



普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbook of Computer Science

# C++语言 程序设计教程

The C++ Programming Language

吕凤翥 编著

- 强调概念理解，引导读者思考
- 传授编程思想，注重编程实践
- 涵盖C语言基础，适用多层次读者



名家系列

人民邮电出版社  
POSTS & TELECOM PRESS



普通高等教育“十一五”国家级规划教材  
21世纪高等学校计算机规划教材  
21st Century University Planned Textbook of Computer Science

TP312/2744  
2008

# C++语言 程序设计教程

The C++ Programming Language

吕凤翥 编著



名家系列

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

C++语言程序设计教程 / 吕凤翥编著. —北京: 人民邮电出版社, 2008.4

21 世纪高等学校计算机规划教材. 普通高等教育“十一五”国家级规划教材

ISBN 978-7-115-16985-3

I. C… II. 吕… III. C 语言—程序设计—高等学校—教材 IV. TP312

## 内 容 提 要

本书系统介绍 C++语言的基础知识、基本语法和编程方法。重点讲述 C++语言面向对象的重要特征, 包括类和对象、继承性和派生类、多态性和虚函数等重要内容。同时, 还介绍 C++语言对 C 语言的继承和改进。

本书内容系统全面, 偏重应用; 通过例子详细讲述 C++语言具有的封装性、继承性和多态性, 并阐述使用 C++语言编程的方法与技巧。为方便教学, 本书每章后备有大量练习题和上机题。

本书适合作为高等院校“C++语言程序设计”课程的教材, 还可作为 C++语言的自学参考书。

普通高等教育“十一五”国家级规划教材

21 世纪高等学校计算机规划教材

## C++语言程序设计教程

◆ 编 著 吕凤翥

责任编辑 张孟玮

◆ 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>

北京市艺辉印刷有限公司

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 20.5

字数: 498 千字

2008 年 4 月第 1 版

印数: 1—3 000 册

2008 年 4 月北京第 1 次印刷

ISBN 978-7-115-16985-3/TP

定价: 35.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154



## 编者的话

为了满足广大读者渴望学习和掌握 C++ 语言编程的愿望,为从事计算机编程工作和深入学习计算机专业知识打下基础,作者在总结长期从事“C++ 语言程序设计”课程教学经验的基础上编写了此书。

本书具有下述特点。

首先,教学重点明确,语言简明,针对难点和疑点,加强了解释和分析。注重培养分析问题解决问题的能力。

其次,内容安排上强调边看书边思考,注重培养学生学习思考能力。本书的许多例题中都提出了思考问题,每章后边都从多方面给出练习题,引导学生去思考,去理解,去掌握学过的内容。

最后,注重实践。书中强调上机实践,每章后都附有上机指导,用来引导读者上机实践。

本教材共分 10 章。其中第 1 章至第 5 章主要复习 C 语言中讲过的一些重要内容,同时指出 C++ 语言对 C 语言的某些改进。这些改进对学习 C++ 语言很重要。第 6 章至第 10 章主要讲述 C++ 语言面向对象的特征,这是 C++ 语言中的重点,这部分内容是 C 语言中所没有的。学好这部分内容对学习其他面向对象语言也会有帮助。

关于讲授 C++ 语言的学时安排,这里提供一个参考意见,各院校可根据自己的情况进行调整。具体安排如下。

章 节	讲授学时数	上机学时数
第 1 章	4	2
第 2 章	6	3
第 3 章	4	2
第 4 章	4	2
第 5 章	4	2
第 6 章	8	4
第 7 章	8	4
第 8 章	6	3
第 9 章	6	3
第 10 章	6	3

说明:

(1) 按上述学时安排, 共讲授 56 学时, 上机 28 学时。

(2) C 语言基础较好的, 第 1 章至第 5 章每章可用 3 学时授课, 上机时间可用 2 至 3 学时。

本书所有的程序, 包括例题、练习题以及习题答案的程序都已在 Visual C++ 6.0 版本的编译系统下通过, 有关资料与电子教案放在人民邮电出版社网站 ([www.ptpress.com.cn](http://www.ptpress.com.cn)) 下载区中。

