山东省高校统编教材

C++ 程序设计教程

第二版

主编　郭爱章

中国石油大学出版社

山东省高校统编教材

C++程序设计教程

第二版

C++CHENGXU SHEJI JIAOCHENG

主　编　郭爱章

副主编　王新刚

编写人员（以姓氏笔画排序）

石　扬　李　侃　张维玉　杨清波　杨振宇

周大钧　周　军　赵桂新　姜　燕　姜雪松

高悟实　徐　鑫　唐为方　董云峰

中国石油大学出版社

内 容 简 介

C++是近年来国内外广泛使用的高级计算机语言，它既支持面向过程的程序设计，也支持面向对象的程序设计，并已经成为编程人员最广泛使用的工具。学好C++，对深入理解程序设计、软件应用开发有着重要的意义。本书是结合多年的教学实践经验，并参阅了国内外有关C++的教材，分析了初学者对程序设计语言的认识规律而编写的，适合用作大学计算机专业和非计算机专业的程序设计基础课程教材，也可供读者自学使用。

本书的起点较低，读者可不具备C语言的基础，甚至未学过其他语言。本书共包括11章，前8章重点介绍C++语言中的面向过程的程序设计，第9、11章重点介绍C++语言的面向对象的程序设计，第10章介绍了C++语言中对文件的操作。

本书是依据ANSI C++标准进行介绍并编写的，能使读者养成良好的编程习惯。

图书在版编目(CIP)数据

C++程序设计教程/郭爱章主编. —2版. —东营：中国石油大学出版社，2009.7

ISBN 978-7-5636-2819-3

Ⅰ. C… Ⅱ. 郭… 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(2009)第104320号

书　　名：C++程序设计教程
主　　编：郭爱章

责任编辑：刘　静
封面设计：刘泽延

出 版 者：中国石油大学出版社(山东 东营　邮编 257061)
网　　址：http://www.uppbook.com.cn
电子信箱：cbs2006@163.com
印 刷 者：东营市新华印刷厂
发 行 者：中国石油大学出版社(电话 0546—8391810)
开　　本：185×260　印张：16.5　字数：417千字
版　　次：2009年7月第2版第1次印刷
定　　价：23.80元

PREFACE 前言

C++程序设计教程

随着计算机科学技术的迅速发展，所有的计算机语言也在不断地发展和完善。C++语言是近年来发展、推广速度最为迅速的一门程序设计语言，它的组成简洁、紧凑，使用方便、灵活，运算符和数据结构丰富，处理功能强，目标代码质量高，既具有高级语言的一般特点，又具有汇编语言对硬件和二进制位操作的特殊功能。

结构化程序设计早已被公认为是一种优秀的程序设计方法，并在各软件开发领域广泛应用，而C++语言正是适合于结构化程序设计的一门语言。它的模块化结构强，可移植性好，很适合编写大型软件。面向对象程序设计方法是近几年来发展迅速且比较新颖的程序设计方法，而C++语言是面向对象程序设计方法应用较广、功能较强的语言，它在C语言的基础上进行了扩充，加进了面向对象的设计思想。读者有了C语言的基础，可以较快地理解并掌握C++语言，掌握面向对象的设计思想和方法。

本教材前8章旨在使读者掌握结构化程序设计的思想方法，并学会用C++语言作为工具来具体实现和解决实际问题。因此，本教材前8章始终贯穿结构化思想，以使初学计算机语言的读者在学习与实践中逐步养成良好的习惯，建立良好的程序设计风格。第9、11章主要介绍如何用C++语言实现面向对象的程序设计，使学生逐步掌握面向对象程序设计的方法，并用C++语言进行面向对象的程序设计。

本书由郭爱章、王新刚负责统稿及定稿工作。全书共分11章。其中，第1章由李侃编写，第2章由董云峰编写，第3章由石扬编写，第4、5章由徐鑫编写，第6章由赵桂新编写，第7章由周大钧编写，第8章由唐为方、高悟实编写，第9章由周军、姜燕编写，第10章由姜雪松编写，第11章由杨振宇编写。全书由张维玉、杨清波校对。在教材编写过程中，耿玉水教授给予了很多宝贵建议，在此表示衷心

的感谢。

作者在编写过程中主观愿望是力求使概念叙述准确，方法条理清楚，并力争用最少的篇幅把最有用、最新颖的内容介绍给读者，但由于水平所限，书中难免存在缺点错误，恳请专家、同行和广大读者批评指正。

作　者

2009 年 7 月

CONTENTS 目录

C++程序设计教程

第1章 C++的初步知识

1.1 C++概述

1.1.1 C/C++语言发展史

大多数高级语言都是面向过程的语言，所以像操作系统这样的系统软件，在过去只能用汇编语言编写。但汇编语言对计算机硬件的依赖很大，使程序的可读性特别是可移植性很差，所以设计一种既具有高级语言优点，又具有低级语言特性的语言，对编写操作系统及其他系统软件具有重要意义。C语言就是在这种情况下诞生的。

