21st CENTURY
规划教材

面向21世纪高等院校计算机系列规划教材

COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

C++面向对象程序设计

张德慧　周元哲　主编

科学出版社
www.sciencep.com

规划教材
面向21世纪高等院校计算机系列规划教材
COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

C++面向对象程序设计

张德慧　周元哲　主编

科学出版社
北京

内 容 简 介

本书简明地讲述了面向对象程序设计的基本概念，对C++的面向对象特性：类、对象、派生类、继承、多态性、虚函数、C++标准库等作了深入浅出的介绍。本书还使用大量简单的实例循序渐进地介绍了C++面向对象程序设计的基本编程方法。

本书论述清晰，系统性强，并且理论与实例紧密结合。本书既可作为高等院校计算机或相关专业的教材，也可作为计算机应用开发人员的自学用书。

图书在版编目（CIP）数据

C++面向对象程序设计/张德慧，周元哲主编. —北京：科学出版社，2005

（面向21世纪高等院校计算机系列规划教材）

ISBN 7-03-015583-1

Ⅰ.C… Ⅱ.①张… ②周… 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052972号

责任编辑：万国清 马 琳/责任校对：耿 耘

责任印制：吕春珉/封面设计：王 浩

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

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

*

2005年7月第 一 版 开本：787×1092 1/16

2006年7月第二次印刷 印张：17 3/4

印数：3 001-5 000 字数：405 000

定价：23.00元

（如有印装质量问题，我社负责调换〈环伟〉）

销售部电话 010-62136131 编辑部电话 010-62138978-8004（HI02）

前　言

面向对象程序设计（Object-Oriented Programming，OOP）是当前主流的程序设计方法。它能有效降低软件的复杂性，极大地改善软件的重用性和可维护性，显著地降低软件的生产成本，提高软件的质量和开发效率。因此面向对象的程序设计方法已经取代了传统的结构化程序设计的地位而成为软件工业界主流的程序设计方法。

C++语言是在C语言的基础上发展而来的。在C++中不但加入了支持面向对象程序设计的特性，还增加了许多新的特性，并且保持了C语言的高效、简洁和灵活性。如果我们撇开C++的面向对象的特性，则可以把C++看成是“更好的C”。使用C++编写结构化程序也比使用传统的C语言更容易一些。

C++与C高度兼容，这正是C++成为最流行的面向对象程序设计编程语言的一个很重要的原因。一个C程序员通过学习C++来掌握面向对象程序设计，可以使学习的困难降低，而且学习的进展也较快。

为了适应信息产业的需要，目前有许多高等院校都开设了面向对象程序设计的课程。我们在教学中发现有相当多的学生感到面向对象程序设计很难学，究其原因，主要有两个：第一，面向对象程序设计的概念比较多，许多书上又将这些概念编排得很集中，以致理解、记忆这些概念有困难；第二，C++语言是一门实用性的语言，它很复杂而且庞大，初学者容易迷失在一些语言细节中。我们的宗旨是要为面向对象程序设计的初学者解决上述困难，提供一本通俗易懂的快速入门的书，使读者能够在较短的时间内理解面向对象的基本概念并掌握C++面向对象程序设计的方法。因此，我们在编写本书时尽量将面向对象程序设计的概念分散到各章节，以便读者循序渐进地来学习这些基本概念。考虑到大部分高等院校都开设了“C程序设计”的课程，所以我们主要介绍C++语言的面向对象特性和一些常用的新特性，着重介绍如何使用C++语言进行面向对象程序设计。C++语言的发明者Bjarne Stroustrup说得很好，学习一门程序设计语言是为了更好地进行程序设计，更有效地设计新系统和维护已有的系统。因此，对程序设计方法和技巧的掌握比对语言细节的理解重要得多，而对语言细节的透彻理解需要大量的时间和长期的编程实践。

