

新世纪
高等职业教育规划教材

C++ 语言程序设计

曹向东 张国海 编



 **机械工业出版社**
CHINA MACHINE PRESS



新世纪高等职业教育规划教材

C++语言程序设计

曹向东 张国海 编

刘 斌 主 审



机械工业出版社

本书以 Visual C++ 作为实验开发平台, 通过大量精选的案例, 系统介绍了 C++ 面向对象程序设计的精髓, 并在此基础上进一步深入而全面地讲解了 Windows 应用程序设计的基本原理和实用开发方法。全书共分为三部分。第一部分为 C++ 语言程序设计基础, 全面而深入地介绍了面向对象程序设计的基本思想、类与对象、继承与派生以及多态性等核心内容; 第二部分以基于 API 的应用程序实例为主, 较为系统地讲述了 Windows 编程的基础知识; 第三部分通过深入分析大量颇具代表性的 MFC 程序案例, 介绍了当今非常热门的 MFC 可视化编程方法。

本书通俗易懂, 重点突出, 偏重应用, 不仅可以作为高职高专院校学生的教材, 还可以供 Visual C++ 应用程序开发的爱好者参考。

图书在版编目 (CIP) 数据

C++ 语言程序设计/曹向东, 张国海编.—北京: 机械工业出版社, 2003.6
新世纪高等职业教育规划教材
ISBN 7-111-12121-X

I. C… II. ①曹… ②张… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 036081 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 贡克勤 王小东 责任校对: 刘秀芝

责任编辑: 苏颖杰 版式设计: 霍永明

封面设计: 张 静 责任印制: 闫 焱

北京瑞德印刷有限公司印刷 • 新华书店北京发行所发行

2003 年 6 月第 1 版第 1 次印刷

1000mm×1400mm B5 • 6.75 印张 • 260 千字

定价: 17.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

新世纪高等职业教育规划教材编审委员会

主任委员	李维东	广东白云职业技术学院	常务副院长
副主任委员	陈周钦	广东交通职业技术学院	院长
	石令明	广西柳州职业技术学院	院长
	蔡昌荣	广州民航职业技术学院	副院长
	覃洪斌	广西职业技术学院	副院长
	姚和芳	湖南铁道职业技术学院	副院长
	刘国生	番禺职业技术学院	副院长
	韩雪清	机械工业出版社教材编辑室	副主任
委	郑伟光	广东机电职业技术学院	院长
员	张尔利	广西交通职业技术学院	院长
	谈向群	无锡职业技术学院	副院长
	陈大路	温州职业技术学院理工学区	主任
	邹 宁	广西机电职业技术学院	副院长
	修德明	济源职业技术学院	副院长
	管 平	浙江机电职业技术学院	副院长
	韦荣敏	广西柳州市交通学校	校长
	田玉柯	遵义航天工业学校	校长
	黄秀猛	厦门市工业学校	校长
	韩书平	新乡市高级技工学校	校长
	张毓琴	广东白云职业技术学院	兼委员会 秘书

编写说明

20 世纪 90 年代以来,我国高职高专教育为社会主义现代化建设事业培养了大批急需的各类专门人才,提高了劳动者的素质,对于建设社会主义的精神文明,促进社会进步和经济发展起到了重要作用。中共中央、国务院《关于深化教育改革,全面推进素质教育的决定》指出:“要大力发展高等职业教育”,教育部在《教育部关于加强高职高专教育人才培养工作的意见》中明确指出:“高职高专教育是我国高等教育的重要组成部分,培养拥护党的基本路线,适应生产、建设、服务第一线需要的,德、智、体、美等方面全面发展的高等技术应用性专门人才;学生应在具有必备的基础理论知识和专门知识的基础上,重点掌握从事本专业领域实际工作的基本能力和基本技能。”加入 WTO 以后,我国将面临人才资源的全球竞争,其中包括研究开发型人才的竞争,也包括专业技能型优秀人才的竞争。高等职业教育要适应我国现代化建设的需要,适应世界市场和国际竞争的需要,培养大批符合市场需求的、有熟练技能的高等技术应用性人才。

