HZ BOOKS
华章教育

高等学校计算机专业规划教材

C++程序设计教程

第3版

王珊珊 臧洌 张志航 编著

Introduction to Programming with C++ Third Edition

机械工业出版社
China Machine Press

高等学校计算机专业规划教材

C++程序设计教程

第3版

王珊珊 臧洌 张志航 编著

Introduction to Programming with C++ Third Edition

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C++ 程序设计教程 / 王珊珊，臧洌，张志航编著 . —3 版 . —北京：机械工业出版社，2016.10
（高等学校计算机专业规划教材）

ISBN 978-7-111-55253-6

I. C… II. ①王… ②臧… ③张… III. C 语言－程序设计－高等学校－教材 IV. TP312.8

中国版本图书馆 CIP 数据核字（2016）第 260783 号

本书包括两部分内容。第一部分为第 1～9 章，以 C++ 语言的基本语法为起点讲述面向过程的程序设计，内容包括基本数据类型、基本控制结构、函数、数组、结构体、指针和链表。第二部分为第 10～15 章，结合 C++ 语言的应用实例，讲述面向对象程序设计的基本概念，内容包括类和对象、继承和派生、多态、模板以及输入输出和文件操作。

本书适用于程序设计语言的初学者，也适用于大学本科理工类各专业学习 C++ 程序设计语言的学生，同时适用于自学 C++ 语言的读者。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：佘　洁　　责任校对：殷　虹
印　　刷：三河市宏图印务有限公司　　版　　次：2017 年 1 月第 3 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：24.25
书　　号：ISBN 978-7-111-55253-6　　定　　价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

前　言

编写背景

各大专院校工科类专业开设了程序设计语言类课程，尤其是电类专业普遍开设了C++程序设计课程，因此需要一本适用于初学者的教材。本书就是为了满足这个层次的读者需求而编写的。本书包含两个方面的内容：1）面向过程的程序设计，目的是让初学者掌握基本的程序设计知识。2）面向对象的程序设计，让初学者了解面向对象程序设计的基本概念，为今后学习以面向对象为基础的通用软件开发工具如Microsoft Visual C++、Delphi、C#.net和Visual Studio等打下坚实的基础。

2005年和2011年本教材分别出版了第1版和第2版，被国内十几所高校和培训机构所使用，并取得了良好的反响。本书在前两版的基础上，修正了部分不足，文字描述更准确；程序在Visual Studio 2013环境中运行，尽量做到符合C++11新标准，并对C++11新标准进行了一定的解释；对源程序例子进行了少量增删，并对源程序编辑格式做了调整，即采用C++标准命名空间的方式编写程序。调整前和调整后源程序书写格式如下：

```
#include <iostream.h>      //调整前程序书写格式，使用带.h的头文件
#include <math.h>
void main()                //主函数无返回值
{
    //…语句
}
#include <iostream>        //调整后程序书写格式，使用不带.h的头文件
#include <cmath>
using namespace std;       //使用C++标准命名空间
int main()                 //主函数的返回值类型为int
{
    //…语句
    return 0;
}
```

编写内容和教学要求

本书分两部分，第一部分包括第1～9章，结合C++语言的基本语法，介绍传统的面向过程的程序设计，内容包括C++语言基本数据类型、基本运算、基本输入输出、结构化流程控制语句、函数、编译预处理、数组、结构体、指针和链表等，基本上是传统的C程序设计语言的内容。第二部分包括第10～15章，以面向对象的三大特点为主线，讲述类和对象的基本概念，类的封装、继承和多态，以及函数模板和类模板，并讲述了输入输出流类体系、文件操作等内容。

本书作者的教学理念是注重程序设计算法的教学，注重对学生算法思路的逻辑训练，而不拘泥于语法要素的细枝末节。本书通俗易懂，配有大量针对各章的教学难点和重点以及各

种算法而设计的例题和习题[⊖]。在选择例题和习题时，尽量涵盖目前程序设计语言课程的各类算法。初学者阅读习题时，能够在教材的例题中找到相似的例子进行模仿，这样对初学者来说解题就不是一件非常困难的事情。除了进行理论教学和上机练习外，教师还可以根据实际情况选用适合不同层次学生的课程设计题目，以加强学生动手编制较大规模程序的能力。

