高等学校计算机教材

C++大学基础教程

徐惠民　主编

人民邮电出版社
POSTS & TELECOM PRESS

高等学校计算机教材

C++大学基础教程

徐惠民　主编

人民邮电出版社

图书在版编目（CIP）数据

C++大学基础教程/徐惠民主编. —北京：人民邮电出版社，2005.2
高等学校计算机教材
ISBN 7-115-13098-1

Ⅰ.C… Ⅱ.徐… 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（2005）第011621号

内 容 提 要

本书是适应计算机技术发展和教学改革需要而编写的大学程序设计课程新教材。

本书共13章。前7章覆盖了C++基本程序设计的内容，后6章讲述了C++面向对象程序设计的思想和基本方法。教材中对于C++中非常重要的指针、引用、封装、继承、多态和异常处理等都作了详细而清晰的叙述。

教材的编写目的是为学生打好程序设计的基础，因此，特别注意在介绍基本概念和基本方法的同时，重视良好编程习惯的培养。

本书适合作大学程序设计课程的教材或专门的培训教材，也可作为研究生的相关课程的参考和程序设计人员的参考。

高等学校计算机教材

C++大学基础教程

◆ 主　　编　徐惠民
策划编辑　滑　玉
责任编辑　须春美

◆ 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：010-67129259
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：20
字数：480千字　　2005年2月第1版
印数：1－5 000册　　2005年2月北京第1次印刷

ISBN 7-115-13098-1/TP・4428

定价：26.00元

编 者 的 话

目前，国内大多数工科专业的计算机程序设计语言都是C语言，主要学习面向过程的程序设计方法。

但是，随着计算机技术，特别是软件技术的发展，面向对象程序设计语言的问题必然提到高校教学的议事日程上来。最简单的解决办法是在学习了“高级语言程序设计”课程后再选修一门“面向对象程序设计”。但是，由于学时数的限制，这种做法很难实现。

北京邮电大学电信工程学院从 2000 年开始，对一年级的程序设计课程作了重大改革，用C++语言程序设计取代了原来的C语言程序设计，至今已历时5年。

要让大学一年级的学生既要学习程序设计的基本方法，还要学习面向对象程序设计的思想和技术，无论对学生还是对老师都很不容易。但是，等到学生学习专业课和做毕业设计的时候，就觉得一年级的这种付出是值得的。

通过5年的实践，我们感到，一本合适的教材对于这样一门课程是特别重要的。其实，想写教材的念头早几年就有，但总是担心积累不够、体会不深。现在，经过几位老师的共同努力，《C++大学基础教程》终于得以出版。

我们将书名定为《C++大学基础教程》是出于以下的初衷：

1．教材的基本对象应该是大学低年级的学生，教材的内容应该是由浅入深的。在第 1 章大家可以看到，我们在介绍面向对象程序设计的特点时，就是用非常易懂的语言来叙述的。

2．教材的目的就是为同学们打好程序设计的基础而写。因此，特别着重基本概念和基本方法的介绍。同时，要在教学过程中，重视良好编程习惯的培养。

3．教材是《C++大学基础教程》，就应该不同于非大学的教程。教材中对于 C++中非常重要的指针、引用、封装、继承、多态和异常处理等都有详细而清晰的叙述。相信就是有经验的编程人员，阅读本书也会有很大收获。

实验和实践环节对于学好程序设计是特别重要的。在教学中，我们精心的设计了各章的实验和课程设计的内容。对于课程设计我们要求是能够完成一个基于 MFC 的程序设计。有关的材料将另行出版。

本书共13章。前7章覆盖了C++基本程序设计，后6章是C++面向对象程序设计。第1、2、10章由徐雅静编写；第3、4、9章由赵衍运编写；第5章由杨文川编写；第6、11、13章由徐惠民编写；第7、8、12章由刘瑞芳编写。全书由徐惠民主编和统稿。