C语言与UNIX操作系统是密不可分的，可以说，C语言就是为编写UNIX操作系统而设计并加以实现的。UNIX操作系统的源代码有90%以上是用C语言编写的，它的流行应归功于C语言。其实C语言并不是孤立产生的，它是在B(BCPL的第一个字母)语言的基础上发展起来的，而B语言又是在A(ALGOL)语言基础上发展而来的。ALGOL60是1960年出现的面向问题的高级语言，它与硬件相差甚远，不宜用来编写系统程序。1963年，英国剑桥大学推出了比较接近硬件的CPL(Combined Programming Language)语言，但该语言规模比较大，难以实现。1967年，剑桥大学的Matin Richards对CPL进行了简化，推出了BCPL(Basic Combined Programming Language)语言。1970年，美国贝尔实验室又在BCPL语言的基础上进一步简化推出了B语言，但B语言功能太简单。1972年，贝尔实验室的D. M. Ritchie又在B语言的基础上推出了C(BCPL的第二个字母)语言。C语言为UNIX系统而设计，又因UNIX系统的日益广泛使用而迅速得到推广，到20世纪80年代已风靡世界。

同其他高级语言一样，C语言自问世以来经过多次改进，版本较多。1978年，Brian W. Kernighan和Dennis M. Ritchie(合称K&R)合作编写了影响深远的名著《The C Programming Language》。此书介绍的C语言成为后来广泛使用的C语言版本的基础，被称为标准C。1983年，美国国家标准化协会(ANSI)又根据各种版本的C语言对C的扩充和发展重新制定了新的标准，称为ANSI C。1988年，K&R按国家标准又重写了《The C Programming Language》。很多C语言教材都是以ANSI C为基准编写的。目前广泛流行的各种C语言版本的编译系统其基本内容是相同的，但个别地方有所不同。如微机上的Turbo C和工作站上的Microsoft C等略有差异。

顾名思义，C++语言是C语言的扩充，它是贝尔实验室的Bjarne Stroustrup博士于20世纪80年代早期开始开发的一种通用程序设计语言。它将C语言作为子集，引入了Simula

67、ALGOL68 和 BCPL 语言中的一些面向对象程序设计的思想和概念，最初称为“带类的 C”，后取名为 C++。C++应用较广泛的有 Borland C++（For DOS 和 For Windows）和 Visual C++两大系列。

1.1.2 C++对 C 的“增强”

C++对 C 的“增强”表现在两个方面：

(1) 在原来面向过程的机制基础上，对 C 语言的功能做了不少扩充。

(2) 增加了面向对象的机制。

面向对象程序设计，是针对开发较大规模的程序而提出来的，目的是提高软件开发的效率。不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

学习 C++，既要会利用 C++进行面向过程的结构化程序设计，也要会利用 C++进行面向对象的程序设计。本书既介绍 C++在面向过程程序设计中的应用，也介绍 C++在面向对象程序设计中的应用。

1.2 最简单的程序设计

1.2.1 程序、程序设计、程序设计语言的概念

程序是什么呢？人们要让计算机解决一个问题时，需把解决这个问题的步骤通过一条条的指令告诉计算机。一般我们把人们事先准备好的，用来指挥计算机工作的描述工作步骤的指令序列称为程序。程序员设计编写程序的过程称为程序设计。用来编写程序的语言称为程序设计语言。

1. 机器语言

最初的阶段，人们直接使用计算机能识别的指令系统（称为机器语言）来编写程序。由于机器语言是二进制代码，人们编写或阅读程序都十分困难，又容易出错，且不同机器的指令系统也不同，所以很难进行交流（在一种机器上调试通过的程序不能到另外一种机器上运行）。虽然机器语言程序的执行效率很高，但花费在程序设计和调试程序上的时间太多，整个解决问题的效率就降低了。

2. 汇编语言

为了解决二进制代码编程带来的困难，人们采用了助忆码和符号地址来代替机器语言中的二进制指令代码和指令地址；然后通过一个人们预先设计好的叫做“汇编程序”的翻译程序一对一地翻译成机器语言程序，再让计算机执行。这种采用助忆码和符号地址的语言称为汇编语言。用汇编语言编写的程序的执行效率与机器语言程序的执行效率一样高，且其阅读性提高了，但由于汇编指令与机器指令之间是一对一的，同样不利于交流，所以汇编语言也是面向机器的语言。

