点上说，对象是进行活动的基本单位。

从理论上讲，一个对象应该包括两个要素：数据和需要进行的操作。这里，数据相当于某个班中的学生，需要进行的操作则相当于学生进行的活动。对象就是包含数据以及与这些数据有关的操作的集合。

前面我们讲到，传统的、面向过程的程序设计是围绕功能进行的。在C语言中，用一个函数实现一个功能。所有的数据都是公用的，一个函数可以使用任意一组数据；反之，一组数据能被多个函数所使用。在这种情况下程序设计者必须考虑每一个细节，什么时候对什么数据进行操作。当程序规模较大、数据很多、操作繁杂时，程序设计人员往往感到难以应付，如同由一位校长管理每一个学生一样，忙得不可开交而又错漏百出。

现在，面向对象的设计丢开这些直接的琐事，它面对的是一个个对象，所有的数据分别属于不同的对象。每一组数据都是有特定用途的，是某种操作的对象。把相关的数据和操作放在一起，形成一个整体，与外界相对分隔开来，这犹如一个班的学生在一起，与外班相对独立一样。这是符合客观世界的本来面目的。校长只把教学任务分派给院系，而不必指派张老师教什么、李老师教什么，具体的分配由院系完成。对于校长，院系是一个“黑箱”，它如何运作，外界人士可以不必详细了解，只要给它一个任务，它就能按规定完成。这就是理论上所说的把对象“封装”起来，各自相对独立，互不干扰。

面向对象的程序设计方法的一个重要特点就是“封装性”，把数据和数据的操作码封装在一个对象中。在这种情况下程序设计者有两方面的任务：一是设计对象，即决定把哪些数据和操作封装在一起；二是如何通知有关对象完成所需的任务。程序员这时就像一个总调度，不断向各个对象发出命令，让各个对象完成各自的操作。各个对象的操作完成了，整个任务也就完成了。

由此可见，对于一个大型的任务，面向对象的程序设计方法能大大降低程序设计人员的工作难度，减少出错机会。

面向对象的设计方法是近年来在软件工程研究领域中有益成果的基础上发展起来的一种软件设计方法。它集抽象性、封闭性、继承性、多态性于一体，将传统软件中的数据和操作组合成易于赋予语义的对象，使软件设计中人们普遍遵循的模块化、信息隐藏、抽象、代码共享等思想，在面向对象语言机制的帮助下得以充分实现。

C++在C语言的基础上增加了数据抽象、继承、多态以及其他一些改善C语言程序设计结构的机制，既为程序员提供了面向对象的能力，也未降低运行时间和空间效率，是一种灵活、高效、可移植的、面向对象的语言。

下面简单介绍一下C++系统。学习C++语言程序设计，当然需要有一个C++系统作程序练习。C++标准系统是1998年通过的，在此之前，有许多C++系统及其语言，它们都与标准存在一些差异：较早的C++系统可能不支持模板机制，或者不支持异常处理，特别是缺乏C++标准所定义的那样一个功能强大的库。C++语言标准化过程中变动最大的就是标准库，而标准模板库的基本框架大约到1994年底才被接受，早于此时的C++系统都没有提供有关功能。使用与标准C++不符的C++系统，也会给学习C++语言带来困难。各种C++系统，如Turbo C++、Borland C++、Bloodshed Dev-C++、Visual C++等，都在不断发展改造，向C++标准靠拢。

下面简单列出几个主要的C++系统。

1. Borland C++

Borland C++简称BC++，由Borland公司开发，是一个很完善的C++系统。其命令行版本免费提供给社会使用。所谓命令行版本，即不包含集成开发环境，要使用它只能通过下达逐条命令来完成。如果我们在此系统的基础上再配用一个适合处理C++语法要求的、有强大功能的程序编辑器(实际上这样的编辑器目前流行的有许多)，就可以很方便地学习C++了。

