

WUTP



普通高等学校计算机科学与技术专业新编系列教材

# C++ for Programmers

# C++程序设计教程



闵联营 何克右 主编

```
#include <stdio.h>
#include <stdio.h>
void main()
void main()
{
    void swap(int *ptr1,int *ptr2);
    void swap(int *ptr1,int *ptr2);
    int x,y,*ptr1,*ptr2;
    int x,y,*ptr1,*ptr2;
    printf("input x,y:");scanf("%d,%d",&x,&y);
    printf("input x,y:");scanf("%d,%d",&x,&y);
    printf("%d\t%d\n",x,y);ptr1=&x;ptr2=&y;
    printf("%d\t%d\n",x,y);ptr1=&x;ptr2=&y;
    if(x<y)
    if(x<y)
    swap(ptr1,ptr2);
    swap(ptr1,ptr2);
    printf("%d\t%d\n",x,y);
    printf("%d\t%d\n",x,y);
}
void swap(int *ptr1,int *ptr2)
void swap(int *ptr1,int *ptr2)
```

武汉理工大学出版社  
Wuhan University of Technology Press

TP312C  
104



普通高等学校计算机科学与技术专业新编系列教材

# C++ for Programmers

# C++ 程序设计教程

主 编 闵联营 何克右

武汉理工大学出版社

Wuhan University of Technology Press

## 内 容 提 要

本书是一本全面、系统地介绍 C++ 程序设计的教程。全书共分十二章,从内容上组织为 4 个部分。第 1 部分是 C++ 的概述;第 2 部分介绍 C++ 面向过程的程序设计方法,主要包括 C++ 的基本数据类型、程序的控制语句、数组和结构体、函数、指针和引用等内容;第 3 部分作为一个过渡,用一个具体的实例探讨了程序设计从过程抽象到数据抽象的转变;第 4 部分讲述 C++ 语言面向对象的程序设计方法,包括类和对象、继承和派生、多态和虚函数、模板、流类库等内容。

本书语言简洁、通俗易懂,注重理论与实践相结合。全书内容全面,实例丰富。书中所有例题均在 Visual C++ 6.0 上运行通过。

本书可以作为高等院校计算机专业和相关专业程序设计课程的教材和计算机专业面向对象程序设计教材,也可以作为全国计算机等级考试(二级 C++)的培训教材和参考书,还可供各类程序设计培训班学员和 C++ 语言自学者参考。

### 图书在版编目(CIP)数据

C++ 程序设计教程/闵联营,何克右主编. —武汉:武汉理工大学出版社, 2005. 7

普通高等学校计算机科学与技术专业(本科)新编系列教材

ISBN 7-5629-2261-6

I. C… II. ① 闵… ② 何… III. C 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 088542 号

出版发行:武汉理工大学出版社(武汉市武昌珞狮路 122 号 邮政编码:430070)

HTTP://www.techbook.com.cn

经 销 者:各地新华书店

印 刷 者:安陆市鼎鑫印务有限责任公司

开 本:787×960 1/16

印 张:25.5

字 数:500 千字

版 次:2005 年 7 月第 1 版

印 次:2005 年 7 月第 1 次印刷

印 数:1~3000 册

定 价:35.00 元

本书如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。本社购书热线电话:(027)87397097 87394412

**普 通 高 等 学 校**  
**计算机科学与技术专业新编系列教材**  
**编 审 委 员 会**

**顾问:**

卢锡城 周祖德 何炎祥 卢正鼎 曾建潮  
熊前兴

**主任委员:**

严新平 钟 珞 雷绍锋

**副主任委员:**

李陶深 鞠时光 段隆振 王忠勇 胡学钢  
李仁发 张常年 郑玉美 程学先 张翠芳  
孙成林

**委员:(以姓氏笔画为序)**

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| 王 浩 | 王景中 | 刘任任 | 江定汉 | 朱 勇 |
| 宋中山 | 汤 惟 | 李长河 | 李临生 | 李跃新 |
| 李腊元 | 李朝纯 | 肖俊武 | 邱桃荣 | 张江陵 |
| 张继福 | 张端金 | 张增芳 | 陈和平 | 陈祖爵 |
| 邵平凡 | 金 聪 | 杨开英 | 赵文静 | 赵跃华 |
| 周双娥 | 周经野 | 钟 诚 | 姚振坚 | 徐东平 |
| 黄求根 | 郭庆平 | 郭 骏 | 袁 捷 | 龚自康 |
| 崔尚森 | 蒋天发 | 詹永照 | 蔡启先 | 蔡瑞英 |
| 谭同德 | 熊盛武 | 薛胜军 |     |     |