3. 高级语言

随着计算机技术的发展，程序设计语言也在不断发展。为了脱离机器，为了非计算机专业人员的使用，人们开始用一种比较接近于自然语言（主要指英语）和数学语言的语言来编写程

序，这样的语言称为高级语言。高级语言脱离了计算机的具体指令系统，非计算机专业人员掌握起来比较容易，克服了面向机器语言的缺点，使得程序易读、易维护、易交流。高级语言发展很快，已达数百种之多。常用的高级语言有：

(1) FORTRAN 语言。诞生于 20 世纪 50 年代中期，是第一个算法语言，适合于科学和工程计算。

(2) BASIC 语言。诞生于 20 世纪 60 年代中后期，语言简单易学，是一种会话型语言，适合初学者学习。

(3) PASCAL 语言。诞生于 20 世纪 70 年代初，是一门结构化程序设计语言，适合于教学、科学计算、数据处理和系统软件开发，但随着 C 语言的出现，逐步被取代。

(4) C 语言。诞生于 20 世纪 70 年代初，80 年代开始风靡全世界，适合于系统软件、数值计算、数据处理。

(5) JAVA 语言。诞生于 20 世纪 90 年代，是一种新型的跨平台分布式程序设计语言，具有简单、安全、稳定、可移植性强等特点，将成为未来网络环境上的“世界语”。

一般地，把用高级语言或汇编语言编写的程序称为源程序。

1.2.2 C++程序举例

例 1.1 输出一行字符：“This is a C++ program.”。

程序如下：

```
#include <iostream>                    //包含头文件 iostream
using namespace std;                   //使用命名空间 std
int main( )                            //主函数
{                                      //函数体开始
cout<<"This is a C++ program.";        //一条完整语句的结束符
return 0;
}                                      //函数体结束
```

运行时会在屏幕上输出以下一行信息：

```
This is a C++ program.
```

main 代表“主函数”的名字。每一个 C++程序都必须有一个 main 函数。main 前面的 int 的作用是声明函数的类型为整型。程序第 6 行的作用是向操作系统返回一个零值。如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为－1。

函数体必须用大括号“{ }”括起来。本例中的主函数的函数体主要是一条以 cout 开头的语句。注意，C++中的所有语句最后都应当有一个分号。

程序的第 1 行“#include <iostream>”不是 C++的语句，而是 C++的一个预处理命令，它以“#”开头，以与 C++语句相区别，行的末尾没有分号。“#include <iostream>”是一个“包含命令”，它的作用是将文件 iostream 的内容包含到该命令所在的程序文件中，代替该命令行。文件 iostream 的作用是向程序提供输入或输出时所需要的一些信息。iostream 是 i-o-stream 3 个词的组合，从它的形式就可以知道它代表“输入输出流”的意思。由于这类文件都放在程序单元的开头，所以称为“头文件”(head file)。在程序进行编译时，先对所有的预处理命令进行处理，用头文件的具体内容代替 #include 命令行，然后再对该程序单元进行整体编译。

程序的第2行“using namespace std;”的意思是“使用命名空间 std”。C++标准库中的类和函数是在命名空间 std 中声明的，因此，程序中如果需要用到 C++标准库(此时需要用 #include 命令行)，就需要用“using namespace std;”进行声明，表示要用到命名空间 std 中的内容。

在初学 C++时，对本程序中的第1,2行可以不必深究，只需知道：如果程序有输入或输出时，必须使用“#include <iostream>”命令以提供必要的信息，同时要用“using namespace std;”使程序能够使用这些信息，否则程序编译时将出错。

例 1.2 求 a 和 b 两个数之和。

可以写出以下程序：

```
// 求两数之和                          (本行是注释行)
#include <iostream>                    //预处理命令
using namespace std;                   //使用命名空间 std
int main( )                            //主函数首部
{                                      //函数体开始
    int a,b,sum;                       //定义变量
    cin>>a>>b;                         //输入语句
    sum=a+b;                           //赋值语句
    cout<<"a+b="<<sum<<endl;           //输出语句
    return 0;                          //如程序正常结束,则向操作系统返回一个
                                       //零值
}                                      //函数结束
```

本程序的作用是求两个整数 a 和 b 之和 sum。第1行“//求两数之和”是一个注释行。C++规定在一行中如果出现“//”，则从它开始到本行末尾之间的全部内容都作为注释。