在本书的编写过程中得到许多同事和朋友的支持和帮助, 特别是吕涛、邵慧、商孟春等同志在程序录入、调试以及校对方面做了大量的工作, 在此一并表示感谢。

本书中若有错漏之处, 敬请读者批评指正。

吕凤翥

2007 年 8 月

于北京大学燕北园

学时	学时	章
2	4	第 1 章
2	6	第 2 章
2	4	第 3 章
2	4	第 4 章
2	4	第 5 章
4	8	第 6 章
4	8	第 7 章
2	6	第 8 章
2	6	第 9 章
2	6	第 10 章

# 目 录

第 1 章 C++语言概述	1
1.1 面向对象的概念	1
1.1.1 面向对象方法的由来	1
1.1.2 面向对象的基本概念	2
1.2 C++语言的特点	3
1.2.1 C++语言是面向对象的程序设计语言	3
1.2.2 C++语言继承了 C 语言	4
1.2.3 C++语言对 C 语言进行了改进	4
1.3 C++程序在结构上的特点	5
1.3.1 C++程序举例	5
1.3.2 C++程序结构上的特点	8
1.4 C++程序的实现	8
1.4.1 C++程序的编辑、编译和运行	8
1.4.2 C++程序实现举例	9
练习题 1	13
上机指导 1	16
第 2 章 变量和表达式	17
2.1 C++语言的字符集和单词	17
2.1.1 C++语言字符集	17
2.1.2 单词及其词法规则	17
2.2 常量	19
2.2.1 常量的种类	19
2.2.2 符号常量	23
2.3 变量	24
2.3.1 变量的三要素	24
2.3.2 变量的定义格式	26
2.3.3 变量的作用域	28
2.4 运算符和表达式	30

2.4.1	运算符的种类和功能	30
2.4.2	运算符的优先级和结合性	33
2.4.3	表达式的值和类型	34
2.4.4	表达式求值举例	35
2.5	类型转换	40
2.5.1	自动转换	40
2.5.2	强制转换	41
2.6	数组	41
2.6.1	数组的定义格式和数组元素的表示方法	41
2.6.2	数组的赋值	43
2.6.3	字符数组和字符串	47
	练习题 2	52
	上机指导 2	61
<b>第 3 章</b>	<b>语句和预处理</b>	<b>64</b>
3.1	表达式语句和复合语句	64
3.1.1	表达式语句和空语句	64
3.1.2	复合语句和分程序	65
3.2	选择语句	65
3.2.1	条件语句	65
3.2.2	开关语句	68
3.3	循环语句	70
3.3.1	while 循环语句	70
3.3.2	do-while 循环语句	72
3.3.3	for 循环语句	73
3.3.4	多重循环	75
3.4	转向语句	77
3.4.1	goto 语句	77
3.4.2	break 语句	78
3.4.3	continue 语句	79
3.5	类型定义语句	79
3.6	预处理功能	81
3.6.1	文件包含命令	81
3.6.2	宏定义命令	82
	练习题 3	84
	上机指导 3	89
<b>第 4 章</b>	<b>指针和引用</b>	<b>90</b>
4.1	指针和指针的定义格式	90