本书第 1、2、9～14 章由王珊珊老师编写，第 5～8 章由臧洌老师编写，第 3、4、15 章由张志航老师编写，全书由王珊珊负责统稿。

本书的实验环境是 Visual Studio 2013，书中全部的例题和习题均在该环境中通过编译和运行。

本书配套的上机实验和课程设计教材为《C++ 语言程序设计上机实验及学习指导》，王珊珊、臧洌和张志航编著，2016 年 1 月由南京大学出版社出版。

本书可能会存在疏漏、不妥和错误之处，恳请专家和广大读者指教和商榷。

作者联系方式：

shshwang@nuaa.edu.cn（王珊珊）

zangliwen@nuaa.edu.cn（臧洌）

zzh20100118@qq.com（张志航）

作者

2016 年 8 月 20 日

于南京航空航天大学

⊖ 限于篇幅，习题请从华章网站（www.hzbook.com）下载。——编辑注

教学建议

教学内容	学习要点及教学要求	学时数 （理论＋上机）
第 1 章 C++ 概述	• 了解程序设计语言的发展历史 • 了解结构化程序设计和面向对象程序设计的概念 • 认识 C++ 语言程序的基本形式 • 掌握 C++ 语言程序开发的步骤	1＋2
第 2 章 数据类型、运算符和表达式	• 掌握保留字和标识符的概念 • 掌握 C++ 基本数据类型及常量和变量的说明、初始化方法 • 掌握 C++ 各类基本运算符及表达式的构成方法和运算规则 • 掌握当不同数据类型的运算量进行运算时，类型的自动转换和强制转换概念	4＋2
第 3 章 简单的输入输出	• 了解输入输出流对象 cin 和 cout • 掌握基本类型数据的输入输出方法及简单的输入输出格式控制方法	1＋2
第 4 章 C++ 的流程控制	• 了解算法的基本概念 • 掌握结构化程序设计的三种基本控制结构，即顺序、选择和循环结构，重点掌握用于实现三种控制结构的语句及其语法和语义 • 重点培养学生使用三种基本控制结构实现分支、穷举、迭代、递推等算法分析问题和解决问题的能力	6＋6
第 5 章 函　　数	• 掌握函数的定义和调用方法，培养学生使用函数进行模块化程序设计的思想 • 掌握函数的嵌套调用和递归调用的概念 • 掌握带默认值的函数、重载函数和内联函数的概念 • 掌握变量的作用域和存储类别 • 了解程序的多文件组织	5＋6
第 6 章 编译预处理	• 了解编译预处理的概念 • 掌握宏定义、文件包含的概念及使用 • 了解条件编译的概念	1＋1
第 7 章 数　　组	• 掌握一维数组和二维数组的定义、赋值及输入输出的方法 • 掌握一维数组和二维数组作为函数参数的方法 • 掌握字符数组和字符串的使用方法，掌握常用字符串处理函数的使用 • 掌握一维数组和二维数组的有关算法	8＋7

（续）

教学内容	学习要点及教学要求	学时数（理论＋上机）
第 8 章 结构体、共用体和枚举类型	• 掌握结构体类型及其变量的定义、输入输出方法 • 掌握结构体数组的概念 • 掌握结构体变量、结构体数组作为函数参数及返回值的方法 • 了解共用体类型 • 掌握枚举类型及其变量的定义和使用方法	2＋3
第 9 章 指针、引用和链表	• 掌握指针的基本概念及相关运算（* 和 & 运算） • 掌握指针变量作为函数参数的本质 • 掌握一维数组和二维数组的指针的使用方法 • 掌握字符串指针的使用方法 • 掌握指针数组和指向指针的指针变量的使用方法 • 掌握指向函数的指针和返回指针值的函数的使用方法 • 掌握引用作为函数参数的使用方法 • 了解 const 型变量和 const 型指针 • 掌握动态存储空间的申请和释放方法 • 掌握采用动态存储分配技术实现单向链表的基本算法	10＋10
第 10 章 类和对象	• 掌握类和对象的概念和定义方法，正确理解类成员的三种访问属性 • 掌握各类构造函数的定义方法，理解构造函数的作用 • 掌握析构函数的定义方法，理解析构函数的作用 • 掌握类的对象成员的初始化过程 • 了解内联函数和外联函数的意义 • 理解 this 指针的意义	5＋6
第 11 章 类和对象的其他特性	• 掌握类的静态成员的特性及其定义和使用方法 • 掌握类的友元的概念及其定义和使用方法 • 了解类的常数据成员和常成员函数	1＋2
第 12 章 继承和派生	• 掌握继承和派生的基本概念及其意义，了解单一继承和多重继承的概念 • 掌握三种继承方式，掌握基类的不同访问属性的成员分别在三种继承方式下在派生类中访问属性的变化规则 • 掌握在派生类中初始化对象成员和基类成员的方法 • 掌握派生过程中的冲突、支配规则和赋值兼容性 • 掌握虚基类的意义及其定义和使用方法	4＋4
第 13 章 多　态　性	• 理解多态性，掌握静态多态和动态多态的概念 • 掌握运算符重载的概念和实现方法 • 掌握利用虚函数实现动态多态的方法 • 掌握纯虚函数和抽象类的概念及使用方法	4＋5
第 14 章 输入输出流	• 了解输入输出流类体系 • 了解输入输出流类体系中有关成员函数的使用方法 • 掌握对用户新定义类的插入和提取运算符的重载 • 了解文本文件流和二进制文件流的区别 • 掌握文本文件流的打开、关闭及读写方法 • 了解文件顺序访问和随机访问的概念 • 了解输入输出流的出错处理	4＋4