本书的电子教案可以通过人民邮电出版社的网站获得。本书的习题答案，已编为《教师手册》，凡使用本书作为教材的老师，如果需要《教师手册》，可以和编者联系，邮件地址是huimin@bupt.edu.cn。本书的其他相关材料，可以通过北京邮电大学的精品课程网站获得，网址是teach.jwc.bupt.cn。

本书不当之处，希望教师同行和同学们不吝指教。

编 者

2004 年 12 月

目　录

第 1 章　C++初步

C++语言是一优秀的程序设计语言，它完全兼容 C 语言，并且以其独特的语言机制在计算机科学领域中得到了广泛的应用。

本章将学习：

- C++语言的发展史；
- 开发 C++程序的步骤；
- 尝试自己第一个 C++程序。

1.1　程序设计语言的发展

1946 年世界上第一台电子计算机 ENIAC 诞生，当时对计算机的控制使用的是**机器语言**。机器语言是简单的“0”和“1”的组合，便于计算机识别，但对于人类来说却晦涩难懂，难于记忆和使用，并且机器语言与硬件相关，不同的计算机使用的机器语言不同。

到了 20 世纪 50 年代末，出现了晶体管计算机，计算机运算速度从每秒几万次达到每秒钟几十万次，**汇编语言**出现并发展起来。汇编语言将机器语言映射为一些可以被人们读懂的助记符，如“ADD”、“SUB”等，方便人类记忆和使用。但汇编语言也是与机器硬件相关的，机器语言和汇编语言都属于低级语言。

随着 20 世纪 60 年代早期集成电路的出现，**高级语言**开始出现并逐步发展起来。高级语言是计算机编程语言的一大进步，人们不必了解机器的细节，通过更高层次的数据抽象使程序更能体现客观事物的结构和逻辑关系，这使得编程语言和人类的自然语言更加接近。但二者之间还是有很大的区别，因此，程序设计语言仍然在不断的发展中。目前，程序设计语言的发展过程如图 1-1 所示。

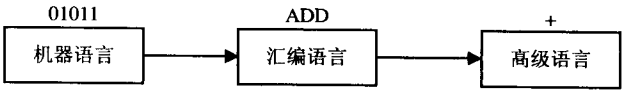

图 1-1　程序设计语言的发展过程

高级语言的发展也经历了不同的阶段，20 世纪 60 年代末出现了面向过程的程序设计语言，进一步提高了语言的层次。面向过程的程序设计语言通过结构化数据、结构化语句、数据抽象和过程抽象等概念使程序便于体现客观事物的结构和逻辑含义；缺点是程序中数据和操作分离，不能够有效地组成自然界中具体事物紧密对应的程序成分。目前，广泛应用的面向过程的高级语言有 Basic、Pascal 和 C 等。

20 世纪 80 年代初出现了面向对象程序设计语言，这种语言设计的出发点就是为了能更

直接地描述客观世界中存在的事物（即对象）以及它们之间的关系。面向对象的编程语言将客观事物看成是具有属性和行为的对象，通过抽象同一类对象的静态特征（数据）和动态特征（操作或行为）形成类。目前，广泛应用的支持面向对象的高级语言有C++和Java等。

这些高级语言由于时代、思维方式、编程任务以及个人喜好等因素的不同，出现了许许多多的描述方式，对这些不同的描述方式，称之为不同的语言。

例如，要向屏幕打印一个“A”字符，不同的语言其描述方式是不同的。

```
Basic 语言的描述：     PRINT"A"
Pascal 语言的描述：    writeln ('A');
C 语言的描述：         printf("A");
C++语言的描述：        cout <<"A";
Java 语言的描述：      System.out.print ("A");
```

对于面向过程的编程语言，使用不同的函数“PRINT”、“writeln”和“printf”来实现输出字符功能；而对于面向对象的编程语言，则将打印功能作为系统对象的一个行为，使用不同的对象“cout”和“System”实现输出字符。