教材建设工作是整个高职高专教育教学工作中的重要环节,在贯彻国家教育教改精神、保证人才质量方面起着重要作用。改革开放以来,各地已出版了一批高职高专教材,但从整体上看,具有高职高专教育特点的教材极其匮乏,教材建设仍滞后于高职高专教育的发展需要。为此,根据目前高等职业教育发展的要求,机械工业出版社组织全国多所在高等职业教育办学有特色、在社会上影响较大的高职院校成立了“新世纪高等职业教育规划教材编审委员会”,选择教学经验丰富、实践能力强的骨干教师,组织、规划、编写了这套“新世纪高等职业教育规划教材”,教材首批四个系列 36 本(书目附后)。它凝聚着全体编审人员、编委会委员的大量心血,同时得到了各委员院校的大力支持,在此表示衷心感谢。

本套教材的作者队伍是经编审委员会严格遴选确定的,他们来自高等职业教育的第一线,教学经验丰富、业务上乘、文笔过硬,大多是各校学科和专业的带头人。他们对本专业的课程设置、教学大纲、教学教改都有深刻的认识和独到的见解,对高职教育的特色把握能力强,有较高的编写水平。这些都为编写出具有创新性、适用性强的高职教材打下了良好基础。

本套教材的编写以保证基础、加强应用、体现先进、突出以能力为本位的职教特色为指导思想,在内容上遵循“宽、新、浅、用”的原则。所谓“宽”,即知识面宽,适用面广;所谓“新”,就是要体现新知识、新技术、新工艺、新方

法；所谓“浅”，是指够用为度、通俗易懂；所谓“用”，就是要注重应用、面向实践。

本套教材的出版，促进了高等职业教育的教材建设，将对我国高等职业教育的发展产生积极的影响。同时，我们也希望在今后的使用中不断改进、完善此套教材，更好地为高等职业教育服务，为经济建设服务。

新世纪高等职业教育规划教材编审委员会

前 言

随着计算机技术的飞速发展,面向对象的编程技术已经成为当今软件开发的重要手段之一,尤其是 Visual C++ 的出现,大大推进了面向对象与可视化编程技术的应用与发展。因此,掌握面向对象与可视化程序设计的内容与方法已经成为对计算机专业人才应用与开发能力的重要要求。本书的目的就是为了帮助计算机专业的初学者,能够在系统学习 C++ 面向对象程序设计思想核心的基础上,快速熟悉和掌握基于 Visual C++ 的 Windows 应用程序开发技术。

全书共分为三部分。第一部分为 C++ 语言程序设计基础,包括面向对象程序设计思想、类与对象、继承与派生以及多态性等核心内容;第二部分以基于 API 的应用程序实例为主,较为系统地讲述了 Windows 编程的基础知识;第三部分介绍使用 MFC 进行可视化编程的基本方法,包括 GDI 绘图、菜单和工具条等资源的应用、对话框和常用控件的使用、文档/视图结构、常用 MFC 类在编程中的应用等知识点。

本书在编写过程中力求通俗易懂、面向实用。在每一章节中,先讲述一些必要的基础知识,之后配以相应的开发实例,实例中配以较为详细的步骤说明及语法说明,一步一步引导读者完成。在完成实例后,再对某些比较难以理解的程序代码给予分析说明。

本书第 5~9 章、附录由曹向东撰写,第 1~4 章由张国海撰写,全书由曹向东负责统稿。广州大学刘斌副教授审阅了全书,并提出了许多宝贵的意见,在此表示衷心感谢!

由于我们水平有限,错误和疏漏在所难免,恳请读者批评指正。

编 者

目 录

编写说明	
前言	
第1章 面向对象程序设计	
概述	1
1.1 面向对象的思想	1
1.2 面向对象方法的特点	2
1.3 C++ 简介	5
1.4 从C到C++	6
1.5 Visual C++ 环境简介	13
习题	15
第2章 类和对象	16
2.1 类定义	16
2.2 对象	19
2.3 构造函数和析构函数	21
2.4 程序实例	32
2.5 友元	36
2.6 静态成员	40
2.7 类的聚集	45
2.8 模板	49
习题	55
第3章 类的继承与派生	58
3.1 继承与派生	58
3.2 访问控制	62
3.3 派生类的构造函数和析构函数	64
3.4 多继承	67
3.5 实例解析	76
习题	80
第4章 多态性	81
4.1 概述	81
4.2 运算符重载	82
4.3 虚函数	90
4.4 抽象类	93
4.5 实例解析	96
习题	100
第5章 Windows 编程基础	101
5.1 窗口	101
5.2 消息和消息驱动	104
5.3 Windows 应用程序的开发方式	105
5.4 句柄	106
5.5 Windows 基本数据类型	107
5.6 匈牙利表示法	108
5.7 Windows 应用程序的基本结构	109
5.8 实例解析	113
习题	119
第6章 创建 MFC 应用程序	120
6.1 开发环境概况	120
6.2 创建 MFC 应用程序框架	121
6.3 应用程序的基本结构	128
6.4 用户代码设计	142
习题	146
第7章 对话框与控件	147
7.1 对话框的创建	147
7.2 常用对话框控件	148
7.3 CDialog 类成员函数	149
7.4 数据交换和数据验证	150
7.5 实例解析	151
习题	154
第8章 图形与界面设计	155
8.1 GDI 绘图	155
8.2 菜单资源	168
8.3 工具条资源	172