本书共分12章。第1章介绍C++语言在非面向对象方面的一些改进。第2章引出类和对象的概念，并讨论构造函数和析构函数的作用。第3章概述面向对象程序设计的基本概念，包括类和对象、消息和方法、面向对象方法的基本思想、数据抽象等；讨论面向对象程序设计的主要特征：封装性、继承性、多态性等；简单介绍了面向对象的软件工程。第4章介绍对象数组、对象指针和引用的使用，讨论对象作为函数参数和返回值的情况，另外还介绍了类的静态成员和友元的概念。第5章讨论堆与拷贝构造函数。第6章介绍C++语言实现继承性的方法：派生类及其运用。第7章讨论运算符重载。第8章重点讨论多态性和实现动态多态性的方法：虚函数及其运用。第9章介绍模板的概念及其应用。第10章讨论类库概念和C++的标准模板库STL使用方法。第11章介绍C++的标准I/O流类库。第12章讨论C++语言的异常处理方法。

学习计算机程序设计的最好方法是实践，因此我们建议读者上机编写、运行和调试本书所给的例程。本书中的所有程序都在 Visual C++ 6.0 环境中调试通过。

本书第 1～5 和 10～12 章由张德慧编写，第 6～9 章由周元哲编写。中国人民解放军第二炮兵青州士官学校的吕建鹏审校了全部书稿，并提出很多宝贵的修改意见。

西安邮电学院的韩俊刚、蒋林对本书的编写给予了大力的支持并提出了指导性意见。西安邮电学院的马力对本教材的写作大纲、写作风格等提出了很多宝贵的意见并审阅了部分书稿。梁琛、王小银也审阅了部分书稿并提出了不少修改意见。衷心感谢上述各位的支持和帮助。

感谢我的家人，如果没有她们的支持，我不可能如期地完成本书的写作。

由于作者水平有限，本书难免有不足之处，我们诚恳期待读者的批评指正，以使本书日臻完善。我们的电子信箱是xazhangdehui@163.com 。

目　录

第1章 C++概述

1.1 C++的起源和特点

1.1.1 C++的起源

C++语言是在 C 语言的基础上发展而来的。C++首先由美国贝尔（Bell）实验室的 Bjarne Stroustrup 博士在 20 世纪 80 年代初期提出。开始他把这种新的语言叫做“含类的 C”，到 1983 年才改名为 C++。这表明 C++语言在语法上对 C 的主要扩展是对类结构的实现。

尽管 C 语言仍然是世界上最受欢迎和应用最广的专业程序设计语言之一，但 C++的出现是必然的，这主要是由程序设计的复杂性决定的。这些年来，计算机程序变得越来越复杂。一个 C 程序的代码长度一旦达到 25000～100000 行，它就会变得十分复杂，非常难于理解和控制。而 C++的目的正是要扫清这个障碍。如果使用 C++语言按照面向对象方法进行程序设计，程序员就能够理解和管理更大、更复杂的程序。

C++自问世以来经历了三次主要修订。第一次修订是在 1985 年，AT&T 公司发布了 C++ 1.0 版和第一个 C++“编译器”。这个 C++“编译器”实际上是一个前端翻译器，被称为 cfront，它的作用是把 C++代码转换成 C 代码，然后 C 代码被 C 编译器和本地装入程序处理成可执行代码。第一个真正的 C++编译器于 1988 年诞生，目前比较流行的是由 Borland、IBM、Microsoft 等公司开发的编译器和集成开发环境（IDEs）。第二次修订是在 1989 年，AT&T 公司发布了 C++ 2.0 版，这是一个重大进步，其中主要包括类的多继承。C++3.0 版本（1991 年）中引入了模板。第三次修订是开始制定 ANSI C++标准。ANSI C++标准草案是以 C++ 4.0 版为基础的，4.0 版中引入了异常处理、运行时类型识别（RTTI）和名字空间。ANSI C++委员会保留了 4.0 版本中定义的所有特性，并增加了一些新的特性。ANSI C++标准在 1997 年被批准。C++的国际标准文本 ISO C++也于 1998 年完成。