4.1.1	什么是指针	90
4.1.2	指针的定义格式	91
4.2	指针的运算	92
4.2.1	指针的赋值运算和增值运算	92
4.2.2	指针的比较运算和相减运算	94
4.2.3	指针运算和地址运算	95
4.3	指针和数组	96
4.3.1	指针可表示数组元素	96
4.3.2	字符指针和字符串处理函数	99
4.3.3	指向数组的指针和指针数组	102
4.4	引用	105
4.4.1	引用和引用的创建方法	105
4.4.2	引用和指针	107
	练习题 4	108
	上机指导 4	113
<b>第 5 章</b>	<b>函数</b>	<b>114</b>
5.1	函数的定义和说明	114
5.1.1	函数的定义	114
5.1.2	函数的说明方法	116
5.2	函数的参数和返回值	117
5.2.1	函数的参数	117
5.2.2	设置函数参数的默认值	118
5.2.3	函数返回值的实现	120
5.3	函数的调用	121
5.3.1	函数的传值调用	121
5.3.2	函数的引用调用	123
5.3.3	函数的嵌套调用	124
5.4	指针和引用作函数参数和返回值	126
5.4.1	指针作函数参数和返回值	126
5.4.2	引用作函数参数和返回值	127
5.5	重载函数和内联函数	129
5.5.1	重载函数	129
5.5.2	内联函数	132
5.6	函数的存储类	133
5.6.1	外部函数	133
5.6.2	内部函数	135
	练习题 5	136
	上机指导 5	142



第6章 类和对象(一)	143
6.1 类的定义格式和对象的定义方法	143
6.1.1 类的定义格式	143
6.1.2 对象的定义方法	146
6.1.3 对象成员表示	147
6.2 对象的初始化	149
6.2.1 构造函数的功能、种类和特点	149
6.2.2 析构函数的功能和特点	150
6.3 数据成员的类型和成员函数的特性	153
6.3.1 类中数据成员类型的规定	153
6.3.2 成员函数的特性	154
6.4 静态成员	157
6.4.1 静态数据成员	157
6.4.2 静态成员函数	159
6.5 常成员	161
6.5.1 常数据成员	161
6.5.2 常成员函数	162
6.6 友元函数和友元类	164
6.6.1 友元函数	164
6.6.2 友元类	165
6.7 类型转换	167
6.7.1 类型的隐含转换	167
6.7.2 一般数据类型转换为类类型	167
6.7.3 类类型转换为一般数据类型	168
练习题6	170
上机指导6	177
第7章 类和对象(二)	179
7.1 对象指针和对象引用	179
7.1.1 对象指针	179
7.1.2 this 指针	183
7.1.3 对象引用	185
7.2 对象数组和对象指针数组	186
7.2.1 对象数组	186
7.2.2 对象指针数组	188
7.2.3 指向对象数组的指针	189
7.3 子对象和堆对象	191
7.3.1 子对象	191

7.3.2	堆对象	193
7.4	类的作用域和对象的生存期	197
7.4.1	类的作用域	197
7.4.2	对象的生存期	198
7.5	结构的应用	200
7.5.1	结构变量和结构数组	200
7.5.2	使用 struct 定义类	204
	练习题 7	205
	上机指导 7	213
<b>第 8 章</b>	<b>继承性和派生类</b>	<b>216</b>
8.1	继承的概念	216
8.1.1	基类和派生类	216
8.1.2	单重继承和多重继承	217
8.1.3	派生类的定义格式	218
8.1.4	派生类成员的访问权限	219
8.2	单重继承	224
8.2.1	单重继承派生类的构造函数和析构函数	224
8.2.2	子类型和赋值兼容规则	231
8.3	多重继承	234
8.3.1	多重继承派生类的构造函数和析构函数	234
8.3.2	多重继承的二义性	237
	练习题 8	241
	上机指导 8	247
<b>第 9 章</b>	<b>多态性和虚函数</b>	<b>250</b>
9.1	运算符重载	250
9.1.1	运算符重载的概念	250
9.1.2	运算符重载的两种方法	251
9.1.3	运算符重载举例	256
9.2	静态联编和动态联编	260
9.2.1	联编的概念	260
9.2.2	虚函数	264
9.2.3	动态联编	266
9.2.4	虚析构函数	269
9.3	纯虚函数和抽象类	271
9.3.1	纯虚函数	271
9.3.2	抽象类	272
	练习题 9	273