1.2 面向过程的程序设计

面向过程的程序设计又称为结构化的程序设计，一般强调的是三种基本结构，以及模块的单入和单出。三种基本结构指的是顺序、分支和循环结构；模块的单入指的是该模块被哪些模块所调用，单出指的是该模块调用了哪些模块。

以前程序被看作是一系列处理数据的过程。一个过程或函数是指一个接一个执行的指令。数据与过程相分离，编程的技巧主要是跟踪哪些函数调用了另一些函数以及哪些数据被改变了。为了避免发生混乱，于是出现了面向过程的程序设计。

面向过程的程序设计的主要思想是：自顶向下，逐步求精。

一个计算机程序可以看成是由一系列任务组成的，任何一项任务如果过于复杂就将其分解成一系列较小的子任务，直至每一项任务都很小，很容易解决。

例如：计算每门课程的平均成绩，可以分成以下子任务：

（1）一共有多少门课

（2）每门课选课的学生总人数

（3）计算每门课所有学生的总分

（4）用每门课的总分除以每门课选课的学生人数

其中，第（3）项任务还可以分成以下一系列子任务：

（1）找出每门课选课的学生档案

（2）从档案中依次读出该课的成绩

（3）累加到总成绩上

与此类似，在子任务中的（1）还可以分成以下一系列子任务：

（1）选择一门课

（2）查找选择该课的学生档案

（3）从磁盘读出数据

结构化编程虽然是处理复杂问题的一种非常成功的方法，然而，到 20 世纪 80 年代末，结构化编程的不足逐渐暴露出来。

首先，结构化编程将数据和过程相分离，但客观事物的特性往往与此相背离，数据（比如学生成绩）和对于数据的操作（排序、编辑等）是一个不可分割的整体。结构化编程重在过程，而不是数据和过程紧密联系在一起的对象。

其次，结构化编程对代码重用支持不够，处理相似的任务时，程序员们不得不作大量重复的工作，而不能使用已有的代码。可重用思想就是创建一些已知属性的组件，然后插入到自己的程序中，这是一种模拟硬件组合的方式，有如当工程师需要一个晶体管时，他不需要自己发明，只需要到仓库中去找一个就行了。

面向对象程序设计就是希望提供一种更加有效的手段，来尝试解决以上问题。

1.3　面向对象的程序设计

面向对象的程序设计将数据和处理数据的过程当成一个整体——对象，并且具有以下三种特性：封装、继承和多态。

1. 封装

当一个技术人员要安装一台计算机时，他将各个备件组装起来。需要声卡时，不需要用原始的集成电路或材料去制作一个，只需要去购买一个声卡，插到机箱中即可。技术人员只关心声卡的功能是不是符合要求，不需要知道声卡的内部原理、硬件电路。声卡是自成一体的，这就是封装性。无需知道内部如何工作就能够使用的思想称为数据隐藏。

在面向对象的程序设计中使用“对象”的概念支持封装。一个对象就是一个将数据和数据操作集合在一起的实体，你只需要知道这个对象的外部接口，不必知道对象的具体实现就可以使用它。

2. 继承

当一个工程师要制造一辆新车时，他有两种选择：或者从头做起，或者对已有的车型进行改进。如果现有的车型已经很好，只需要再添加一个新的变速装置，或添加一个新的功能，就更加完美了。那么我们没有必要从头来做，只需要利用现有车型，添加一些附加装置就可以产生一种新的车型。新的车型继承了原有车型的所有属性和行为，又增加了新的属性。这就是继承机制。

在面向对象的程序设计中，使用“类”的概念支持继承。我们可以实现一个原有车型的类，然后对这个类进行扩展，从而产生一个新的类。新类是从已有的类中派生出来的，不需要实现原有类的功能，只需要实现新添加的功能即可，从而实现了代码重用。