1.1.2 C++的特点

C++发明者 Stroustrup 认为，C++在增加对面向对象程序设计的支持的同时，很有必要保持 C 的原来的精髓，包括 C 的高效性、灵活性以及设计思想。令人欣慰的是他最终实现了这个目标。用 Stroustrup 的话说，C++的面向对象的特点就是“使程序结构清晰、易于扩展、易于维护而不失其效率”。C++继承了 C 的所有优点，并有自己独到的特点，其主要特点有：

① 支持面向对象的程序设计，能直接在程序中映射问题空间的结构，可方便地构造出模拟现实问题的对象和操作。

② C++程序代码具有很好的可重用性，从而显著地降低软件开发费用，缩短开发时间。

③ 用 C++编写的代码可读性更好，程序结构更为合理并容易理解，使得软件能够更好地反映真实世界，大大降低了开发大中型软件系统的难度和复杂性，同时增加了软件的可维护性。

④ C++中类的定义方式增强了数据隐藏性，外部（客户）程序在使用类时不会改变类的内部实现，从而使得使用 C++开发的软件具有很好的可靠性。

⑤ C++保持与 C 兼容，这就使得多年积累下来的数目庞大的 C 代码仍然可以被 C++的程序员所使用。这一点还使得为数众多的 C 程序员比较容易转向面向对象的程序设计，而且可以保护软件公司过去在 C 语言上的投资。

由于 C++具有这些吸引人的特点，加上其标准化工作已经完成，它目前已经成为软件工业界主流的程序设计语言。现在人们不但使用 C++语言开发软件程序，还用它来设计硬件芯片。

1.2　C++程序的结构

1.2.1　C 程序与 C++程序比较

C++语言是在 C 语言的基础上发展起来的，它保持了与 C 语言的高度兼容性。下面，我们通过表 1.1 比较 C 程序与 C++程序的结构，来了解 C++语言的程序结构。

表 1.1　C 程序和 C++程序比较之一

行号	C 程序	C++程序
1	#include "stdio.h"	#include "iostream.h"
2	main ()	int main ()
3	{ int a, b, sum;	{ int a, b, sum;
4	/* 定义三个整型变量 */	//定义三个整型变量
5	a = 123; b = 456;	a = 123; b = 456;
6		
7	sum = a + b;	sum = a + b;
8	printf ("sum is %d\n", sum) ;	cout<< sum;
9	}	}

表 1.1 中的 C 程序和 C++程序完成相同的功能。通过对比这两个程序，可以看出二者的结构基本上是一样的。下面先对两个程序的不同之处进行说明。在源程序的第 2 行，C 程序的 main 函数没有给出返回值类型，而在 C 语言中，如果没有给出函数的返回值类型，则函数的返回值类型默认为 int 类型；在 C++语言中，必须给出函数的返回值类型，否则会出现编译时错误或者警告信息。在源程序的第 4 行，C++程序采用了一种新的注释方式，即 C++语言新引入的单行注释方式。这种注释方式以“//”开始，到行尾

结束。在源程序的第 8 行，C 程序中使用 printf 函数来完成结果的输出；而在 C++程序中使用标准的输出流对象 cout 来实现结果的输出。最后我们回头看源程序的第 1 行，C 程序通过包含头文件 stdio.h，才能使用标准输入/输出库中的函数；C++程序通过包含头文件 iostream.h，才能使用标准输入/输出流库中的对象 cin 或者 cout，进行 I/O 操作。iostream 是 input output stream 的缩写。

接下来再比较表 1.2 中的两个程序，可以发现这两个程序的主要差异在第 5 行和第 7 行。表中 C 程序中的第 5 行通过 scanf 函数从键盘输入变量 a，b 的值，C++程序中的第 5 行通过输入流对象 cin 从键盘输入变量 a，b 的值。在第 7 行，C 程序中通过 printf 函数输出变量 c 的值，而 C++程序使用输出流对象 cout 输出变量 c 的值。可见完成相同功能的 C++程序比 C 程序更加简明。

