高等学校计算机程序设计课程系列教材

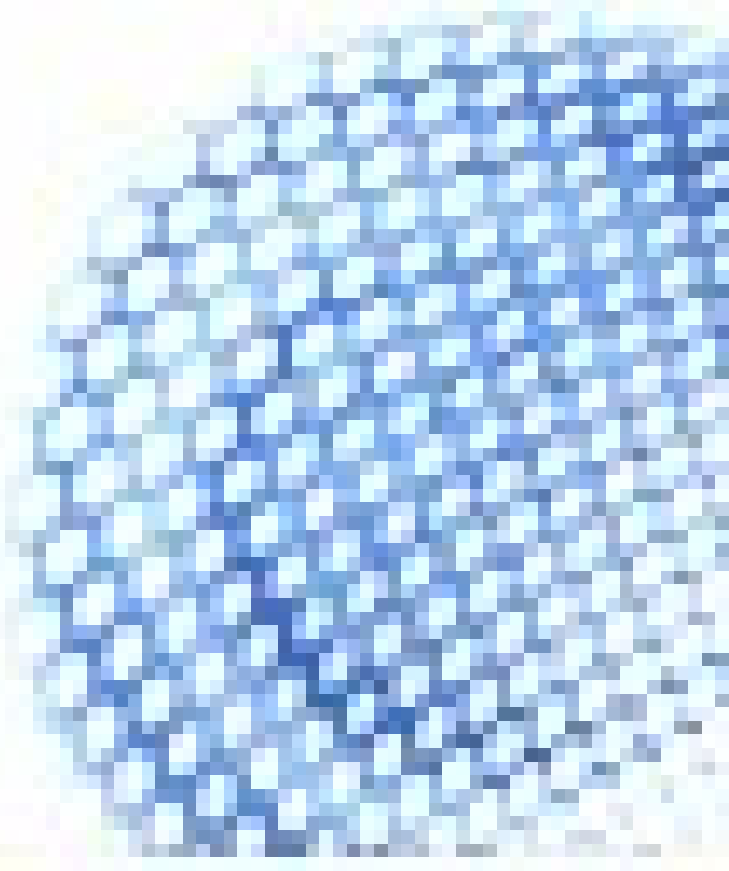
C++程序设计教程

陈建平　刘维富　葛建芳　编著

高等教育出版社
Higher Education Press

C++程序设计教程

[redacted]



高等学校计算机程序设计课程系列教材

C++程序设计教程

陈建平　刘维富　葛建芳　编著

高等教育出版社

内容提要

本书按程序设计方法的发展及C++实际编程能力形成的3个关键期，将C++内容分为三大单元，即结构化程序设计、模块化程序设计和面向对象程序设计，符合读者的认识规律和编程能力的形成规律，便于教学的组织、实施和考核，利于教学效果的巩固和教学质量的提高。

本书通过以基本语法和基本算法为主线的典型、综合范例程序，阐述有关程序设计的方法和思想，将C++语法、基本算法、程序设计方法和编程技巧有机结合起来，理论联系实际，注重对读者实际编程能力的培养。本书不回避教学和实际编程中的难点，精选综合范例程序，力求讲清讲透，帮助读者突破难点，进一步提高编程能力。

本书结构合理、定位明确、图文并茂、习题丰富，适合作为高校学生学习C++程序设计的基础教材，也适合程序设计的初学者或有一定编程经验、希望突破编程难点的读者自学。

图书在版编目(CIP)数据

C++程序设计教程/陈建平，刘维富，葛建芳编著.
—北京：高等教育出版社，2007.11
ISBN 978-7-04-022255-5

Ⅰ.C… Ⅱ.①陈…②刘…③葛… 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(2007)第141130号

策划编辑 张 龙 **责任编辑** 郭福生 **封面设计** 于文燕 **责任绘图** 吴文信
版式设计 陆瑞红 **责任校对** 俞声佳 **责任印制** 毛斯璐