8.4 代码设计	174	9.4 拆分窗口	182
习题	176	9.5 实例解析	183
第9章 文档与视图	177	习题	197
9.1 文档/视图结构	177	附录 MFC 类库基本结构	198
9.2 CDocument 类与 CView 类	178	参考文献	206
9.3 文档类与视图类的相互作用 ...	180		

第 1 章 面向对象程序设计概述

这一章主要介绍面向对象程序设计技术的概念、特点，C++语言发展及如何尽快从 C 过渡到 C++等内容。本章中的一些基本概念，随着后续章节内容的学习会逐步加深理解。

本章要点：

- ◆ 面向对象程序设计
- ◆ 类和对象
- ◆ 输入输出流、函数重载、内联函数、动态内存分配

1.1 面向对象的思想

自人类发明第一台电子计算机以来，计算机科学与技术以其特有的速度迅速发展，大大促进了人类社会的进步。特别是 20 世纪 80 年代以来微型计算机的普及，使计算机的应用已进入人类生活的各个领域，掀起全球信息化的热潮。

众所周知，无论具体应用内容是什么，计算机系统中必不可少的是软件。有人把软件称为计算机的灵魂是毫不过分的。在计算机发展的这几十年来，从应用角度看，软件的功能、形式、使用方便性等方面有了很大发展。在软件开发技术与方法、标准等方面，不少研究机构、学者也孜孜不倦地进行着探索与总结，并取得了可喜的发展。然而，这些研究成果尚未根本改变软件开发的模式，很大比例上的软件开发仍是采用原始的、不规范的方式，使得软件开发的效率十分低下，同时难以保证软件的质量。随着应用的日益普及与深入，软件的开发难度越来越大，使得软件开发和维护面临的问题日趋严重，陷入“软件危机”。软件已成为影响计算机系统与应用的重要因素。

“软件危机”产生的原因，除了软件本身特点外，还与原有的软件开发与维护的方法有关。传统的结构化程序设计方法用过程化的方式来描述应用系统，这种方法以计算机处理过程为主要特征，人们在编程时必须既要设计要处理的数据格式，又要考虑实现算法，不同的数据格式作相同处理、相同的数据作不同的处理需要编写不同的程序，即软件的可重用性很差。正因为如此，软件的维护的工

作量很大。另外，数据的安全性等也难以保证。

面向对象的方法认为：客观世界是由许多各种各样的对象组成的，每个对象具有各自的内部状态和运动规律，不同对象之间的相互作用和联系构成了各种不同的系统与客观世界。

面向对象的方法以人类认识世界的一般规律和思维方式，将要解决的问题抽象为系统，不仅便于描述、理解，而且适用实体的范围与形式广泛。采用面向对象技术，可以使得系统分析、设计和实现的方法得到统一。由于系统是由若干相互独立的对象组成的，使得系统的维护局部化，改变了面向过程方法中某一改动往往涉及系统中的多个方面，系统维护甚为繁重的现象。

可以说，面向对象的方法以其符合人类的思维方式，自然地表示了现实社会的各类实体和问题，从而以较好的方法实现了对软件复杂性的控制，提高了软件生产率，因此对软件开发具有重要意义。

面向对象程序设计中有几个主要概念，就是类、对象、继承、封装和多态。

1.2 面向对象方法的特点

1.2.1 对象 (Object)

客观世界存在着各种各样的事物，有些是有形的，有些是无形的，往往还存在着互相之间的关系。所以我们在描述这些事物时，不仅需要描述事物的本质特征，也要描述其与其他事物的相互关系。

在面向对象程序设计方法中，我们用对象来描述客观世界，即对象是客观世界中的事物在计算机系统的一种描述。对象由描述事物内部特征（称为属性）的一组数据和描述对其数据及与外部系统的相关操作组成。