**秘书长:**田道全

**总责任编辑:**段 超 徐秋林

## 出版说明

当今世界已经跨入了信息时代,计算机科学与技术正在迅猛发展。尤其是以计算机为核心的信息技术正在改变整个社会的生产方式、生活方式和学习方式,推动整个人类社会进入信息化社会。为了顺应时代潮流,适应计算机专业调整及深化教学改革的要求,充分考虑到不同层次高校的教学现状,满足广大高校的教学需求,武汉理工大学出版社经过广泛调研,与国内近 30 所高等院校的计算机专家进行探讨,决定组织编写“普通高等学校计算机科学与技术专业新编系列教材”。

我们在组织编写本套新编系列教材时,以培养现代化高级人才为重任,以提高学生综合素质、培养学生应用能力和创新能力为目的,以面向现代化、面向世界、面向未来为准绳,注重系列教材的特色和实用性,反映最新的教学与科研成果,体现本专业的时代特征。同时,面对教育改革的需要、人才的需要和社会的需要,在编写本教材时,借鉴、学习国外一流大学的先进教学体系,结合国内的实际需要,吸取具有先进性、实用性和权威性的国外教材的精华,以更好地促进国内教材改革顺利进行。从时代和国际竞争要求的高度来思考,为打造一套高起点、高水平、高质量的系列教材而努力。

本套教材具有以下特色:

**与时俱进,内容科学先进**——充分体现计算机学科知识更新快的特点,及时更新知识,确保教材处于学科前沿,以拓宽学生知识面,培养学生的创新能力。

**紧跟教学改革步伐,体现教学改革的阶段性成果**——符合全国高校计算机专业教学指导委员会、中国计算机学会教育委员会制订的“计算机学科教学计划 2000”的内容要求。

**实现立体化出版,适应教育方式的变革**——本套教材努力使用和推广现代化的教学手段,凡有条件的课程都准备组织编写、制作和出版配合教材使用的实验、习题、课件、电子教案及相应的程序设计素材库。

本套教材首批 25 种预定在 2003 年秋季全部出齐。我们的编审者、出版者决不敢稍有懈怠,一定高度重视,兢兢业业,按最高的质量标准工作。教材建设是我们共同的事业和追求,也是我们共同的责任和义务,我们诚恳地希望大家积极选用本套教材,并在使用过程中给我们多提意见和建议,以便我们不断修订、完善全套教材。

武汉理工大学出版社

2002 年 10 月

# 前 言

C++是当今最为流行和实用的一门高级程序设计语言,它不仅保留了传统的结构化程序设计方法,而且对面向对象程序设计方法提供了完整的支持。

目前,国内高校普遍开设了面向对象程序设计之类的课程,一些高校将C++语言作为程序设计语言课程的首选语言,需要有一本全面系统介绍C++程序设计语言的教材。为此,我们在近几年教学实践的基础上编写了这本书。

本书全面、系统地介绍了C++面向过程和面向对象的程序设计方法。

全书共分12章,从内容上组织为4个部分。第1章是C++的概述,介绍了C++的发展和特点、程序的基本框架以及程序的一般开发过程等。第2章至第6章介绍C++面向过程的程序设计方法,主要介绍了C++的基本数据类型、运算符和表达式、程序的控制语句、数组和结构体、函数、指针和引用等。第7章作为一个过渡,用一个具体的实例探讨了程序设计从过程抽象到数据抽象的转变,简单介绍了面向对象的基本特征和概念。第9章至第12章讲述C++面向对象程序设计的方法,主要包括类和对象、继承和派生、多态和虚函数、模板、流类库等内容。