出版发行	高等教育出版社	**购书热线**	010-58581118
社　　址	北京市西城区德外大街4号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经　　销	蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印　　刷	唐山市润丰印务有限公司	**畅想教育**	http://www.widedu.com
开　　本	787×1092　1/16	**版　　次**	2007年11月第1版
印　　张	24.25	**印　　次**	2007年11月第1次印刷
字　　数	590 000	**定　　价**	30.20元

物料号　22255-00

郑重声明

前　言

C++是兼容C的面向过程和面向对象的主流程序设计语言，广泛用于编写系统软件和应用软件，广泛用于程序设计、数据结构等课程的教学。“C++程序设计”是高等学校普遍开设的计算机核心基础课程，涉及程序设计的思想、方法、语法、算法、调试技术和操作技能，理论性、综合性和实践性强，使不少人感到难学、难入门，甚至入门后半途而废。

作者多年来一直从事C++程序设计的教学、研究、建设和软件开发，阅读了国内外大量C++教材，了解初学者学习C++的困难，积累和总结了C++教学的成功经验，形成了颇具特色的负反馈教学法，力求使C++不再难学。本书经过作者长期构思，精心写作，具有如下特点。

1. 结构合理，层次分明

本书按程序设计方法演进发展的自然顺序及C++实际编程能力形成的3个关键期，将C++教学内容分为三大单元，即结构化程序设计（1～4章）、模块化程序设计（5～9章）和面向对象程序设计（10～14章）。这样组织，由浅入深，循序渐进，符合读者的认识规律和编程能力的形成规律，便于教学的组织、实施和考核，利于教学效果的巩固和教学质量的提高。

需要强调的是，面向对象程序设计方法是以对象为模块的结构化程序设计方法，是对结构化程序设计方法的继承和发展，是计算机世界向现实世界迈进的重要一步，不应将面向对象程序设计方法与面向过程程序设计方法对立起来。

2. 紧跟标准，内容求新

本书介绍的是标准C++，符合ANSI/ISO C++标准（包括1998第1版和2003第2版），引导读者按C++标准编写程序。本书对以往教材中带有一定普遍性的问题，如“指针就是地址，地址就是指针”、“引用型变量不占内存”等，进行了澄清和纠正。

3. 不避难点，力求突破

许多有一定编程经验的读者学习程序设计半途而废的重要原因之一是未能突破实际编程中的难点，这与不少主流教材以各种理由极力回避难点有密切的关系。本书针对教学和实际编程中的难点（如递归、指针、动态内存分配和虚函数等），精挑细选范例程序，力求讲清讲透，帮助读者突破难点，学以致用，不至半途而废。

4. 范例程序，精心设计

程序是语法、算法、思想和方法有机结合的载体，是计算机解决实际问题的钥匙，学习程序设计离不开程序。书中大量的范例程序是经过作者精心挑选和精心设计的，表达准确、简练，书写规范，示范性强。

本书的配套教案是作者在校内讲授C++程序设计课程的实际教案。配套教案采

用 PowerPoint 制作，内容翔实，使用本书的教师可通过高等理工教学资源网（http://www.hep-st.com.cn）免费下载。

本书的配套教材《C++程序设计实验与编程实践》也将由高等教育出版社出版。

本书第1~4章由陈建平编写，第5、6、8、9、11、12和13章由刘维富编写，第7、10和14章由葛建芳编写，全书由刘维富副教授统稿，由陈建平教授和王波教授主审。书中带“*”号章节为选讲内容或选做习题，有一定难度，可以根据实际情况取舍。

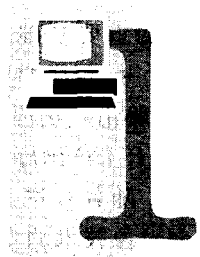
本书的编写得到了江苏省精品教材建设项目（苏教高[2005]26号）的资助，在出版过程中得到了高等教育出版社的支持和帮助，在此表示衷心的感谢。

由于作者水平所限，书中难免有疏漏与不妥之处，恳请同行和读者批评指正。

作　者

2007年5月

目 录



第1章 C++语言概述

本章首先介绍C++语言的起源、发展和特点；其次通过4个程序实例介绍C++程序的基本结构，以及面向过程和面向对象的程序设计思想和方法；最后介绍C++程序的开发步骤和上机调试流程，以及使用Visual C++ 6.0集成开发环境调试C++程序的详细过程。