表 1.2　C 程序和 C++程序比较之二

行号	C 程序	C++程序
1	#include "stdio.h"	#include "iostream.h"
2	int max (int x, int y) ;/*函数声明*/	int max (int x, int y) ; //函数声明
3	main ()	int main ()
4	{ int a, b, c;	{ int a, b, c;
5	scanf ("%d, %d", &a, &b) ;	cin>>a>>b;
6	c=max (a , b) ;	c=max (a , b) ;
7	printf ("max= %d\n", c) ;	cout<<"max= "<<c<<'\n';
8	}	}
9	int max (int x, int y)	int max (int x, int y)
10	{ int z;	{ int z;
11	if (x>y) z=x;	if (x>y) z=x;
12	else z=y;	else z=y;
13	return (z) ;	return (z) ;
14	}	}

函数定义　　函数定义

1.2.2　C++程序结构

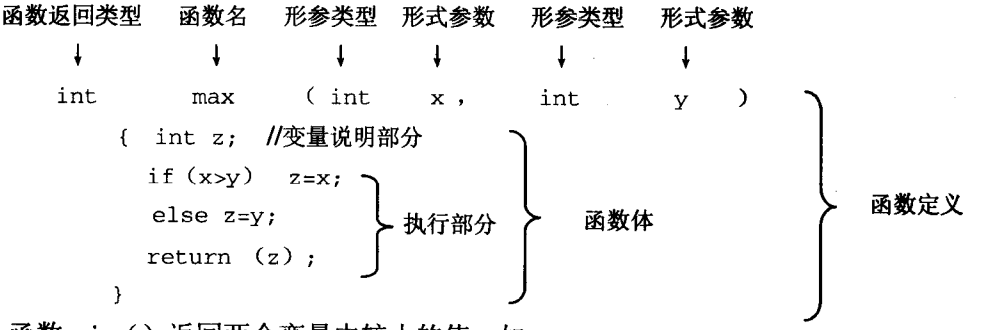
从 1.2.1 节内容可以看出 C++程序与 C 程序在结构上很相似，有以下特点。

① 所有 C++程序都是由一个或者多个函数组成。每个 C++函数都必须有一个名字，其中名为 main 的函数称为主函数，可以将它放在程序的前部，也可放在程序的中部或后部。通常程序运行时第一个被执行的函数是主函数。因此，一个完整的 C++程序必须包含主函数 main ()。

被调用的函数可以是 C++标准库中的函数，如 printf () 函数；也可以是用户自己编写的函数，例如表 1.2 中的函数 max ()。对于用户自己定义的函数，使用前应给予“声明”(declaration)，如表 1.2 中的 int max (int x，int y)；并且还需要在程序中给出函数的“定义”(definition)。

② C++函数可以被看作是一个子程序（subroutine，子例程），它是由用户定义的、

完成一个或者几个特定任务的操作过程，可以被反复调用。函数的操作数称为参数（parameter），由一个位于括号中，并且用逗号分隔的形式参数表（parameter list）指定。函数的结果被称为返回值（return value）。返回值的类型被称为函数返回类型（return type）。不产生值的函数返回类型是 void，意思是什么都不返回。函数执行的动作在函数体（body）中指定，函数体包含在花括号｛｝中，有时也称为函数块（function block）。函数返回类型以及其后的函数名、形式参数表和函数体构成了函数定义（function definition）。函数定义说明如下所示。



函数 min（）返回两个变量中较小的值，如：

```
int min ( int p1,  int p2 )
{// 返回较小值
    return ( p1 < p2 ? p1 : p2 ) ;
}
```

函数 abs（）返回整型变量的绝对值，如：

```
int abs ( int n )
{ // 返回 n 的绝对值
    return ( n < 0 ? -n : n ) ;
}
```

从以上几段程序可见 C++的函数组成与 C 的函数组成是相同的。函数都有一个名字，我们通过函数的名字和实际参数的列表（argument list）来调用函数。例如：

```
c=max (a, b) ; //a, b 为实际参数 (actual argument)
```

调用函数时，实际参数的个数必须与形式参数的个数相等，实际参数的类型必须与其对应的形式参数类型匹配。我们也可以定义无参数的函数，但无论是在函数定义或者函数调用时，函数名后面的圆括号不能省略。