由于C++语言是从C语言发展而来,为了与C兼容,C++允许继续使用C的句法。我们认为,既然学习C++,就应遵循C++标准,故本书所有的句法和程序都是依据C++标准介绍的。在教学实践中我们感到学生对面向对象的基本特征和概念的理解有一定的困难,因此本书专门用一章的篇幅介绍了程序设计从过程抽象到数据抽象的转变,逐步引入面向对象的概念,帮助读者对面向对象程序设计进行理解。对于学过C语言而需要学习面向对象程序设计的读者,可以直接从第7章开始学习。

本书语言简洁、通俗易懂,注重理论与实践相结合。全书内容全面,实例丰富。书中所有例题均在Visual C++ 6.0上运行通过。

本书第1、2、5、8章由闵联营编写,第3、4、6、9章由何克右编写,第7、10、11、12章分别由谭新明、刘传文、郭小兵、马保才编写。闵联营统编全书。

由于水平有限,书中难免有不足之处,恳请读者批评指正。

编 者

2005年5月



# 目 录

|                                       |      |
|---------------------------------------|------|
| 1 C++概述 .....                         | (1)  |
| 1.1 C++的发展和特点 .....                   | (1)  |
| 1.1.1 C++的发展 .....                    | (1)  |
| 1.1.2 C++的特点 .....                    | (2)  |
| 1.2 C++程序的基本框架 .....                  | (3)  |
| 1.2.1 最简单的C++程序 .....                 | (3)  |
| 1.2.2 结构化程序设计框架 .....                 | (5)  |
| 1.2.3 面向对象程序设计框架 .....                | (7)  |
| 1.3 C++程序的开发过程 .....                  | (9)  |
| 1.3.1 C++程序开发的一般过程 .....              | (9)  |
| 1.3.2 用 Visual C++开发 C++应用程序的步骤 ..... | (11) |
| 本章小结 .....                            | (14) |
| 习题与思考题 .....                          | (15) |
| 2 C++数据类型 .....                       | (16) |
| 2.1 基本数据类型 .....                      | (16) |
| 2.1.1 数据在计算机中的存储 .....                | (16) |
| 2.1.2 数据类型修饰符 .....                   | (17) |
| 2.2 常量和变量 .....                       | (18) |
| 2.2.1 常量 .....                        | (18) |
| 2.2.2 变量 .....                        | (21) |
| 2.3 运算符和表达式 .....                     | (23) |
| 2.3.1 算术运算符和算术表达式 .....               | (24) |
| 2.3.2 赋值运算符和赋值表达式 .....               | (25) |
| 2.3.3 关系运算和逻辑运算 .....                 | (26) |
| 2.3.4 条件运算符 .....                     | (28) |
| 2.3.5 逗号运算符 .....                     | (28) |
| 2.3.6 位运算符 .....                      | (29) |

---

|       |                           |      |
|-------|---------------------------|------|
| 2.3.7 | sizeof 运算符 .....          | (30) |
| 2.3.8 | 运算符优先级与结合性 .....          | (30) |
| 2.3.9 | 数据类型转换 .....              | (32) |
| 2.4   | 数据的输入和输出 .....            | (32) |
| 2.4.1 | 用 cout 进行输出 .....         | (32) |
| 2.4.2 | 用 cin 进行输入 .....          | (34) |
|       | 本章小结 .....                | (35) |
|       | 习题与思考题 .....              | (37) |
| 3     | 程序控制语句 .....              | (39) |
| 3.1   | 选择语句 .....                | (39) |
| 3.1.1 | if...else 语句 .....        | (39) |
| 3.1.2 | switch 语句 .....           | (45) |
| 3.2   | 循环语句 .....                | (48) |
| 3.2.1 | while 语句 .....            | (48) |
| 3.2.2 | do 语句 .....               | (49) |
| 3.2.3 | for 语句 .....              | (50) |
| 3.2.4 | 循环的嵌套 .....               | (52) |
| 3.2.5 | break 和 continue 语句 ..... | (54) |
| 3.2.6 | goto 语句 .....             | (55) |
|       | 本章小结 .....                | (56) |
|       | 习题与思考题 .....              | (57) |
| 4     | 数组和结构体 .....              | (61) |
| 4.1   | 数组 .....                  | (61) |
| 4.1.1 | 数组的定义和存储 .....            | (61) |
| 4.1.2 | 数组元素的引用 .....             | (64) |
| 4.1.3 | 数组的初始化 .....              | (65) |
| 4.1.4 | 数组的应用 .....               | (68) |
| 4.2   | 结构体 .....                 | (74) |
| 4.2.1 | 结构体概述 .....               | (74) |
| 4.2.2 | 结构体类型变量的定义和引用 .....       | (76) |
| 4.2.3 | 结构体数组 .....               | (78) |
|       | 本章小结 .....                | (79) |
|       | 习题与思考题 .....              | (80) |
| 5     | 函数 .....                  | (82) |
| 5.1   | 函数概述 .....                | (82) |