3. 多态

不同的车型，当司机踩下油门时，车的反应是不一样的，原因是不同的车型使用了不同的变速器。但作为司机不必知道这些差别，只需要知道驾驶的是哪款车，跟这款车型配套的机器自然运转就可以了。

面向对象程序设计中支持这种思想，它使用相同的接口（踩油门）但不同的类（车型）运行状态（车的反应）不一样。

1.4 C++的诞生

C++语言是从C语言发展演变而来的，因此我们首先回顾一下C语言的发展历程。

C语言最初是由贝尔实验室的Dennis Ritchie在B语言的基础上开发出来的，1972年他在一台DEC PDP-11计算机上实现了最初的C语言。之后C语言作为UNIX操作系统的开发语言而被人们广泛使用。C语言是与硬件无关的高级语言，并且由于C语言设计严谨，使得C语言编写的程序能够被移植到大多数计算机上。到20世纪70年代末期，C语言已经发展得比较成熟，Brian Kernighan和Dennis Ritchie在1978年合作出版的《The C Programming Language》一书中，全面地介绍了最初的C语言，该书只有两百多页，是最早的经典传统C语言的书，也被称为C语言的圣经。

C语言在各种计算机上的推广导致当时很快出现了多个不兼容的C语言版本，因此，需要制定一种C语言的标准（ANSI C）。1988年，Brian Kernighan和Dennis Ritchie编著的第二版《The C Programming Language》，全面介绍了标准C的内容。

C语言是一种面向过程的编程语言，它的优点很多：

① 语言简洁、紧凑、灵活；

② 丰富的运算符和数据类型；

③ 可直接访问内存地址，能进行位操作；

④ 程序运行效率高；

⑤ 可移植性好。

但随着时代的发展，它的缺点也逐步暴露出来：

① 类型检查机制弱，导致许多错误不能在编译时发现；

② 几乎不支持代码重用；

③ 对于大规模程序，很难控制程序的复杂性。

随着程序规模的逐步扩大，C语言的局限性也越来越明显，为了满足管理大规模程序的复杂性需要，1980年贝尔实验室的Bjarne Stroustrup开始对C语言进行改进和扩充，他根除了C语言中存在的问题，并使其支持面向对象的程序设计，将“类”的概念引入到C语言中。因此，形成了最初的“带类的C”。1983年经过进一步改进这种“带类的C”正式取名为C++。1985年Bjarne Stroustrup出版的《The C++ Programming Language》一书是最早介绍C++语言的经典著作，因此Bjarne Stroustrup也被誉为C++之父。

C++的标准化工作从1989年开始，经历了三次修订，最终于1998年ISO标准被批准。因此，Bjarne Stroustrup对《The C++ Programming Language》一书也经过了三次改版，1991年第2版，1997年第3版，2000年特别版。

C++语言全面兼容C语言，它保持了C语言的简洁、高效等特点，而且比C语言更安全，并全面支持面向对象的程序设计，这极大地促进了C++语言的发展。C和C++语言的关系如图1-2所示。Bjarne Stroustrup曾这样描述C++语言：“C++是一种通用的程序设计语言，其设计就是为了使认真的程序员能觉得编写程序变得更愉快。”

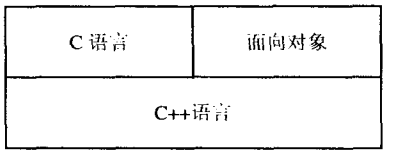

图 1-2　C 和 C++关系图

1.5　程序开发过程

大多数现代的编译程序都提供了一个集成开发环境（Integration Development Environment，IDE），本书选用的集成开发环境是 Microsoft Visual C++ 6.0，所有的程序都在该集成环境下编辑、编译、运行。为了帮助读者更好地理解程序开发的过程，我们首先描述几个基本概念。