使用该系统之前应设置好若干环境变量，有关情况可阅读系统所带的 Readme 文件。

2. Bloodshed Dev-C++

Bloodshed Dev-C++简称 Dev-C++ 4.0，是一个自由软件，由 Bloodshed Software 开发，功能全面且自身较小，安装后约占 20M，对系统要求不高，支持 Windows 下的应用程序开发。本系统可以在网站 http://www.bloodshed.net/下载。由于它是一个自由软件，所以存在自由软件的通病：帮助信息过于简单，没有内容详尽的文档和手册。

这个编译系统原本是为在 Unix 上运行而开发的，所以有些操作，如文件结束命令是 Ctrl+B，而不像一般 DOS/Windows 下用 Ctrl+Z 那样普及。这个系统中没有<sstream>头文件及其支持的字符流功能，所提供的是一个较老的<strstream>头文件。

3. Visual C++

我们推荐使用的是 Microsoft 公司开发的 Visual C++ V 6.0(简称 VC 6.0)，这是一个功能强大的 C/C++系统，它同时支持 C 和 C++语言程序设计，并为开发 Windows 下的程序提供了非常庞大的扩充库。为了今后的 Windows 编程，我们一定要熟悉 VC 6.0 的使用。

VC 能支持本课程的教学和实习，但在 VC 系统里，每一个程序都必须建立一个工作项目即工程(project)。如果编写的程序并未放入一个 project 中，在编译加工时系统会自动创建一个默认的 project。例如，在 TC 2.0 下的程序

```
main( )
{
   int a,b,sum;
   a=10;b=20;sum=a+b;
   printf("sum is %d\n",sum);
}
```

在 VC++ 6.0 之下则要写为

```
#include <stdio.h>
void main( )
{
   int a,b,sum;
   a=10;b=20;sum=a+b;
   printf("sum is %d\n",sum);
}
```

等等这些都说明 VC++系统会给初学者编写小程序和理解开发环境造成暂时的小困难和不便，但它的正规性好、标准化程度高。

在这里要求大家能够掌握和使用 Visual C++ 6.0，具体内容我们在下一章介绍。

有关 VC 系统的书籍在市面上很多，大致可分为两大类：一类是基于 VC 系统学习 C 和 C++语言程序设计，这是我们当前要用的；另一类则是讨论如何用 VC 开发 Windows 系统上的应用程序，这是本课程的后续课程“Windows 编程”要解决的问题。

4. C#

C#(C sharp)语言是由 Microsoft 公司在 2000 年 6 月 26 日正式对外发布、作为.NET 平台开发的语言，它是一种全新的、面向对象的、现代的编程语言。它同样来源于 C\C++。

C#的集成开发环境 Visual Studio.NET 提供了类似于 Visual Basic 的开发环境，称为 Visual

C#。C#与 Java 的语言规范都是从 C++继承改造而来，有人称 C#是 C/C++家族中的第一种面向组件的语言。

Turbo 系列的 Turbo C++对我国的软件事业也有深远的影响。Turbo 版本的 C 语言系统是在微机上、DOS 环境之下使用的，其上机操作方法将在本书附录 B 中介绍，仅仅是介绍，不作要求，因为它们是基于 DOS 的，我们要求掌握的则是基于 Windows 的 Visual C++ 6.0 的操作使用技术。

1.4 Visual C++

在 C 和 C++基础上发展起来的 Visual C++是可视化的系统，属于 Windows 编程技术，它在面向对象方面发展得更好。

随着视窗平台(Windows)的出现和普及，微软公司的 C++ 7.0 版提供了 MFC(Microsoft Foundation Class)库，即微软基础类库，它为人们提供了一条开发 Windows 应用程序的捷径。而当人们正在回味这种绝妙之处时，Microsoft 公司又不失时机地推出了适应于 Windows 和 Windows NT 平台上的 Visual 系列产品 Visual studio。Visual studio 也称作 Developer studio，或是 IDE(集成开发环境)，即可视化程序设计软件，包括 Visual Basic、Visual Pascal、Visual J++、Visual FoxPro、Visual Dev 以及我们在这里和下一章要介绍的 Visual C++等等。