（续）

教学内容	学习要点及教学要求	学时数（理论＋上机）
*第 15 章 模　板	• 掌握函数模板 • 掌握类模板	选讲
课程设计	采用面向对象的方法实现一个小型信息管理系统，核心数据结构可使用结构体数组或链表结构。类似选题如下： 1）图书库存记录管理 2）学生成绩管理	16

说明：

1）本教材主要针对本科层次的理工科院校各专业学生编写。建议教学分成以下三大模块：

①理论授课 56 学时：主要完成 C++ 程序设计课程内容的理论教学。上表中的理论教学部分含习题课。

②上机实验 60 学时：针对每一章基本教学内容，安排相应的上机编程实习，目的是巩固理论教学内容，从简单的上机编程题入手，为课程结束时完成课程设计打基础。

③课程设计 16 学时：完成一个小型信息管理系统。

2）标注“*”号的章节根据具体情况选讲或学生自学。

3）理论教学可使用本教材，上机实验和课程设计可使用《C++ 语言程序设计上机实验及学习指导》（王珊珊、臧洌、张志航编著，南京大学出版社，2016）。

目　录

第1章 C++概述

1.1 计算机语言与程序

人类语言是人与人之间交流信息的工具，而计算机语言是人与计算机之间交流信息的工具。用计算机解决问题时，人们必须首先将解决问题的方法和步骤按照一定的规则和序列用计算机语言描述出来，形成计算机程序，然后让计算机自动执行程序，完成相应功能，解决指定的问题。下面先介绍计算机语言与程序经历的3个发展阶段。

1.1.1 机器语言与程序

机器语言是第一代计算机语言。任何信息在计算机内部都是采用二进制代码表示的，指挥计算机完成一个基本操作的指令（称为机器指令）也是由二进制代码表示的。每一条机器指令的格式和含义都是计算机硬件设计者规定的，并按照这个规定制造硬件。一个计算机系统全部机器指令的总和称为指令系统，它就是机器语言。用机器语言编制的程序为如下形式：

```
0000 0100 0001 0010
0000 0100 1100 1010
0001 0010 1111 0000
1000 1010 0110 0001
...
```

每一行都是一条机器指令，代表一个具体的操作。机器语言程序能直接在计算机上运行，且运行速度快、效率高，但必须由专业人员编写。机器语言程序紧密依赖于硬件，程序的可移植性差。所谓移植，是指在一种计算机系统下编写的程序经过修改可以在另一种计算机系统中运行，并且运行结果一样。改动越少，可移植性越好；改动越多，可移植性越差。

1.1.2 汇编语言与程序

机器语言是由二进制代码构成的，难以记忆和读写，用它编写程序比较困难。于是计算机工作者发明了汇编语言，用来代替机器语言编写程序。汇编语言是一种符号语言，它用一个有意义的英文单词缩写来代替一条机器指令，如用ADD表示加法，用SUB表示减法。英文单词缩写被称为助记符，每一个助记符代表一条机器指令，所有指令的助记符集合就是汇编语言。用汇编语言编写的程序有如下形式：

```
MOV  AL  12D  // 表示将十进制数12送往累加器AL
SUB  AL  18D  // 表示从累加器AL中减去十进制数18
      ...
HLT           // 表示停止执行程序
```