---

|                    |              |
|--------------------|--------------|
| 5.2 函数的定义和调用       | (84)         |
| 5.2.1 函数定义的一般形式    | (84)         |
| 5.2.2 函数的调用        | (87)         |
| 5.2.3 函数原型         | (89)         |
| 5.2.4 函数的参数传递      | (90)         |
| 5.3 函数的嵌套和递归       | (99)         |
| 5.3.1 函数的嵌套调用      | (99)         |
| 5.3.2 函数的递归调用      | (101)        |
| 5.4 内联函数           | (105)        |
| 5.5 函数重载           | (106)        |
| 5.6 变量的作用域和存储期     | (109)        |
| 5.6.1 局部变量和全局变量    | (109)        |
| 5.6.2 变量的存储期       | (112)        |
| 5.6.3 存储类别小结       | (116)        |
| 5.7 预处理命令          | (118)        |
| 5.7.1 宏定义命令        | (118)        |
| 5.7.2 “文件包含”处理     | (119)        |
| 5.7.3 条件编译命令       | (120)        |
| 本章小结               | (122)        |
| 习题与思考题             | (123)        |
| <b>6 指针和引用</b>     | <b>(129)</b> |
| 6.1 指针             | (129)        |
| 6.1.1 地址和指针的概念     | (129)        |
| 6.1.2 指针的定义和使用     | (131)        |
| 6.1.3 指针与数组        | (138)        |
| 6.1.4 指针与结构体       | (150)        |
| 6.1.5 指针与函数        | (158)        |
| 6.1.6 const 修饰符和指针 | (162)        |
| 6.2 引用             | (163)        |
| 6.2.1 引用的概念        | (163)        |
| 6.2.2 用 const 限定引用 | (165)        |
| 6.2.3 引用作函数参数      | (165)        |
| 6.2.4 引用返回值        | (168)        |
| 本章小结               | (169)        |
| 习题与思考题             | (170)        |

|                                  |              |
|----------------------------------|--------------|
| <b>7 从面向过程到面向对象 .....</b>        | <b>(173)</b> |
| 7.1 从过程抽象到数据抽象 .....             | (173)        |
| 7.1.1 集合的实现与使用 .....             | (174)        |
| 7.1.2 将集合的实现与使用分开 .....          | (178)        |
| 7.1.3 将集合用链表实现 .....             | (182)        |
| 7.1.4 将集合的数据表示和操作封装在一起 .....     | (185)        |
| 7.2 面向对象程序设计的基本概念和特征 .....       | (190)        |
| 7.2.1 面向对象程序设计的基本概念 .....        | (191)        |
| 7.2.2 面向对象程序设计的三大特征 .....        | (194)        |
| 7.2.3 从面向过程的程序设计到面向对象的程序设计 ..... | (201)        |
| 本章小结 .....                       | (204)        |
| 习题与思考题 .....                     | (204)        |
| <b>8 类和对象 .....</b>              | <b>(205)</b> |
| 8.1 类 .....                      | (205)        |
| 8.1.1 类的定义 .....                 | (205)        |
| 8.1.2 类的数据成员 .....               | (207)        |
| 8.1.3 类的成员函数 .....               | (208)        |
| 8.1.4 类成员的访问控制 .....             | (211)        |
| 8.2 对象 .....                     | (212)        |
| 8.2.1 对象的定义和使用 .....             | (212)        |
| 8.2.2 this 指针 .....              | (216)        |
| 8.3 构造函数和析构函数 .....              | (218)        |
| 8.3.1 构造函数 .....                 | (218)        |
| 8.3.2 缺省参数的构造函数 .....            | (220)        |
| 8.3.3 重载构造函数 .....               | (222)        |
| 8.3.4 拷贝构造函数 .....               | (224)        |
| 8.3.5 析构函数 .....                 | (226)        |
| 8.3.6 成员对象的构造和析构 .....           | (227)        |
| 8.4 对象数组和对象指针 .....              | (231)        |
| 8.4.1 对象数组 .....                 | (231)        |
| 8.4.2 对象指针 .....                 | (234)        |
| 8.5 静态成员 .....                   | (240)        |
| 8.5.1 静态数据成员 .....               | (240)        |
| 8.5.2 静态成员函数 .....               | (242)        |
| 8.6 友元 .....                     | (244)        |