现实世界中事物的特征作用是有差异的，如描述人的特征，有姓名、性别、身高等基本数据，也有像身份证、银行存折的密码等私有数据。在面向对象语言如 C++ 中，将数据分为公有 (public)、保护 (protected) 和私有 (private) 三种类型。外部对不同类型的数据有不同的操作权限。

就像客观世界中，同一事物在不同场合因应用要求不同而关注不同的特征和不同处理方法的情况类似，面向对象程序设计中的对象的属性与方法由用户根据应用需要确定。

另外在面向对象程序设计中，对象的属性和方法是不可分开的整体，这就是封装的概念，其有效地保证了数据的安全性。

1.2.2 类 (Class)

人们在认识客观世界时，面对千姿百态的众多事物与现象，往往采用分析、归类的方法。分类就是以本质特征为依据，找出事物或现象的共性，将具有共性的事物归纳为一类，并提升到抽象概念。有了类的概念，不仅有助于我们方便地对某类事物进行统一的描述，也有助于通过相同的特征来认识同类中的其他事物与现象。正因为如此，面向对象方法也引用了类的概念。

类是对一组具有相同属性和方法的对象的抽象描述，这组对象则是这个类的具体实例。我们说类是对象的抽象描述，实际上它的作用就是定义对象的特征。因此，一个类的定义包含两方面的描述：

- 1) 该类对应对象的属性定义或结构定义。
- 2) 对该类对应对象的属性操作的方法定义。

每个对象均属于某个类，是某个类的一个实例。一个类可以有多个对象，这些对象具有同样的特征（即属性和方法），当然每个对象的属性值可以不同。

如我们定义一个学生类，其属性主要有学号、姓名、年龄、专业等，其方法之一为打印上述属性值。在此基础上我们可以定义几个学生类的对象，每个对象可记录及打印某一学生的学号、姓名、年龄、专业等信息。

1.2.3 继承 (Inheritance)

前面提到人们在认识和描述现实世界的事物时，经常用分类的方法。这就是运用已掌握的同类其他事物或现象的特征去认识新的事物或现象，并根据需要可将类分为多个层次，即大类下可依照某些特征细分为若干小类，小类具有大类的基本特征，又有互相之间不同的特征，而且小类还可以分为更小的类，以此类推。

面向对象的思想与人们认识和描述现实世界的事物或现象的层次思想相同，可以在已定义的类上形成一个新类，新类自动拥有已定义的类的属性和方法，这就是继承。继承是面向对象程序设计方法中的重要特征之一，也是实现软件复用的一种手段。

利用继承而得到的新类，不仅可以继承已定义类的属性和方法，还可以增加定义新的特征，使之既方便地利用已有的成果，又可灵活地扩充，满足新的需要。

如假设已有关于人的类，有姓名、性别、年龄等属性，我们就可以在这类中派生出学生类，这学生类自动拥有人类的姓名、性别、年龄等属性，在这基础上我们可以增加新的属性，如所读专业、年级等。

在面向过程程序设计中，开发新系统时往往需要利用已有软件的功能，这需要花不少时间和精力去修改源代码才能符合新系统的要求，而应用面向对象方法中的继承则无需已定义类的源代码，又提供了扩充与修改新特征的途径。

在不同的面向对象程序设计语言中，有多继承和单继承之分。允许新类可以从多个类派生而得，称为多继承，仅允许新类从单个类派生而得称为单继承，C++允许多继承。

1.2.4 封装 (Encapsulation)

从形式看，封装就是把对象的属性和方法组织为一个独立的单元。这样对外部而言，对象的内部是隐蔽的，不可随意地直接访问，必须通过对象的接口实现与对象的联系。

封装的定义为：

- 1) 一个明确的边界，其限制了本对象的内部变化范围边界。
- 2) 一个接口，其用以描述本对象与其他对象之间的相互作用。
- 3) 受保护的内部实现，其给出了由软件对象提供的功能细节，不允许外部直接访问。

封装不仅将对象的属性和方法组织为独立单元，而且还隐蔽了对象的内部细节，使外部只能通过接口实现访问，避免了外部有意无意地对对象内部的破坏，提高了数据的安全性。

现实生活中，人们只知道对象作用和如何使用，而不关心其内部细节的场合是很多的。就像广泛使用的个人电脑，一般用户是通过键盘、鼠标、显示器等人机接口来调用各种功能的，整个系统安装在机箱内，不仅易于使用，而且也安全可靠。

1.2.5 多态性 (Polymorphism)