1.1 C++语言的起源和发展

随着计算机科学技术的迅速发展，程序设计技术和程序设计语言也不断发展，经历了面向机器的程序设计、面向过程的程序设计和面向对象的程序设计几个阶段。目前，面向对象程序设计是软件开发领域的主流技术，而C++又是主流的面向对象程序设计语言。C++语言是在C语言的基础上发展起来的。早在20世纪60年代，Martin Richards为便于软件人员开发系统软件设计出BCPL(Basic Combined Programming Language)语言。1970年，Ken Thompson在吸收BCPL语言优点的基础上设计了B语言，但B语言功能有限。1972年，贝尔实验室的Dennis Ritchie和Brian Kernighan在B语言的基础上设计出了C语言，以编写UNIX操作系统。随着UNIX的成功和流行，C语言赢得人们的青睐。到20世纪80年代，C语言成为非常流行的结构化程序设计语言，应用领域从系统软件延伸到应用软件。

C语言具有以下主要特点。

- 语言结构化、简洁、规模小，数据类型丰富，使用灵活方便。既适用于设计和编写大型系统软件、大型应用软件，又适用于编写小程序。
- 兼有高级语言和汇编语言的特点，目标程序质量高，程序执行效率高，大大缩小了汇编语言的使用范围，提高了编程效率。
- 语言接口开放，不但便于C语言编译系统提供丰富的通用库函数，而且便于用户创建自己的专用库函数，也为第三方提供更为广泛的各类专业库函数打开了方便之门。由于有大量的库函数可供使用，因此，用户编写程序时，不必事事从

零开始，大大提高了编程效率。其实，库函数是出色的代码重用机制之一，是C语言广泛应用的重要基础。

• 程序的可移植性好。一是源于程序本身已将与硬件有关的语言成分尽可能剥离（如输入/输出），由库函数实现；二是源于程序有丰富的预编译命令支持；三是源于程序标准化程度高，有ISO国际标准。

随着C语言的广泛应用，它的一些不足受到人们的关注，如对数据类型检查较弱，没有对面向对象技术的支持，随着软件工程规模的扩大难以适应开发特大型的程序等。

1980年，贝尔实验室的Bjarne Stroustrup博士及其同事对C语言进行了改进和扩充，并把Simula 67中类的概念引入C中，1983年正式命名为C++（C Plus Plus），其含义是C语言的扩充。后来又把运算符的重载、引用、虚函数和模板等功能加入C++中，使C++的功能日趋完善。目前，C++已成为面向过程程序设计和面向对象程序设计的主流的通用语言。

C++语言受到软件厂商的极大支持，纷纷推出各自的商业化C++编译系统，从早期的Turbo C++、Borland C++、Watcom C++、Quick C++到目前流行的Visual C++和C++ Builder。C++语言也受到开放源代码组织的积极支持，它们也纷纷推出自己的非商业化C++编译系统，如GNU C++和DEV C++等。

C++语言的标准化工作始于1989年，于1994年制定了ANSI C++标准草案，经过不断修改完善，于1998年11月被国际标准化组织（ISO）批准为国际标准。

1.2 C++语言的特点

C++语言除了具备C语言的特点外，还具有以下的特点。

• 全面兼容C语言，全面支持面向过程的结构化程序设计。C++语言是C语言的超集，大多数的C程序代码略作修改或不做修改就可在C++编译系统下编译通过。这样，既保护了用C语言开发的丰富软件资源，也保护了丰富的C语言软件开发人力资源。

• 全面支持面向对象程序设计。以对象为基本模块，使程序模块的划分更合理，模块的独立性更强，程序的可读性、可理解性、可重用性、可扩充性、可测试性和可维护性等更好，程序结构更加合理。