上机指导 9	281
<b>第 10 章 C++语言文件的输入/输出操作</b>	<b>282</b>
10.1 I/O 流类库概述	282
10.1.1 输入/输出流	282
10.1.2 I/O 流类库的主要功能	283
10.2 标准文件的输入/输出操作	283
10.2.1 屏幕输出操作	284
10.2.2 键盘输入操作	287
10.3 格式输出操作	291
10.3.1 使用流对象的成员函数进行格式输出	291
10.3.2 使用控制符进行格式输出	294
10.4 磁盘文件的操作	295
10.4.1 打开文件和关闭文件操作	296
10.4.2 文件的输入/输出操作	297
10.4.3 随机文件操作	301
练习题 10	303
上机指导 10	308
<b>附录 A 字符的 ASCII 码表</b>	<b>309</b>
<b>附录 B Microsoft Visual C++ 6.0 集成开发工具简介</b>	<b>310</b>

# 第 1 章 C++语言概述

本章介绍面向对象的概念和 C++语言的特点。通过 C++程序实例，分析 C++程序在结构上的特点及书写程序应注意的事项。本章还介绍使用 Microsoft Visual C++ 6.0 编译系统实现 C++程序的方法。

## C 1.1 面向对象的概念

C++语言是一种面向对象的程序设计语言，在介绍 C++语言之前，先介绍一下有关面向对象的概念，有助于对 C++语言的理解和掌握；通过对 C++语言的学习又会进一步加深对面向对象方法的认识。

### 1.1.1 面向对象方法的由来

面向对象方法是人们开发软件的一种方法，这种方法的提出是软件研究人员对软件开发在认识上的一次飞跃，它是软件开发史上的一个里程碑。它标志着软件开发产业进入一个新阶段。

在面向对象方法出现之前，人们采用的是面向过程的方法。面向过程方法是一种传统的求解问题的方法。该方法将整个待解决问题按其功能划分为若干个相对独立的小问题，每个小问题还可以按其功能划分为若干个相对独立的更小的问题，依此类推，直到将所划分的小问题可以容易用程序模块实现为止。在面向过程的程序设计中，每个程序模块具有相对独立的功能，由小模块组成大模块，最后组成一个完整的程序。整个程序的功能是通过模块之间相互调用来实现的。这种面向过程的方法具有很多的弊病。第一，该方法将数据和数据处理过程分离成为相互独立的实体，当数据结构一旦发生变化时，所有相关的处理过程都要进行相应的修改，因此，程序代码的可重用性较差。第二，该方法对于图形界面的应用开发起来比较困难，而图形界面越来越被人们广泛使用。第三，面向过程的程序设计中，模块之间有较大的依赖性，这对调试程序和修改程序带来一定的难度。

面向对象方法是求解问题的一种新方法，它把求解问题中客观存在的事物看做各自不同的对象，这符合人们习惯的思维方式，再把具有相同特性的一些对象归属为一个类，每个类是对该类对象的抽象描述。对象之间可以进行通信。类之间可以有继承关系，函数和运算符



可以重载，这样可以提高程序的可重用性，便于软件开发和维护。

总之，面向对象方法是计算机科学发展的要求。随着人们对信息的需求量越来越大，软件开发的规模也越来越大，对软件可靠性和代码的重用性的要求越来越高。这时，面向过程的方法使得分析结果不能直接映射待解决的问题，并且分析和设计的不一致给在编程、调试、维护等诸方面造成不便和困难。在这种情况下，面向对象方法应运而生。由于面向对象方法具有封装、继承和多态等特性，与面向过程方法相比，它较好地克服了在编程、调试和维护等方面的不便和困难，提供了代码的重用率，使得软件开发变得更为容易和方便。

## 1.1.2 面向对象的基本概念

面向对象是一种由对象、类、封装、继承和多态性等概念来构造系统的软件开发方法。这些新的概念描述了面向对象这种新方法。

### 1. 对象

对象是现实世界中客观存在的某种事物，它可以是有形的，也可以是无形的。对象是一种相对独立的实体，它具有静态特性和动态特性，通常通过一组数据来描述对象的静态特性，使用一组行为或功能来表示对象的动态特性。