新一代的 Visual 系列软件正在以不可阻挡和不可抗拒之势取代以前的各种 C，这是当今开发工具的方向。所谓可视化(Visual)程序设计，是指一种开发图形界面(GUI)方法。用这种方法，编程人员不必编写大量代码去描述界面元素的外观和位置，只要把预先建立的界面元素如按钮、列表框、对话框等 Windows 风格的部件，用鼠标拖放到屏幕上适当的位置即可。这种快速创建用户界面的方法技巧，如同在 Windows 中使用诸如画笔之类的绘图程序那样轻松。

Visual C++是使用 C 语言进行可视化程序设计的开发工具，是在原有 C++语言的基础上结合 Windows GUI 进一步发展，其界面友好、简洁、易用，它不仅在开发一般的 Windows 应用程序上十分方便，而且在进行多媒体(multimedia)、互联网(Internet)和数据库(data base)等应用程序的开发上，同样得心应手。

Visual C++ 6.0 系统是当今应用十分广泛的开发软件，它兼容 C 及 C++的程序处理。我们在后面深入讲解时，所有的程序及例题都在 VC 6.0 平台上实习。

1.5 小 结

在 C 语言家族中，C 是结构化和模块化的语言，它是面向过程的。所谓面向过程，暂时可以大体上理解为：把重点放在对具体问题解决方法的详细描述上，是围绕功能进行的。程序设计人员在这种情况下必须细致地设计程序中的每一个细节，准确地考虑到程序运行中的每一时刻会发生的事情，例如：各个变量的值如何变化、什么时候要进行哪些输入、程序什么时候在屏幕上会输出什么等等，这对程序员的要求较高，甚至是苛刻的！这种工作方式在处理较小规模的问题时，会使编程人员感到得心应手。但当实际问题比较复杂、程序的规模当然也就大了时，程序员还用这种方法解决问题就会感到力不从心。这时，结构化程序设计方法会显露出它的不足。

结构化程序设计方法的提出在当初是为了解决软件设计危机的问题，但可惜这个方法没有有效地解决软件危机，于是人们继续寻求解决方法。20 世纪 80 年代，OOP(object-oriented programming)即面向对象的程序设计方法被提出，这时，C++应运而生。C++保留了 C 的所有优点，增加了面向对象机制，是 C 的超集，与 C 完全兼容，用 C 编写的程序可不加修改地用于 C++。

C++既可用于结构化程序设计，又可用于面向对象的程序设计，是一个功能强大的混合型程序设计语言。C++不是简单地对 C 做了某些改进，而是在 C 的基础上成功进行了一场革命，赋予了 C 以新的生命力。

处理 C++源程序，必须事先安装 C++编译系统，简称 C++系统。基于操作系统 DOS 平台的有 Turbo C++和 Borland C++等。我们学习 C 语言，既可以在 Turbo C 系统下进行源程序处理，也可以在 Turbo C++系统下处理。

C 的源程序文件的后缀为 C，而 C++的源程序文件的后缀为 CPP，即 C Plus Plus 的缩写，C++的表示。例如，Borland C++开发环境中，实际有两个编译系统，根据被处理的源程序文件名的后缀是 C 还是 CPP 来自动调用其中的一个编译系统来支持工作。

同 Turbo C 一样，基于 DOS 的 C++编译系统我们不要求大家掌握，而要求大家使用和掌握的是基于 Windows 的 Visual C++系统。

有一点要特别说明的是，C++是一种大型的计算机程序设计语言，其概念、功能和语法规定都比较复杂，要深入掌握 C++需要花较多较长的时间，尤其是需要有较丰富的实践经验。面向对象的程序设计方法主要是解决大型软件的设计问题，只有编写或曾经编写过大型程序的人，才会体会 C 的不足和 C++的优点。所以实际上使用 C++编写程序的，主要是软件专业的从业人员，而高校里的程序设计或高级语言程序设计的课程的任务，则主要是进行程序设计的基本训练。因此，对于当前的大学生而言，应先掌握好 C 语言程序设计！有了 C 语言的基础，需要时再系统学习 C++是较容易的。我们在这里重点介绍 C 语言，同时在基础知识方面也介绍相应的 C++知识，使同学们对 C++有一个印象和了解，为今后必要时进行系统学习 C++打下基础。