汇编语言改善了程序的可读性和可记忆性，使编程者在编写程序时稍微轻松了一点。但是汇编语言程序不能在计算机中直接运行，必须把它翻译成相应的机器语言程序才能运行。将汇编语言程序翻译成机器语言程序的过程称为汇编。汇编过程是计算机运行汇编程序自动完成的，如图1-1所示。汇编语言是第二代计算机语言。

图1-1 汇编过程

1.1.3 高级语言与程序

机器语言和汇编语言都是面向机器的语言，统称为低级语言。它们受特定计算机指令系统的限制，通用性较差，一般只适用于专业人员。非专业人员若想学习使用低级语言编写程序比较困难，为解决这一问题，计算机工作者发明了高级程序设计语言，简称高级语言。高级语言是第三代计算机语言。高级语言用类似于人类自然语言和数学语言的方式描述问题、编写程序。例如，用 C++ 语言编写的程序片段如下：

```
int a, b, c;          // 定义变量a、b和c
cin >> a >> b;        // 输入变量a、b的值
c = a + b;            // 将变量a、b的值相加，结果赋给变量c
cout << c;            // 输出变量c的值
```

该程序片段的功能见每条语句后面的说明。用高级语言编写程序时，编程者不需要考虑具体的计算机硬件系统的内部结构，即不需要考虑计算机的指令系统，而只要告诉计算机“做什么”即可。至于计算机“怎么做”，即用什么机器指令去完成，不需要编程者考虑。

高级语言程序也无法在计算机中直接运行。若要运行高级语言程序，首先必须将它翻译成机器语言目标程序，这个翻译的过程称为编译，编译是由“编译程序”（也称为“编译器”）完成的。然后由“连接程序”将目标程序与系统提供的标准函数的库程序连接，生成可执行程序。可执行程序可以在计算机中运行。编译、连接过程如图 1-2 所示。“编译程序”和“连接程序”属于计算机系统软件。

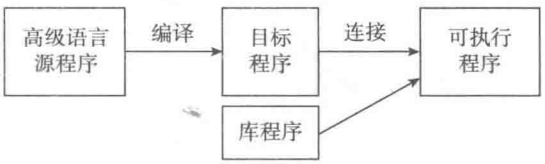

图 1-2 编译、连接过程

高级语言不仅易学易用、通用性强，而且具有良好的可移植性。如果想把高级语言程序移植到另一个计算机系统中，只需对源程序稍加改动甚至不改动，使用目标计算机系统的编译程序将高级语言源程序重新编译即可。不同的计算机系统具有不同的编译程序。

目前世界上有数百种高级语言，应用于不同领域，而 C++ 作为其中的优秀语言得到了广泛的使用。

1.2 从 C 到 C++

C 语言是美国贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的，1972 年在计算机 DEC PDP-11 上实现了最初的 C 语言。当时设计 C 语言是为了编写 UNIX 操作系统，UNIX 操作系统 90% 的代码由 C 语言编写，10% 的代码由汇编语言编写。随着 UNIX 操作系统的广泛使用，C 语言也被人们认识和接受。

C 语言在各种计算机上的快速推广导致出现了许多 C 语言版本。这些版本虽然是类似的，但通常不兼容。显然人们需要一个与开发平台和机器无关的标准的 C 语言版本。1989 年美国国家标准协会（American National Standard Institute，ANSI）制定了 C 语言的标准（ANSI C）。Brian Kernighan 和 Dennis Ritchie 编著的《The C Programming Language》（1988 年）介绍了 ANSI C 的全部内容，该书被称为 C 语言的圣经（C Bible）。

C 语言具有如下特点：1）语言简洁、紧凑，使用方便、灵活。C 语言只有 32 个关键字，程序书写形式自由。2）具有丰富的运算符和数据类型。3）可以直接访问内存地址、进行位操作，完成类似于汇编语言的操作，能够胜任开发系统软件的工作。因此，有时 C 语言也被称为“中级语言”，其意义是它既具有高级语言的特点，又具有低级语言的硬件直接操作特性。4）目标代码质量高，程序运行效率高。5）可移植性好，即可以很容易地将程序改写后运行在不同的计算机系统中。

但是，C语言也有如下局限性：1）数据类型检查机制较弱，这使得程序中的一些错误不能在编译时被发现；2）语言本身几乎没有支持代码重用的机制，因此，一个编程者精心设计的程序很难被其他程序所使用；3）当程序达到一定规模时，编程者很难控制程序的复杂性。