◆　源程序

使用源语言编写的、有待翻译的程序称为源程序。Microsoft Visual C++ 6.0 集成环境支持 C++程序和 C 程序的编译和调试，因此源程序文件扩展名为.c 时，称为 C 源程序；在本书范围内，一律使用 C++语法编写的源程序，扩展名为.cpp，称为 C++源程序。

◆　目标程序

源程序经过翻译加工后所生成的程序称为目标程序。一般来说目标程序使用机器语言表示（也称为目标代码），扩展名为.obj。

◆　可执行程序

目标程序不能够直接运行，必须将目标程序和所用的其他资源进行链接，生成的可执行程序才能够运行。在 VC6.0 环境下，可执行程序的扩展名为.exe。

◆　翻译程序

翻译程序是指用来将源程序翻译为目标程序的工具。对翻译程序来说源程序是它的输入，目标程序是它的输出。翻译程序分成三类：汇编程序、编译程序和解释程序。

（1）汇编程序：将汇编语言编写的源程序翻译成机器语言形式的目标程序。

（2）编译程序：将使用高级语言编写的源程序翻译成机器语言形式的目标程序。在 VC6.0 环境下，编译程序是安装目录下的一个名为“CL.exe”的文件，在集成环境中，一般使用“compile”命令进行编译。

（3）解释程序：将使用高级语言编写的源程序翻译成机器指令。它与编译程序的区别在于解释程序是边翻译边执行，即输入一句、翻译一句、执行一句，直到将整个源程序翻译并执行完毕。解释程序不产生整个目标程序，因此对于源程序中循环执行的语句需要反复解释执行，效率较低。Basic 语言就是典型的使用解释程序的编程语言。

◆　链接程序

用来将汇编程序或编译程序生成的目标程序，加上所需的其他资源进行链接，生成可执行文件的程序。对链接程序来说，目标程序是它的输入，可执行程序是它的输出。在 VC6.0 环境下，链接程序是安装目录下的一个名为“LINK.exe”的文件，在集成环境中，一般使用“build”命令进行链接。

VC++ 6.0 集成开发环境是使用编译方式进行的。在该环境下，开发 C++程序的步骤如图 1-3 所示。

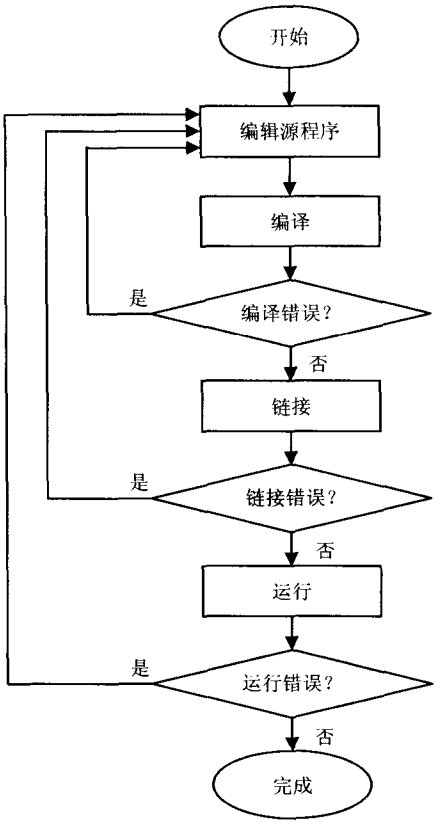

图 1-3 开发 C++程序的步骤

1.6 最简单的程序

我们从一个最简单的程序入手，来学习和分析 C++的程序构成。下面的程序能够在屏幕上输出“Hello World!”的字样，具体程序清单如下：

```
/*************************************************
程序文件：ch1_1.cpp
程序功能：在屏幕上输出 hello world!
作　　者：XXX
创建时间：XX 年 XX 月 XX 日
输　　入：无
输　　出：字符串 Hello World!
*************************************************/
#include <iostream>
using namespace std;
void main()
```