对象是系统中用来描述客观事物的一个实体，它是软件系统的基本构成单位。对象是由一组属性和一组行为构成的。属性是描述对象的静态特性的数据项，行为是描述对象动态特性的操作。

### 2. 类

类是人们对于客观事物的高度抽象。抽象是忽略事物的非本质特性，只抓住与当前相关的特性，从而找出其共性，把具有共同特性的事物划分为一类，得到一个抽象的概念。例如，在生活中经常遇到的抽象出来的概念有桌子、房屋、汽车和足球等。

面向对象方法中的类是一种类型，它是具有相同属性和行为的对象的集合。类是具有相同属性和行为的若干对象的模板。类为属于该类的全部对象提供了抽象的描述，这种描述包括了属性和行为两大部分。类与对象的关系就像模具和铸件的关系。某个类的对象又称为该类的一个实例。

### 3. 封装

封装是指把对象的属性和行为结合成一个独立的单位，又称为封装体。对象的属性通常用一组数据项来表示，对象的行为又称为服务，通常用方法或函数来表示。封装体具有独立性和隐藏性。独立性表现在封装体内所包含的属性和行为形成了一个不可分割的独立单位；隐藏性表现在封装体内的有些成员在封装体外是不可见的，这部分成员被隐藏了，具有一定的安全性。一个封装体与外部联系只能通过有限的接口。

### 4. 继承

继承是面向对象方法提高重用性的重要措施，继承表现了特殊类与一般类之间的关系。

当特殊类包含了一般类的所有属性和行为，并且特殊类还可以有自己的属性和行为时，称作特殊类继承了一般类。特殊类又称为派生类，一般类称为基类。

继承的重要性就在于它大大地简化了对于客观事物的描述。例如，已经描述汽车这个类属性和行为，由于如小轿车是汽车类的特殊类，它具有汽车类的所有属性和行为，在描述小轿车类时，只需描述小轿车本身的属性和行为，而汽车类的属性和行为不必再重复了，因为小轿车类继承了汽车类。

## 5. 多态性

多态性指的是一种行为对应着多种不同的实现。在同一个类中，同一种行为可对应着不同的实现，例如，函数重载和运算符重载都属于多态性。同一种行为在一般类和它的各个特殊类中可以具有不同的实现，例如，动态联编是属于这类多态性。在一般类中说明了一种求几何图形面积的行为，这种行为不具有具体含义，因为并没有确定具体的几何图形，又定义一些特殊类，如“三角形”、“圆形”、“正方形”、“矩形”、“梯形”等，它们都继承了一般类。在不同的特殊类中都继承了一般类的“求面积”行为，可以根据具体的不同几何图形使用求面积的公式，重新定义“求面积”行为的不同实现，使之分别实现求“三角形”、“圆形”、“正方形”、“矩形”和“梯形”等面积的功能。这就是面向对象方法的重要的多态性。

## 1.2 C++语言的特点

C++语言是20世纪80年代初期由美国贝尔实验室的科研人员提出的，它是一种继承了C语言的面向对象的程序设计语言。

### 1.2.1 C++语言是面向对象的程序设计语言

C++语言支持面向对象的程序设计，主要表现在它支持面向对象方法中的3个主要特性。

#### 1. 支持封装性

C++语言允许使用类和对象。类是支持数据封装的工具，对象是数据封装的实现。类中成员有不同的访问权限。类中的私有成员仅由该类体内的成员函数访问，因此，私有成员具有隐藏性，类体外是不可见的。类中的公有成员是类体与外界的一个接口，类体外面的函数可以访问类体中的公有成员。类中还有一种保护成员，它具有公有成员和私有成员的双重特性，它具体使用在类的继承中。