③ C++中的每个语句和数据定义必须以分号结束。分号是 C++语句的必要组成部分，程序中的最后一个语句也应是以分号结束。由于用分号来区分每个语句，因此几乎可以用任何格式来书写 C++程序。一行内可以写多个语句，一个语句也可以分开写在几行上。

④ C++标准要求在所有 C++程序中都必须包含一个 main（）函数。主函数 main（）是 C 程序执行开始的地方。从理论上来说，C++程序的执行也开始于对函数 main（）的调用，并且在大多数情况下，当从主函数 main（）返回时程序执行结束。但是当 C++程序中存在全局对象时，对全局对象进行初始化的函数可能会在主函数 main（）开始执行之前执行。

与 C 语言一样，C++语言的函数之间也可以互相调用，但主函数 main（）是一个特殊的函数，它不能被其他函数调用。我们可以将函数 main（）看作是由操作系统调用的一个函数，对于大多数计算机操作系统而言，返回值为零表示程序正常结束，而返回非零值则表示程序由于遇到某种错误而结束。因此，C++标准要求主函数 main（）的返回类型必须为整型 int，可以使用 return 语句从 main（）中向操作系统返回一个整型值。

【例 1.1】主函数 main（）的返回值的含义。

```
#include <iostream>
using namespace std;
int main ()  // int 指定了 main () 函数返回值的数据类型
{ int a,  b,  c;
  cin>>a>>b;
  if (b==0)  return -1; // main () 函数返回非 0 值表示程序遇到错误而结束
  c= a/b;
  cout<<"c= "<<c<<'\n';
  return 0; //程序正常结束，函数 main () 返回值为 0
}
```

执行例 1.1 的程序时，当用户输入的除数为零，程序将返回值-1，表示程序在执行过程中出现了错误。

例 1.1 程序中的第 1 行和第 2 行，用来支持本程序使用的 C++标准库中提供的输入和输出功能，与下面这一行的作用是一样的。

```
#include "iostream.h"
```

例 1.1 程序中的第 2 行语句告诉编译器，在源文件中将使用 std 名字空间。名字空间（namespace）是 C++中的新增特征。在本书第 10 章将对名字空间进行详细的讨论，这里仅作简单的介绍。名字空间创建了一个用于声明的区域，程序的各种要素都可以放在这个区域中。在不同名字空间中声明的程序要素是相互独立的。在组织大型程序的时候，名字空间是很有用的。本例程序中通过使用 std 名字空间，可以简化在程序中访问标准库的方法。

注意： *头文件 iostream 与头文件 iostream.h 是不同的两个文件，在本书的第 10 章将比较详细地讨论这个问题。作为 C++的初学者，我们现在只需要记住，在 C++程序中必须包含下面的两行，才能使用 C++标准库中的输入/输出功能。以后读者将会看到，本书中几乎所有的 C++程序都包含了这两行代码。*

```
#include <iostream>
using namespace std;
```

1.2.3　C++程序的编辑、编译和运行

学习 C++程序设计的最好方法是动手实践，因此我们建议读者从开始学习 C++面向对象程序设计课程时就要建立一个合适的实践环境，可使用本书中的示例程序或者自己动手编写一些简单的 C++程序上机练习编程。

要将 C++源程序转换成可执行的程序，我们需要按照下面的步骤来进行操作。

① 输入C++源程序。

② 将源程序编译成目标程序；然后将各个目标模块链接成可执行程序。

③ 运行程序。

在不同的操作系统平台上，使用不同的C++编译器完成C++程序的编辑、编译和运行的过程都有差异，所以我们必须参考所使用的编译器的帮助文档来获得更详细的信息。

C++标准文本对C++源程序文件（存储源代码的文件）的名字没有任何规定，因此从理论上说可以是任意的。然而，通常所见的C++编译器要求将C++程序的源代码存储在扩展名为.cpp（即C plus plus的缩写）的文件中。因此，可以将程序的源文件命名为一个自己喜欢的任意名字，但文件的扩展名应该是.cpp。在上面的示例中，我们可以将源文件命名为ex1_1.cpp，这样就符合了文件名的规定。我们可以模仿上面的方法为其他的C++程序源文件选择名字。