• 全面支持面向过程和面向对象的混合编程，充分发挥两类编程技术的优势。

1.3 C++语言程序设计

目前，C++语言的编程环境通常是集多种编程工具于一体的，包括源程序的编辑、编译、链接、运行和调试等，使用非常方便，这种编程环境称为集成开发环境（IDE）。

C++语言的编程环境不仅支持C++程序的编译和调试，而且支持C程序的编译和调试。通常，C++的编程环境约定：源程序文件的扩展名为“.c”时，则为C程序；文件的扩展名为“.cpp”时，为C++程序。

本书主要介绍标准 C++语言及其程序设计，书中所有程序都在 Visual C++ 6.0 编程环境下调试运行。当然，读者也可以选用其他的 C++编程环境，如 Dev C++等。除非特别声明，书中所有范例的源程序文件扩展名均为“.cpp”。

为了对 C++程序的基本结构有所了解，例 1.1～例 1.4 分别给出了用面向过程和面向对象的程序设计方法计算圆面积的 C++程序。

例 1.1 面向过程程序设计。输入圆的半径，计算并输出该圆的面积。

```
/*******************************************
源程序名:ex1_1.cpp
功    能:计算并输出该圆的面积
计算方法:圆的面积 = π × r × r
输入数据:圆的半径
输出数据:圆的面积
程序设计:陈建平
设计日期:2006.2.8
*******************************************/
#include <iostream>
using namespace std;

int main(void/*无参*/)              //每个 C++程序有且仅有一个 main( )函数
{   float r;                        //定义浮点型变量 r,用于存放圆的半径
    cout <<"输入圆的半径:";         //显示提示信息,方便用户输入数据
    cin >> r;                       //从键盘上输入圆的半径送给变量 r
    cout <<"半径为" << r            //输出运算结果
         <<"的圆的面积 =" <<3.14159f * r * r <<'\n';
    return 0;
}
```

程序运行结果为：

```
输入圆的半径:1.5↙
半径为 1.5 的圆的面积 =7.06858
```

例 1.1 程序说明：

- 程序第 13 行中，用 main 代表主函数的名字，由于它是整个程序执行的入口，因此，这个名字不能改变，而且每个 C++程序无论规模大小都有且仅有一个 main( )函数。main( )函数在程序中的位置不限，但为了便于阅读，通常置于程序的头部或尾部。一个 C++程序至少包含一个 main( )函数，其余为库函数和自定义函数。main 前面的“int”的作用是声明函数的类型为整型，main 后面的“void”的作用是声明函数没有参数。
- 程序第 14 行～第 20 行由一对大括号({ })括起来，这部分是函数体。函数体可由零条或若干条语句组成，每一条语句均以分号(;)结束。

主函数内,首先声明了一个浮点型变量 r,用于存放圆的半径值,注意分号(;)是一个语句结束的标志,不能遗漏。

接下来的语句是用"cout"开头的,其作用是显示提示信息,方便用户输入数据,这对于使用该程序的人是非常有用的。由于 C++语言没有专门的输入/输出(简称 I/O)语句,此处借用了 C++标准库 iostream 中预定义的标准输入对象 cin(默认时 cin 代表键盘)和标准输出对象 cout(默认时 cout 代表显示器)实现数据的输入/输出,因此第 10 行需要包含头文件 iostream。由于所有 C++标准库定义的类型名、函数名和对象名等都限定在标准命名空间 std 内使用,因此,为了能在标准命名空间 std 外使用这些名称,就必须用到第 11 行的"using namespace std;"语句。cout 后面的"<<"称为插入运算符,用于将其后的数据插入 cout 对象的输出数据行列中,送到输出设备,若该设备是显示器(默认情况下),就能看到显示的内容。插入运算符后接的数据可以是字符(如程序中出现的'\n')、字符串(如程序中出现的"输入圆的半径:"等)和数值(如程序中出现的 r 等),其中,'\n'代表换行,表示此后显示的内容从下一行第 1 列开始,类似的,endl 也代表换行。cin 后面的">>"称为提取运算符,用于从 cin 对象(默认情况下是键盘)的数据行列中提取一个指定类型的数据送入其后的变量,提取数据的类型由其后的变量类型确定,如第 16 行的变量 r 为浮点数型,则提取一个浮点数送入变量 r。