#### 2. 支持继承性

C++语言支持面向对象方法中的继承性，它不仅支持单重继承，而且支持多重继承。继承性给C++语言编程带来了方便，增强了程序的扩展性和可重用性，提高了软件开发的效率。继承是两个类之间的关系，基类和派生类是继承中的重要概念。派生类继承了基类中的所有成员，并且还可以定义自身的新成员，继承实现了抽象和共享的机制。继承和封装是衡量一

种语言是否是面向对象的程序设计语言的两个重要标准。C++语言支持封装，又支持继承，因此，可以断定 C++语言是面向对象的程序设计语言。

### 3. 支持多态性

多态性是在继承性基础上的面向对象方法中的重要特性之一。C++语言支持多态性主要表现在如下两个方面。

① 支持函数重载和运算符重载。重载是指一个函数名可以有多种实现，即同一个行为对应不同实现，这便是多态性。

② 支持动态联编。动态联编反映了基类和派生类中同名函数的多态性。动态联编是在公有继承的前提下，通过虚函数来实现的。动态联编虽然没有静态联编运行效率高，但它可以透过高度抽象，提高程序的灵活性和扩充性。

## 1.2.2 C++语言继承了 C 语言

C++语言与 C 语言兼容，C 语言是 C++语言的一个子集。C 语言的词法、语法和其他规则都可以用到 C++语言中。例如，C 语言中的类型、运算符和表达式在 C++语言中都可以使用；C 语言中的语句也是 C++语言的语句；C 语言中的函数定义和调用也适用于 C++语言，只是 C++语言对此稍有改进和扩充；C 语言中的预处理命令也可用于 C++语言，只是宏定义命令在 C++语言中较少使用；C 语言中的构造类型，如数组、结构和联合在 C++语言中也是适用的，只不过 C++语言中较多是使用类类型；C 语言中的指针在 C++语言中也可使用，但是 C++语言中引入了引用概念，在某些情况下减少了指针的使用；还有 C 语言中作用域的规则，存储类的规定等在 C++语言中也都适用。

由于 C++语言继承了 C 语言，使得已经掌握了 C 语言的人们学习 C++语言比较容易，这也是 C++语言得以广泛使用的原因之一。已经学会了 C 语言的人们学习 C++语言时，要做到一种思维方式的转变，即从面向过程的思维方式转变到面向对象的思维方式，没有这个转变是学不好 C++语言的。

由于 C++语言继承了 C 语言，因此，C++语言仍旧具有 C 语言的简练明了的风格，同时还不得不保留某些 C 语言的面向过程的特性。所以，有人说 C++语言是一种不完全的面向对象的程序设计语言。在 C++程序中，允许类体外的函数存在，这便是保留面向过程的特征。

## 1.2.3 C++语言对 C 语言进行了改进

C++语言虽然保留了 C 语言的风格和特点，但又针对 C 语言的某些不足做了较大的改进。改进后的 C++语言与 C 语言相比，在数据类型方面更加严格了，使用更加方便了。

下面简单扼要地列举一些 C++语言对 C 语言的改进内容，更详细的介绍可参见本书第 2 章至第 5 章。

① C++语言中规定所有函数定义时必须指出数据类型，不允许默认数据类型。无返回值的函数使用 `void` 进行说明，返回值为整型的函数使用 `int` 进行说明。

② C++语言规定函数说明必须使用原型说明，不得用简单说明。



③ C++语言规定凡是从高类型向低类型转换时都需加强制转换。

④ C++语言中符号常量建议使用 `const` 关键字来定义,这种方法可以指出常量类型,使用简单宏定义命令定义符号常量没有类型说明。

⑤ C++语言中引进了内联函数,建议使用内联函数取代带参数的宏定义命令,这也是增加了对参数的类型说明。

⑥ C++语言允许设置函数参数的默认值,提高了程序运行的效率。

⑦ C++语言引进了函数重载和运算符重载的规则,为编程带来了方便。

⑧ C++语言引进了引用概念,使用引用作函数的参数和返回值,比使用指针作函数参数和返回值更加方便,并且二者具有相同的特点。这就使得 C++程序中减少了对指针的使用,避免由于指针使用不当造成的麻烦。