习　题　一

1. 什么是结构化程序设计方法?
2. 什么是面向对象程序设计方法?
3. C 语言是在什么语言的基础上发展起来的？当初设计它的目的是什么?
4. 与汇编语言相比，C 语言有哪些特点?
5. C++对 C 语言做了哪些修改?
6. 什么是可视化(Visual)程序设计方法?

第二章　Visual C++ 6.0

2.1 概　　述

第一章我们曾讲过面向对象的程序设计方法。其实，“面向对象”是软件界的一种潮流。面向对象的程序设计 OOP 实际上是一种观念，用什么语言实现它都可以。当然，面向对象的程序设计语言 OOPL(object-oriented programming language)是专门为面向对象观念而发展起来的，能更方便地实现面向对象的封装、继承、多态等特性。

C++语言虽然不像 Small Talk 和 Java 那样是一种纯粹的面向对象的语言，但由于它站在 C 语言的“肩膀”上，因而成为了最重要的面向对象的程序设计语言。

面向对象的设计与面向过程的设计是有很大区别的，甚至可以说这是一种思维模式的转变。这就是为什么有的人使用 C++语言，仍然编不出面向对象的程序的原因。学习 C++语言不仅要懂得其语法(这可以通过学习任何一本 C++的教材或专著来达到)，更要懂得其语意，尤其要认识 C++语言面向对象的特性和实现面向对象的方法。

我们知道，Visual C++是可视的、基于 Windows 的面向对象的程序设计语言，Windows 的最初开发是在 20 世纪 80 年代早期，在 C++出现之前，但是当时已经提出了面向对象程序设计思想。Windows 的开发者们有意识地将界面上的不同元素当作对象来处理，在 Windows 的界面设计和软件开发环境中，处处贯穿着面向对象的思想。

另外，在 Windows 中，程序的基本单位不是过程和函数，而是窗口。一个窗口是一组数据的集合和处理这些数据的方法和窗口函数。从面向对象的角度来看，窗口本身就是一个对象。Windows 程序的执行过程本身就是窗口和其他对象的创建、处理和消亡过程。Windows 中的消息发送可以理解为一个窗口对象向别的窗口对象请求对象的服务过程。因此，用面向对象方法来进行 Windows 程序的设计与开发是极其方便和自然的。

还有，Windows 是一个多任务的操作系统，也就是说，在同一时刻，在 Windows 中有着多个应用程序的实例正在运行。在这样的一个操作系统中，不可能像过去的 DOS 那样，由一个应用程序来享用所有的系统资源，这些资源是由 Windows 统一管理的。那么，特定的应用程序如何获得用户输入的信息呢？事实上，Windows 时刻监视着用户的一举一动，并分析用户的动作与哪一个应用程序相关；然后，将用户的动作以消息的形式发送给该应用程序，应用程序时刻等待着消息的到来，一旦发现它的消息队列(message queue)中有未处理的消息，就获取并分析该消息；最后，应用程序根据消息所包含的内容采取适当的动作来响应用户的操作。这里提到的消息，其实是一种 Windows 内设的数据结构(MSG)。

使用可视化开发工具开发 Windows 的应用程序，看起来就好像是在屏幕上绘出程序的各个窗口，在窗口中恰当地安排每一个控件。但是随着学习的深入，你会发现对于 Windows 程序设计，更重要的在于知道 Windows 操作系统和 Windows 应用程序的运行机制，以及它们之间以何种方式来进行通信，然后明确自己在编写 Windows 应用程序时所需做的工作。