此外,第 18 行出现的 3.14159f 是单精度浮点型常量。

- 程序中加入必要的注释,可提高程序的可读性。注释有以下两种。

一种是用"/*"和"*/"把注解括起来,可出现在程序的任何位置,可作为多行注释、单行注释,也可作为嵌入注释(如程序第 13 行,对 void 的说明),通常用于多行注释,如程序的第 1~9 行,对程序作了详细说明。

另一种是用"//"表示从此开始到本行结束为注释,通常用于单行注释,如程序的第 13 行。

编译器对程序中的注释不进行任何处理,注释对目标代码没有任何影响。

- 程序第 10 行。#include <iostream> 不是 C++语句,而是 C++的一个编译预处理命令(详见第 6 章),作用是将头文件 iostream 的内容包含到该命令所在的程序文件中,代替该行命令。编译预处理命令以"#"开头,行尾没有分号。由于每个 C++程序都涉及数据的输入/输出操作,即都会用到标准输入对象 cin 和标准输出对象 cout,因此,通常都要用#include <iostream>。

- C++语言严格区分字母大小写,例如,main 与 Main 是不同的。在书写程序或编辑程序时,要注意这一点。

- 在 C++程序中,程序的书写比较自由,一条语句可以写成若干行,如程序第 17~18 行,也可以一行写若干条语句。程序的可读性对编译器来说无关紧要,但程序的自由书写要便于程序的自我阅读和与他人交流。

- 程序运行结果的第 1 行中,用下划线标出的部分是实际运行程序时通过键盘输入的数据,"↙"表示 Enter 键(即通常所称的回车键)。

特别声明,为了节省篇幅,本书程序中不再使用像第 1~9 行那样的注释。

例 1.2 面向过程的模块化程序设计。输入圆的半径,计算并输出其面积。

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

```
float area(float r)                    //定义函数 area( ),求半径为 r 的圆面积
{   return 3.14159f * r * r; }

int main(void)
{   float r;                           //定义浮点型变量 r,存放圆的半径
    cout << "输入圆的半径:";           //显示提示信息,方便用户输入数据
    cin >> r;                          //从键盘输入圆的半径送给变量 r
    cout << "半径为" << r              //输出运算结果
         << "的圆的面积 = " << area(r) << '\n';
    return 0;
}
```

例 1.2 程序说明：

模块化程序设计的主要思路是:把一个复杂问题按功能分解成较为简单的若干个子问题,每个子问题通过定义一个函数来解决。如果子问题还不够简单,再继续按功能分解下去,直到所有子问题都能解决为止。这样,解决一个复杂问题的函数就可以通过调用一系列解决子问题的函数来实现。这就是所谓的“自顶向下,逐步求精”的程序设计方法,是“化整为零,各个击破”思想在程序设计中的体现。

例 1.2 的程序包括两个函数:主函数 main( )和被调用函数 area( )。程序的第 4 ~ 5 行是 area( )函数的定义,它的作用是计算半径为 r 的圆的面积。return 语句将所算的圆的面积值返回给调用它的函数。返回值是通过函数名 area( )带回到调用处的,如程序的第 12 行的“area(r)”所示。主函数 main( )将要解决的问题分解成 3 个简单问题,即输入圆的半径、计算圆的面积和输出圆的面积,然后分别通过使用标准输入函数 cin 实现数据输入,通过使用标准输出函数 cout 实现数据输出,通过调用自定义函数 area( )实现圆的面积计算,最终解决整个问题。有关函数的具体内容,可参见第 5 章。

***例 1.3** 兼容 C 的模块化程序设计。输入圆的半径,计算并输出其面积。

```
#include < stdio. h >

float area(float r)                          /* 定义函数 area( ),求圆的面积 */
{   return 3.14159f * r * r; }

int main(void)
{   float r;                                 /* 定义浮点型变量 r,存放圆的半径 */
    printf("输入圆的半径:");                 /* 显示提示,方便用户输入 */
    scanf("%f",&r);                          /* 从键盘输入圆的半径送给变量 r */
    printf("半径为%f 的圆的面积 = %f\n",r,area(r));     /* 输出结果 */
    return 0;
}
```

例 1.3 程序说明：