---

|              |                      |       |
|--------------|----------------------|-------|
| 8.6.1        | 友元函数 .....           | (244) |
| 8.6.2        | 友元类 .....            | (248) |
| 8.7          | 常类型 .....            | (248) |
| 8.7.1        | 常对象 .....            | (249) |
| 8.7.2        | 用 const 修饰的类成员 ..... | (249) |
| 本章小结 .....   |                      | (252) |
| 习题与思考题 ..... |                      | (254) |
| 9            | 继承与派生 .....          | (264) |
| 9.1          | 派生类 .....            | (264) |
| 9.1.1        | 派生类的定义 .....         | (264) |
| 9.1.2        | 派生类生成过程 .....        | (266) |
| 9.1.3        | 派生类的构造函数和析构函数 .....  | (269) |
| 9.2          | 访问控制 .....           | (273) |
| 9.2.1        | 公有继承 .....           | (273) |
| 9.2.2        | 私有继承 .....           | (274) |
| 9.2.3        | 保护继承 .....           | (276) |
| 9.3          | 虚基类 .....            | (279) |
| 9.3.1        | 多继承中的二义性问题 .....     | (279) |
| 9.3.2        | 虚基类的概念 .....         | (284) |
| 9.3.3        | 虚基类的构造函数 .....       | (286) |
| 本章小结 .....   |                      | (291) |
| 习题与思考题 ..... |                      | (292) |
| 10           | 多态 .....             | (297) |
| 10.1         | 概述 .....             | (297) |
| 10.1.1       | 静态联编 .....           | (298) |
| 10.1.2       | 动态联编 .....           | (300) |
| 10.2         | 运算符重载 .....          | (303) |
| 10.2.1       | 运算符重载的方法和规则 .....    | (303) |
| 10.2.2       | 运算符重载为成员函数 .....     | (308) |
| 10.2.3       | 运算符重载为友元函数 .....     | (312) |
| 10.3         | 虚函数 .....            | (315) |
| 10.3.1       | 虚函数的定义及使用 .....      | (315) |
| 10.3.2       | 纯虚函数和抽象类 .....       | (323) |
| 10.3.3       | 虚函数的应用示例 .....       | (329) |
| 本章小结 .....   |                      | (335) |

---

|                           |       |
|---------------------------|-------|
| 习题与思考题                    | (336) |
| <b>11 模板</b>              | (339) |
| 11.1 模板的概念                | (339) |
| 11.2 函数模板和模板函数            | (341) |
| 11.2.1 函数模板的定义和模板函数的生成    | (341) |
| 11.2.2 函数模板的使用            | (343) |
| 11.2.3 函数模板与模板函数的关系       | (347) |
| 11.3 类模板和模板类              | (347) |
| 11.3.1 类模板的定义             | (347) |
| 11.3.2 类模板的使用             | (349) |
| 11.3.3 类模板的派生             | (354) |
| 本章小结                      | (360) |
| 习题与思考题                    | (361) |
| <b>12 C++的 I/O 流类库</b>    | (363) |
| 12.1 C++的流及流类库            | (363) |
| 12.1.1 C++的流              | (363) |
| 12.1.2 流类库                | (364) |
| 12.2 格式化 I/O              | (367) |
| 12.2.1 用 ios 类的成员函数进行格式控制 | (367) |
| 12.2.2 使用 I/O 操纵符进行格式控制   | (371) |
| 12.3 重载 I/O 运算符           | (374) |
| 12.3.1 重载输出运算符“<<”        | (374) |
| 12.3.2 重载输入运算符“>>”        | (376) |
| 12.4 文件流                  | (378) |
| 12.4.1 文件的打开和关闭           | (379) |
| 12.4.2 文件的读写              | (382) |
| 本章小结                      | (391) |
| 习题与思考题                    | (392) |
| <b>参考文献</b>               | (395) |