如果在运行时从键盘输入：

```
123  456↙
```

则输出为：

```
a+b=579
```

例 1.3 给两个数 x 和 y，求两数中的大者。

在本例中包含两个函数。

```
#include <iostream>                    //预处理命令
using namespace std;
int max(int x,int y)                   //定义 max 函数,函数值为整型,形式参数 x,y
                                       //为整型
{                                      //max 函数体开始
 int z;                                //变量声明,定义本函数中用到的变量 z 为整型
 if(x>y) z=x;                          //if 语句,如果 x>y,则将 x 的值赋给 z
 else z=y;                             //否则,将 y 的值赋给 z
 return(z);                            //将 z 的值返回,通过 max 带回调用处
}                                      //max 函数结束
int main( )                            //主函数
{                                      //主函数体开始
```

```
    int a,b,m;                          //变量声明
    cin>>a>>b;                          //输入变量 a 和 b 的值
    m=max(a,b);                         //调用 max 函数,将得到的值赋给 m
    cout<<"max="<<m<<endl;              //输出大数 m 的值
    return 0;                           //如程序正常结束,则向操作系统返回一个零值
}                                       //主函数结束
```

本程序包括两个函数:主函数 main 和被调用的函数 max。程序中第 3～9 行是 max 函数,它的作用是将 x 和 y 中较大者的值赋给变量 z。return 语句将 z 的值返回给主函数 main。返回值通过函数名 max 带回到 main 函数的调用处。主函数的 cin 语句的作用是输入 a 和 b 的值。main 函数的第 5 行为调用 max 函数,在调用时将实际参数 a 和 b 的值分别传送给 max 函数的形式参数 x 和 y。经过执行 max 函数得到一个返回值,把这个赋值给变量 m,然后通过 cout 语句输出 m 的值。

程序运行情况如下:

```
18  25↙                             (输入 18 和 25 给 a 和 b)
max=25                              (输出 m 的值)
```

注意:输入的两个数据间用一个或多个空格间隔,不能以逗号或其他符号间隔。

在上面的程序中,函数出现在 main 函数之前,因此在 main 函数中调用 max 函数时,编译系统能识别出 max 函数是已定义的函数名。如果把两个函数的位置对换一下,即先写 main 函数,后写 max 函数,在编译 main 函数遇到 max 时,编译系统无法知道 max 代表什么含义,因而无法编译,按出错处理。

为了解决先写 main 函数,后写 max 函数,在编译 main 函数遇到 max 而无法编译,按出错处理的问题,在主函数中需要对被调用函数作声明。上面的程序可以改写如下:

```
#include <iostream>
using namespace std;
int main( )
{
 int max(int x,int y);                  //对 max 函数作声明
 int a,b,c;
 cin>>a>>b;
 c=max(a,b);                            //调用 max 函数
 cout<<"max="<<c<<endl;
 return 0;
}
int max(int x,int y)                    //定义 max 函数
{
 int z;
 if(x>y) z=x;
 else z=y;
 return(z);
}
```

细心的读者会发现程序第5行的函数声明与程序第12行max函数的首部基本相同，因此，我们就能够很容易地写出函数声明，只要在被调用函数的首部的末尾加一个分号，就成为对该函数的函数声明。函数声明的位置应当在函数调用之前。

下面举一个包含类(class)和对象(object)的C++程序，目的是使读者初步了解C++是怎样体现面向对象程序设计方法的。

例1.4 包含类的C++程序。

```
#include <iostream>                          // 预处理命令
using namespace std;
class Student                                // 声明一个类，类名为 Student
{
private:                                     // 以下为类中的私有部分
  int num;                                   // 私有变量 num
  int score;                                 // 私有变量 score
public:                                      // 以下为类中的公用部分
  void setdata( )                            // 定义公用函数 setdata
  { cin>>num;                                // 输入 num 的值
    cin>>score;                              // 输入 score 的值
  }
  void display( )                            // 定义公用函数 display
  { cout<<"num="<<num<<endl;                 // 输出 num 的值
    cout<<"score="<<score<<endl;             // 输出 score 的值
   };
};                                           // 类的声明结束
Student stud1,stud2;                         // 定义 stud1 和 stud2 为 Student 类的变
                                             // 量，称为对象
int main( )                                  // 主函数首部
{   stud1.setdata( );                        // 调用对象 stud1 的 setdata 函数
    stud2.setdata( );                        // 调用对象 stud2 的 setdata 函数
    stud1.display( );                        // 调用对象 stud1 的 display 函数
    stud2.display( );                        // 调用对象 stud2 的 display 函数
    return 0;
}
```

在一个类中包含两种成员：数据和函数，分别称为数据成员和成员函数。在C++中，把一组数据和有权调用这些数据的函数封装在一起，组成一种称为“类(class)”的数据结构。在上面的程序中，数据成员num、score和成员函数setdata、display组成了一个名为Student的“类”类型。成员函数是用来对数据成员进行操作的。也就是说，一个类是由一批数据以及对其操作的函数组成的。

类可以体现数据的封装性和信息隐蔽。在例1.4的程序中，在声明Student类时，把类中的数据和函数分为两大类：private(私有的)和public(公用的)。把全部数据(num、score)指定为私有的，把全部函数(setdata、display)指定为公用的。在大多数情况下，会把所有数据指定