常用的C++编译器，如微软的Visual C++、Bloodshed的Dev-C++以及Borland C++ Builder都带有C和C++两种编译器，当源程序文件扩展名为.c时，启动C编译器；当源程序文件扩展名为.cpp时，启动C++编译器。

1.3　C++的新特性

C++是从C发展而来，C程序中的表达式、语句、函数和程序的组织方法等在C++中仍可以使用。C++在C语言的基础上增加了许多新特性，其中最重要的是支持面向对象程序设计的特性，这也是本书要介绍的核心内容。C++还增加了一些非面向对象的新特性，这些新特性使C++程序比C程序更简洁或更安全。下面就先介绍C++对C的非面向对象特性的扩展。

1.3.1　单行注释和新的I/O流

在很多的实际程序中，人们发现其中大量的注释不超过一行。因此，C++语言引进了一种单行注释（single line comment）方式，该方式以“//”开始，到本行尾结束。如例1.1中的第一行。这种注释比传统的C注释方式更简洁高效。传统的C注释方式以“/*”（斜杠后面跟一个星号）开始，在遇到“*/”时结束。当然，在C++语言中仍然支持“/*…*/”这种方式，这种类型的注释被称为多行注释（multiline comment）。多行注释的长度可以是一行或者几行。通常，C++程序员在写大段的详细注释时使用多行注释，而在只需要简短说明的时候使用单行注释。

【例1.2】演示单行注释和C++的标准输出流对象和输入流对象。

```
// I/O stream  单行注释 (single line comment)
#include <iostream>
using namespace std;
int main ()
{  int i;   float f;
   char s[80];
```

```
    cout <<"Enter an integer, float, and string:";
    cin >>i>>f>>s;
    cout <<"Here's your data:"<<i<<' '<<f<<endl<<s<<'\n';
    return 0;
}
```

在例 1.2 中，cout 是预定义的标准输出流对象，类似于 C 语言中的 stdout，标准输出流对象代表控制台输出（console output），通常指的是计算机的屏幕。“<<”为输出运算符，可用于输出 C++语言中任何基本类型的数据。在 C++中将数据送到输出流中称为“插入”（inserting）或“放到”（putting）。所以，“<<”常被称为“插入运算符”（stream insertion operator）。

cin 是预定义的标准输入流对象，类似于 C 语言中的 stdin，标准输入流对象代表控制台输入（console input），通常指的是计算机的键盘。输入运算符“>>”从输入设备键盘取得数据送到输入流 cin 中，然后送到内存。在 C++中，这种输入操作称为“提取”（extracting）或“获取”（getting）。“>>”常被称为“提取运算符”（stream extraction operator），可用于输入 C++语言中任何基本类型的数据。

注意： 输入和输出并不是 C++语言的组成部分，它们由流库 iostream 支持。流库在 iostream.h 中声明并定义。因此 C++程序必须包含头文件 iostream.h，才能使用标准输入输出流库中的对象 cin 或者 cout 进行 I/O 操作。使用 cin 和 cout 来进行输入输出时，我们无须指出输入输出数据的类型，再也不像使用 scanf( )~printf( ) 之类的函数那样，由于格式控制字符串中指定的数据类型与实际变量的类型不一致而常常出现错误。

程序设计的初学者常常容易忽视“注释”（comment）的作用。实际上，适当的注释是一个良好程序的要素之一。加入恰当注释以后，不但能大大改善程序的可读性，而且能帮助程序员避免一些错误。当然，写出良好的注释通常与编写程序代码一样困难。大量的经验表明，为给程序加上合适的注释，付出时间是值得的。下面是一些注释的例子，值得我们学习。

```
// insert new value in the gap
ASSERT (nIndex + nCount <= m_nSize) ;

// copy elements into the empty space
while (nCount--)
        m_pData[nIndex++] = newElement;
...
/***
*eh.h - User include file for exception handling
*
*Copyright (c) 1993-1997, Microsoft Corporation
*All rights reserved.
*
*Purpose:
*User include file for exception handling
```