# 1 C++概述

## 本章提要

C++是一门广泛应用的程序设计语言,它在C语言的基础上扩展了面向对象程序设计的特点。本章介绍C++的发展和特点,C++程序的基本结构及C++程序的开发过程。

## 1.1 C++的发展和特点

### 1.1.1 C++的发展

C++语言是一种面向过程和面向对象都适用的混合型语言,它是在C语言的基础上逐步发展和完善起来的,而C语言则是在吸收了其他语言的一些优点后逐步发展为实用性很强的语言。

C语言是在B语言的基础上发展起来的,它的根源可以追溯到ALGOL60。ALGOL60是1960年出现的一种面向过程的高级语言,它离硬件比较远,不适合编写系统程序。1963年英国剑桥大学推出了旨在更接近硬件的CPL(Combined Programming Language)语言。由于CPL语言规模较大且难以实现,因此,1967年英国剑桥大学的Martin Richards对CPL语言作了简化,推出了BCPL(Basic Combined Programming Language)语言。同年美国贝尔实验室的K. Thompson在BCPL的基础上作了进一步简化,设计出了简单而又接近硬件的B语言,并用B语言写了一个Unix操作系统。由于B语言过于简单,功能有限,且目标代码运行速度慢,美国贝尔实验室的D. M. Ritchie于1972年在B语言的基础上设计出了C语言,并在第二年和K. Thompson合作用C语言重写

了 Unix 操作系统。现在的 Unix 操作系统就是在此基础上发展起来的。

后来, C 语言又经过多次改进, 在 1977 年出现了不依赖于具体机器的 C 语言编译程序, 并开始移植到非 Unix 环境中, 逐步成为一种独立的程序设计语言。1988 年美国国家标准协会 ANSI 在继承和发展 D. M. Ritchie 和 K. Thompson 的 C 语言的基础上进行了标准化, 产生了 ANSI C。目前所有的 C 语言编译系统都是以它为基础的。

C 语言和其他高级语言一样, 从程序设计角度看是“面向过程”的, 从语言的处理方法上看是“面向问题”的。20 世纪 80 年代提出面向对象(Object Oriented)的概念以来, 也产生了许多面向对象的程序设计语言, 如 Smalltalk、Actor 等纯面向对象的程序设计语言, C++、Delphi 等混合型的面向对象的程序设计语言, 即在传统的过程型程序设计语言中加入了面向对象的成分。随着 C 语言的推广应用, C 语言的一些缺陷和不足也开始表露出来, 如对数据类型的检查机制较为薄弱, 缺少支持代码重用的结构等。

1980 年美国贝尔实验室的 Bjarne Stroustrup 等对 C 语言进行了改进和扩充, 并引入了类的概念, 加入了运算符重载、引用、虚函数等功能, 使 C++ 成为目前最为流行的一种混合型程序设计语言。

### 1.1.2 C++的特点

C++ 语言具有下列主要特点:

① C++ 完全兼容 C, 具有 C 语言的“简捷、紧凑, 运算符丰富, 可直接访问机器的物理地址, 使用灵活方便, 程序书写形式自由”的特点。大多数的 C 语言代码略作修改或不修改就可在 C++ 的集成环境下运行。

② C++ 作为一种面向对象的程序设计语言, 它使程序的各个模块间更具独立性, 程序的可读性更好, 代码结构更加合理, 对设计和编制大型软件更为方便。

③ 用 C++ 设计的程序扩充性强。

C++ 在面向对象的程序设计语言方面获得主导地位, 很大程度上是由于它继承了 C 语言的主要特征, 再加上它加强了数据类型的检查机制, 引入了类的概念等。从 C++ 的名字可以看出它是 C 的超集。C++ 既可用于面向过程的结构化程序设计, 又可用于面向对象的程序设计, 是一种功能强大的混合型的程序设计语言。