多态性是指类的属性和方法在被特殊类继承后，可以具有不同的数据类型或表现出不同的动作。从程序设计语言角度看，多态性表现为一个名词可具有多种语义，这为软件开发带来很多方便。

与面向过程的方法相比，多态性是面向对象方法中的一个重要特征。如在面向过程的语言中，对画不同形状的图形需定义不同的函数，而在面向对象的方法中，将作各种几何图形的共同特征抽象出来，在一个基本类中加以定义，然后利用继承派生出诸如画点的类、画圆的类等。显然各派生类有一个同名画图形的方法，但具体图形的实现细节是不同的。

在 C++ 语言中，多态性主要表现为运算符重载和虚函数等。

1.2.6 面向对象程序设计方法

面向过程的程序设计方法是 20 世纪曾迅速普及并至今仍在使用的一种软件开发方法，而 20 世纪 90 年代异军突起的面向对象的程序设计方法，以其可重用、

易维护等优点成为软件开发方法的发展趋势。那么从开发的具体过程而言,二者有哪些异同点呢?

系统需求分析是任何开发方法都必须进行的首要工作,这个阶段的目的就是通过调查研究,确定用户的需求和系统所要实现的功能、主要指标等。

在问题分析与实现方法的设计方面,面向过程与面向对象的方法采用不同的思路与方法。面向过程的方法是通过自顶向下、逐步求精的思路,将系统以功能划分为若干个模块,通过算法设计和数据结构设计,确定每个模块的实现方法。面向对象的方法,则从复杂的系统中抽象出各个对象的属性和方法等,并进一步归类、整理、补充,形成对类和对象的规范描述。

面向过程程序设计中的编程与调试是按照模块设计中的算法,以某种语言的形式编写成相应的程序,程序员对算法的理解与经验直接影响到程序质量与开发效率,由于算法与数据结构关系密切,使得程序编写与调试的难度增大。而面向对象的程序实现,首先是尽可能地调用系统提供的类库,或自行建立必要的类和对象。应用继承、多态等手段,充分利用已有的成果,使编程与调试的工作量相对减少、难度相对降低。

面向对象程序设计方法是在面向过程的方法基础上发展起来的,以自然形式表现现实世界中的事物和对象,更符合人类思维习惯,表现出强大的生命力。

从上面分析中可以看到,面向过程和面向对象程序设计方法两者采用的思路和方法是有所差异的。然而两者都要求程序员有较强的逻辑思维能力,就像面向过程程序设计能力需要不断学习、实践,逐步提高一样,学习面向对象程序设计也需要一个过程。真正理解和掌握本节所介绍的面向对象程序设计中的重要概念并不容易,本书在后面章节还会结合 C++语言中的具体内容,介绍这些概念,逐步加深理解。

1.3 C++简介

C++是在 C 语言的基础上,增加了面向对象程序设计的内容而形成的,它是由 Bjarne Stronstrup 博士于 1980 年在 AT&T 的贝尔实验室开发的。最初称为“带类的 C 语言”,1983 年正式取名为 C++。

C++语言的特点主要体现在两个方面:

- 1) 全面兼容 C 语言。C++是 C 的超集,与 C 语言全面兼容。一般 C 语言源代码无须修改,可直接在 C++语言环境下进行编译、执行。C++秉承了 C 语言高效、灵活,适合进行各种软件的开发。

- 2) 支持面向对象方法。支持面向对象方法是 C++的重要特点,通过类、对

象、继承、封装和多态等概念和实现方法，体现了面向对象的设计思想，也使 C++ 语言得到广泛的应用。

本教材定位在对 C 语言已有一定了解的层次上，所以对 C++ 中与 C 相同的部分不作介绍，仅在 1.4 节中介绍 C++ 中面向过程程序设计与 C 不同的相关内容。

C++ 是由 C 发展而来，但二者的程序设计思想与方法是有很大差异的。学习 C++ 要注意二者的差异，逐步掌握 C++ 的特点及面向对象程序设计方法。

1.4 从 C 到 C++

在前面我们简单介绍了 C++ 的特点，对 C 语言是有一定基础的读者如何尽快从 C 过渡到 C++ 呢？

C++ 除了在 C 的基础上增加了面向对象的部分，在面向过程的某些部分，为了从原理、结构上与面向对象部分统一，对功能进行了扩充，语句形式也有所变化，这些将在本节中进行介绍。