1980年贝尔实验室的Bjarne Stroustrup博士及其同事对C语言进行改进和扩充。最初的成果称为“带类的C”，而后称为“新C”。1983年由Rick Mascitti提议正式命名为C++（C Plus Plus）。在C语言中，运算符“++”的意义是对变量进行增值运算，因此C++的喻义是对C语言进行“增值”，扩充内容。1994年制定了ANSI C++草案。以后又经过不断完善，形成了目前的C++，C++仍然在不断地发展。

1.3 程序设计方法

C++语言的两个组成部分是过程性语言部分和“类”部分。过程性语言部分和C语言没有本质差别。“类”部分是C语言中没有的，它是面向对象程序设计的主体。要学好面向对象程序设计，首先必须具有过程性语言的基础。所以学习C++，首先必须学习其过程性语言部分，然后再学习“类”部分。

过程性语言部分采用的是结构化程序设计方法，“类”部分采用的是面向对象程序设计方法，程序设计方法正在从结构化方法向面向对象方法演变。C语言仅支持结构化程序设计，而C++语言既支持结构化程序设计也支持面向对象程序设计。

1.3.1 结构化程序设计方法

结构化程序设计的主要思想是：将任务按功能分解并逐步求精。将复杂的大型任务分解成若干模块，每个模块进一步划分成更小的、功能完整的子模块，继续划分直到得到原子模块，每个原子模块用一个过程或函数完成。一个过程或函数由多条可顺序执行的语句构成。编程者把数据与过程或函数分开存储。编写程序的主要技巧在于追踪函数的调用及返回过程，追踪在这个过程中数据发生了怎样的变化。结构化程序设计方法能够较好地分解并解决一些复杂任务。其主要缺点是，程序依赖于数据结构，当数据结构发生变化时，必须对过程或函数进行修改。另外，当开发一个新任务时，适用于旧任务的程序一般不能重复利用。从编程的角度来说需要重复投入，即重新开发程序。而基于可重用指导思想的面向对象程序设计方法能够较好地解决这一问题。

1.3.2 面向对象程序设计方法

面向对象程序设计（Object Oriented Programming，OOP）方法是近年来十分流行的一种程序设计方法，它试图用客观世界中描述事物的方法来描述一个程序要解决的问题。对象是客观世界中一个实际存在的事物，如一个具体的“人”就是一个对象。将“人”的共同属性抽象出来就可以构成“类”，如“人”类，它具备的静态属性有姓名、年龄、性别、身高和体重等，它同时具备的动态属性有学习、思考、走路、说话和吃饭等，一般将静态属性作为类的数据成员，而将动态属性作为类的执行代码。类是一个抽象的概念，而对象是类的具体实例，如“人”类的一个对象就是指一个具体的人。

面向对象程序设计的3个主要特性如下所示。

1）**封装性** 封装是实现信息隐蔽的基础。将描述对象的数据（静态属性）及对这些数据进行处理的程序代码（动态属性）有机地组成一个整体，同时对数据及代码的访问权限加以限制，这种特性称为封装。封装可以使对象内部的数据隐藏起来，在类外不能直接访问它们，而只能通过对象的公有执行代码接口来间接访问对象内部的数据。这样既可保护类中的数据成员，也可使编程者只关心该对象可完成的动作，而不必去关心其内部的数据及代码实现细节。

2）**继承性** 继承是软件重用的基础，它可以提高软件开发效率。通过继承可以在已有类的基

础之上扩充并产生一个新类。已有类称为基类或父类，新类称为派生类或子类。派生类除了继承基类的数据及代码之外，可以按需要增加数据和代码。基类的数据和代码在派生类中是可以直接使用的，即基类的代码可以在派生类中重复利用，这就是软件重用，它可以提高代码编写效率。

3）**多态性** 多态性是提高编程效率及提高编程灵活性的机制。多态分为静态多态和动态多态。

- **静态多态** 静态多态分为函数重载、运算符重载、函数模板和类模板。

  函数重载是指同名函数完成不同功能，一般用于完成类似功能，如两个同名函数abs()，分别可以用来求整型量和实型量的绝对值。如果没有函数重载机制，求整型量和实型量的绝对值就必须用两个不同的函数名来实现。函数重载减轻了编程时记忆多个完成类似功能的函数名的负担。

  运算符重载是将C++提供的基本运算符应用到新类的机制。例如，加号（+）运算符可以实现C++基本数据类型的整型量、实型量的相加等。对于用户新定义的类如“复数”类，通过运算符重载机制，可以使用加号实现两个复数对象的直接相加。