⑨ C++语言提供了与 C 语言不同的 I/O 流类库,方便了输入/输出操作。

⑩ C++语言为方便操作还采取了其他措施。例如,使用运算符 `new` 和 `delete` 代替函数进行动态存储分配;增添了行注释符 (`//`),为行注释信息提供了方便;取消了 C 语言中在函数体和分程序中说明语句必须放在执行语句的前边的规定等。

## 1.3 C++程序在结构上的特点

本节介绍 C++语言编写的程序在结构上的特点和在书写上应注意的事项。为此,首先列举两个 C++语言的程序,从这两个程序实例中分析 C++程序的特点。

### 1.3.1 C++程序举例

**【例 1.1】**从键盘上输入两个 `int` 型数,编程求这两个 `int` 型数之和。

程序内容如下:

```
#include <iostream.h>

int add(int ,int);

void main()
{
    int a,b;
    cout<<"Enter a b: ";
    cin>>a>>b;
    int c=add(a,b);
    cout<<"a+b="<<c<<endl;
}

int add(int x,int y)
{
    return x+y;
}
```



运行该程序后，显示下述提示信息：

```
Enter a b:
```

这时，在键盘上输入 18\_36 后，输出显示下述结果：

```
a+b=54
```

其中，使用 “\_” 表示空格符，使用 “↵” 表示按 Enter 键。

程序分析：

例 1.1 是一个 C++ 语言的程序，初看上去该程序从结构形式和书写规则上与 C 语言程序很相似。该程序由一个文件组成，该文件有两个函数，一个是主函数 main()，另一个是被调用函数 add()。

再仔细看会发现该程序与 C 语言程序有如下的区别。

① 该程序开头包含了 iostream.h 文件，该文件中包含了 C++ 语言的输入/输出操作中的相关内容。例如，该程序中使用的插入符 (<<) 和提取符 (>>) 都被重载定义在 iostream.h 文件中，还有 endl 也定义在该头文件中，它与换行符 ('\n') 功能相同。

② 该程序中第 2 条语句是函数说明语句，这里使用的是原型说明，不仅要说明函数名字和类型，还要说明函数参数的个数和类型。

③ 主函数 main() 和被调用函数 add() 在定义时都加了类型说明符 void 和 int，这是不可省略的。

④ 在主函数中出现了如下所示的输出和输入语句：

```
cout <<"Enter a b: ";
cout <<"a+b="<<c<<endl;
cin>>a>>b;
```

其中，前边两条是输出语句，后边一条是输入语句。

下面结合该例程序介绍 C++ 语言的标准文件的输出语句和输入语句。关于输入/输出操作的内容详见本书第 10 章。由于标准文件输入/输出操作在一开始的程序中就不可避免地要出现，因此在这里仅就使用重载运算符进行标准文件的输入/输出介绍如下。

● 使用插入符进行输出操作的格式如下：

```
<操作数 1> << <操作数 2> << <操作数 3> ...
```

其中，<操作数 1> 是输出流对象名，C++ 语言规定标准输出设备屏幕的对象名为 cout。<操作数 2>，<操作数 3>... 是待输出的表达式。这里，<< 是被系统重载的左移运算符，重载后的功能是将右操作数的值输出到左操作数指定对象上。下列输出语句：

```
cout <<"Enter a b:";
```

将字符串常量 “Enter a b:” 输出到屏幕的当前光标处。

下列输出语句：

```
cout <<"a+b="<<c<<endl;
```

先将字符串常量 “a+b=” 输出到屏幕的当前光标处，接着再将变量 c 的值输出到屏幕的当前光标处，最后，再输出一个换行符到屏幕的当前光标处。于是，输出结果为：

```
a+b=54
```

● 使用提取符进行输入操作的格式如下：

```
<操作数 1> >> <操作数 2> >> <操作数 3> ...
```