## 1.2 C++程序的基本框架

要用C++进行程序设计,需要了解C++程序的基本框架,由于C++语言既支持结构化程序设计,又支持面向对象程序设计,所以它同时具有结构化程序设计和面向对象程序设计两种基本框架。本节我们先介绍一个最简单的C++程序,然后分别介绍C++程序的两种基本框架。

### 1.2.1 最简单的C++程序

下面先介绍一个简单的C++程序,说明C++程序的基本组成结构。

**【例 1.1】** 一个最简单的C++程序。

```
// This is a simple C++ program.  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout<<"This is a simple C++ program."<<endl;  
    return 0;  
}
```

这个程序运行时会在屏幕上输出以下一行信息:

This is a simple C++ program.

上述程序虽然只有8行,但却包含了一个C++程序要具备的几个基本组成部分,下面我们对它进行逐行解释:

第一行: // This is a simple C++ program.

这是一个注释行。注释是程序编写者为读者作的一种说明,有助于提高程序的可读性。C++中提供了“行”和“块”两种注释方法。行注释的内容从双斜杠“//”开始到本行末尾结束,本程序中用到的就是行注释;块注释的内容从符号“/\*”开始到符号“\*/”结束,例如:

```
/* This is a simple C++ program. */
```

和

```
/* This is a example.
```

```
It is a simple C++ program. */
```

都是块注释。

第二行: #include <iostream>

这是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。本行预处理命令“#include <iostream>”的作用是将头文件“iostream”的内容加入到程序中。“iostream”是一个 C++ 标准头文件,其中定义了一些输入输出流对象。在程序进行编译时,先对所有的预处理命令进行处理,将头文件的具体内容代替 #include 命令行,然后再对该程序进行整体编译。

第三行:using namespace std;

这是命名空间声明。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 C++ 标准库,就需要用“using namespace std;”作声明,表示要用到命名空间 std 中的内容。

第四行:int main()

这是主函数的声明。每一个 C++ 程序都必须有一个 main 函数,它是 C++ 程序开始执行的入口。无论主函数处于程序中的什么位置,其中的代码总是最先被执行。main 前面的 int 的作用是声明函数的类型为整型(标准 C++ 规定 main 函数必须声明为 int 型,即此主函数带回一个整型的函数值)。

第五、八行:{ }

在主函数 main 的声明之后用花括号“{ }”括起来的是函数主体部分。即函数体由“{ }”括起来。

第六行:cout<<“This is a simple C++ program.”<<endl;

这是一条 C++ 语句。它完成了此程序的主要功能,向屏幕上输出一行字符串。cout 是 C++ 中的标准输出流对象。cout 在标准头文件 iostream 中被声明,因此要使用它就必须先包含此文件(见程序的第 2 行)。“<<”是插入运算符,与 cout 配合使用,在本程序中它的作用是将右侧的用双引号括起来的字符串“This is a simple C++ program.”和 endl(换行)插入到输出流 cout 中。C++ 系统将输出流 cout 的内容输出到系统指定的设备(一般为显示器)中。本行末尾的分号“;”表示这条语句的结束。在 C++ 中所有的语句都要以分号结束。

第七行:return 0;

这一行是 return 语句,它的功能是使主函数 main 结束并将整数 0 返回给运行此程序的操作系统。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。

上面这个程序是按照 ANSI C++ 规定的语法编写的。由于 C++ 是从 C 语言发展而来的,为了与 C 兼容,C++ 保留了 C 语言中的一些规定。其中之一是头文件的形式,在 C 语言中头文件用 .h 作为后缀,如 stdio.h、math.h 等。在 C++ 发展初期,为了与 C 语言兼容,许多 C++ 编译系统保留头文件以 .h 为后缀的用法,如 iostream.h。但后来 ANSI C++ 建议头文件不带后缀 .h,如用 iostream、cmath 等作为头文件名。但为了使原来编写的 C++ 程序能够运行,