  函数模板是将结构相同而仅仅数据类型不同的多个函数进行数据类型虚拟化得到的函数的抽象描述。类模板是将结构相同而仅仅数据类型不同的多个类进行数据类型虚拟化得到的类的抽象描述。在调用函数模板和使用类模板定义对象时，编译器能够根据实际的数据类型以函数模板和类模板为基础自动生成含有具体类型的函数和类，以提高编程的自动化水平，即提高编程效率。

- **动态多态** 动态多态是指不同的对象在接收到相同的消息后，以不同的行为去应对。所谓消息是指对象接收到的需要执行某个“操作”的命令，操作是函数完成的。动态多态的实现机制是，在基类中定义完成某个操作的虚函数，在不同的派生类中重新定义完成这个“操作”的同名函数，不同派生类中的这些函数完成不同的工作，那么不同派生类对象接收到同样的“消息”时，就可以表现出不同的行为。

  例如，基类是一个抽象的“几何图形”，它具有“绘图”行为，但这个行为没有具体含义，因为并不知道具体绘制什么图形。从“几何图形”类派生出“三角形”类、“圆”类或“矩形”类，在派生类中“绘图”功能有具体含义，可重新定义“绘图”功能，实现具体图形的绘制。在基类中的虚拟共同行为“绘图”，在派生类中具有不同的实现行为。不同的派生类对象接收到“绘图”消息时，即产生了不同的行为。

  动态多态既提供了“消息”的统一入口，又实现了不同对象对同一消息的不同响应，提高了编程的灵活性。

1.4 简单的C++程序介绍

下面通过一个简单的例子来说明C++程序的基本结构。

例1.1 一个简单的C++程序。

```
/* ------------------------------------------------------------------
   Li0101.cpp  该程序用于求一个数的平方
   ------------------------------------------------------------------
*/
#include <iostream>
using namespace std;
int main(void)
{
    int num, square;                          // 定义整型变量num和square
    cout << "num=";                           // 输出提示信息num=
    cin >> num;                               // 输入一个数，赋给变量num
    square = num * num;                       // 计算num的平方，结果赋值给变量square
```

```
    cout << "num的平方为:" << square << '\n';    // 输出变量square的值，'\n'表示换行
    return 0;
}
```

上述C++程序由注释语句、编译预处理命令和主函数构成，下面做简单介绍，详细介绍见后续章节。

1. 注释语句

注释是对程序功能、算法思路、语句的作用等所做的说明。注释有两种形式，一种是在“/*”和“*/”之间加注释，此种形式的注释可以跨多行书写，如在开头对程序做总体说明；另一种是以两个斜杠“//”开头直到该行结束，在“//”和行末之间加注释，此种形式的注释只能在一行中书写。

2. 编译预处理命令

在本程序中“#include <iostream>”表示文件包含，即编译时将系统头文件iostream的内容插入本源程序头部。一般地，在程序中如果需要使用系统预先定义的标准函数、符号或对象，在程序的头部均要包含相应的头文件。

在本程序中包含头文件iostream，是因为在函数中使用了系统预先定义的、与数据的输入输出有关的流对象cin和cout。cin代表标准输入设备，通常指键盘。cout代表标准输出设备，通常指显示器屏幕。

3. 主函数main()

一个C++程序必须包含一个主函数main()，它是程序流程的主控函数，程序从主函数开始执行。main()前面的int表示该函数的返回值是int类型的数据。void表示函数无参数。函数体用花括号（{}）括起来。在函数体中，按照算法写出语句，完成功能。

经过编译连接，执行上述程序时，首先在屏幕上显示提示信息：

```
num=
```

等待用户输入一个整数，假如输入的是“8 < Enter >”（<Enter>表示<回车>），则程序在屏幕上显示：

```
num的平方为: 64
```

例1.2 一个由两个函数构成的C++程序，源程序名为Li0102.cpp。

```
#include <iostream>
using namespace std;
int sum(int x, int y)                    // A
{
    int z;
    z = x + y;
    return z;                            // B
}
int main(void)
{
    int a, b, c;                         // 定义变量a、b和c
    a = 3; b = 5;                        // 给变量a和b赋值
    c = sum(a, b);                       // C，调用函数sum()求a与b之和，结果赋给变量c
    cout << c << '\n';                   // 输出变量c的值
    return 0;
}
```

本程序由两个函数组成。程序从主函数main()开始，当执行到C行时发生函数调用，将