1.4.1 C 和 C++ 的简单程序

我们通过一段简单程序，看看 C 与 C++ 的异同点。

例 1.1 求两个整数之和的 C 与 C++ 程序。

/* example1. c C 语言程序 */

```
main()
{
    int a,b,s;
    scanf("Enter two integer number  %d,%d",&a,&b);
    s=a+b;
    printf("a+b=%d\n",s);
}
```

// example2. cpp C++ 语言程序

```
#include<iostream.h>
void main()
{
    int a,b,s;
    cout<<"Enter two integer number:";
    cin>>a>>b;
```

```

s=a+b;
cout<<"a+b="<<s<<endl;
}

```

从上面的例子，我们可以看到 C 和 C++ 程序在结构上是相同的，都以函数为基本构成模块，通常由一个主函数和若干个子函数组成。

另外正像我们在前面所说的那样，C 和 C++ 在一些地方还是存在不同的实现方法和语句格式的。如对于不需返回的函数，则在其名前用 void 加以表明，语句注释除了 C 语言中的 /*...*/ 格式外，增加了 // 至本行结束的注释内容这种格式。

下面我们重点介绍二者主要的不同之处。

1.4.2 输入输出流

C 语言的标准函数库提供了许多标准函数，包括像 scanf()、printf() 等输入输出函数，在源程序中我们可以直接调用，在 C++ 中仍可以这样使用。从面向对象的角度，C++ 将数据在不同对象间的流动抽象为“流”，不仅使 I/O 操作统一到面向对象方法上，而且实用中更加灵活安全。

C++ 中为输入输出建立了 ios 类，ios 类派生出输入流 (istream)、输出流 (ostream)、文件流 (fstreambase) 和串流 (streambase) 等 4 个派生类。将 I/O 设备定义为对象，cin 是输入流的对象，cout 是输出流的对象。预定义输入流 cin 表示标准输入设备——键盘，输出流 cout 表示标准输出设备——显示器。

C++ 中，将从流中获取数据的操作称为提取（也称为读操作），将向流中添加数据的操作称为插入（也称为写操作）。分别用“cout<< 输出项”和“cin>> 输入项”的格式实现输入输出。cin 后可以用多个>> 输入项实现多个数据的输入，同理 cout 也是如此。在上面的例子中我们用了默认格式的输入输出，还可以用 C++ 中各种格式化方式定义输入输出格式，见表 1-1。

表 1-1 常用格式化操作符

操 作 符	功 能
dec	数值类数据采用十进制形式
hex	数值类数据采用十六进制形式
oct	数值类数据采用八进制形式
ws	输入插入空字符
endl	输出换行
ends	输出空字符
flush	刷新 I/O 缓冲区
setfill(int c)	设置填充字符

(续)

操 作 符	功 能
setprecision(int n)	设置浮点数精度
setw(int n)	设置输入、输出数据项的宽度

在应用常用格式操作符对输入输出的数据进行格式说明时,将操作符作为 I/O 操作内容,就实现了后继数据格式的说明。

例 1.2 输入输出流实例。

```
#include <iostream.h>
void main()
{
    cout<<setw(10)<<"personal"<<"computer"<<endl;
    cout<<setw(10)<<"personal"<<setw(10)<<"computer"<<endl;
}
```

程序运行结果为:

```
personalcomputer
personal  computer
```

这是因为 `setw` 操作符仅对其下一个数据有效,如 `setw` 设置宽度小于后继的数据宽度,对于输出仍按数据宽度输出,对于输入则进行截位处理。对于数值数据的进制格式操作,则一直保持至下一个格式设定为止。

C++ 还有其他格式操作符以及用户自定义格式控制函数,限于篇幅在此不作介绍。

1.4.3 引用调用

C 语言中,函数调用时参数传递的方式是传值。传值是单向的,往往无法满足一些要求将已变化的参数的值反传回主调函数的应用要求。解决这问题的办法之一是采用指针。在 C++ 中增加了引用调用,采用引用调用可使程序变得简单明了。引用调用是 C++ 的一个新特性。其概念是对某个对象或变量赋予一个别名,当采用引用调用时,程序对别名的操作实际上就是对原对象或变量的操作,这样就可将变化的数值传回至主调函数。

C++ 中的引用调用格式为标识符“&”。“&”在 C++ 中还可作其他运算符,编译器会根据语句具体情况分别加以解释。

上面我们从函数调用时参数传递的角度讨论了引用调用的含义,实际上引用还可以扩展到一般变量调用中的引用调用,其含义与函数调用的参数传递是相同的。