张祖浩 葛建芳 编著 康祥顺 审校

C++程序设计

基础与实践教程

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

C++程序设计基础与实践教程

张祖浩　葛建芳　编著

康祥顺　审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是以未学过程序设计语言的读者为主要对象进行编写的。本书主要分前后两大部分。前一部分主要介绍面向过程，以函数为模块的程序设计。内容包括：基本数据及别名和指针、数据的运算和指针的整数加减运算及简单输入/输出运算、流程控制、函数、数组、枚举类型和结构类型。后一部分主要介绍面向对象，以对象为模块的程序设计。内容包括：类和对象、继承和派生、多态性、输入/输出流类体系、命名空间、模板和异常处理。

本书可作为高等院校程序设计课程的教学用书，也可作为自学C++程序设计用书或教学参考书。

图书在版编目（CIP）数据

C++程序设计基础与实践教程 / 张祖浩，葛建芳编著.北京：电子工业出版社，2009.5
ISBN 978-7-121-08551-2

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

中国版本图书馆CIP数据核字（2009）第041653号

责任编辑：姜　影
特约编辑：孔　群
印　　刷：北京天竺颖华印刷厂
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
　　北京市海淀区翠微东里甲2号　邮编：100036
开　　本：787×1092　1/16　印张：28.75　字数：730千字
印　　次：2009年5月第1次印刷
定　　价：49.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

C++语言是从 C 语言发展演变而来的一种面向对象的程序设计语言。它保持了 C 语言的简洁、高效、源程序的可移植性好等特点；同时克服了 C 语言的类型检查机制薄弱的缺点，增加了对面向对象的支持，特别适合中等和大型程序项目的开发。目前，C++语言已经在各个领域都得到了广泛的应用，成为主流的面向对象程序语言。

C++程序设计是高等学校普遍开设的计算机核心基础课程，涉及程序设计的思想、方法、语法、算法、调试技术和操作技能，理论性、综合性和实践性强，因而对初学者而言，具有相当的难度。本书作者凭借多年积累的丰富教学经验，广泛学习和研究了C++程序设计教材；深入到学员之中，听取意见，面对面探讨困难所在以及克服困难的办法；从而摸清了问题的症结，重新作出思考，编写本书。

本书是以未学过程序设计语言的读者为主要对象进行编写的。较系统地介绍了 C++语言的基本概念和程序设计的基本方法，共分为 12 章。

本书具有如下特点：

1．结构合理，层次分明

本书按程序设计方法演进发展的自然顺序及 C++实际编程能力形成的过程，将 C++教学内容分为两大部分，即面向过程程序设计（第 1～7 章）和面向对象程序设计（第 8～12 章）。这样组织，由浅入深，循序渐进，符合读者的认知规律和编程能力的形成规律，便于教学的组织、实施和考核，利于教学效果的巩固和教学质量的提高。

2．程序设计为主，语法为辅

本书的宗旨是：不仅要使读者掌握 C++语言本身，而且能够对现实世界中较简单的问题及其解决方法用计算机语言进行描述。因此，本书讲解语法时着重从程序设计方法学的角度讲述其意义和用途。

3．化难为易，平易近人

C++语言教学和实际编程中有许多难点，例如递归、指针、链表、虚函数和输入/输出流等，本教材采取多种措施克服难点：用简洁浅显的语言简述复杂的概念；用启发式的方法引入新的知识点，使读者没有突兀的感觉；用简明的图表配合文字叙述来直观地展示程序执行的流程，使读者很容易读懂程序；用一步步指示的方式讲解解决问题的思路，使读者掌握实际分析问题、解决问题的方法。

4．范例程序，精心设计

程序是语法、算法、思想和方法有机结合的载体，是计算机解决实际问题的钥匙，学习程

序设计离不开程序。书中大量的范例是经过作者精心挑选和精心设计的，表达准确、简练，书写规范，示范性强。

本书每一章后均提供了一定数量的练习题，可供读者复习时参考。

本书可作为高等院校程序设计课程的教学用书，也可作为自学C++程序设计用书，或教学参考书。

本书第1～11章由张祖浩编写，第12章由葛建芳编写。康祥顺审校。由于编者的水平有限，时间仓促，错误和疏漏在所难免，敬请广大读者提出宝贵意见。

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（http://www.hxedu.com.cn），在“下载”频道的“图书资料”栏目下载。

目　录

第1章 概 论

1.1 算法、程序和语言

1.1.1 算法和程序

要让计算机给我们做数据处理，我们必须把数据处理操作的步骤考虑好，交给计算机。目的是要让计算机按我们所考虑的步骤一步步进行操作，最后达到我们的目的。

我们所考虑的、对数据处理操作的步骤叫做算法。

算法必须用某种语言写出来形成一个程序。程序输入计算机，使计算机按程序指令一步步地对数据进行操作，最终实现按算法对数据进行的处理。

那么，当我们对数据处理的算法考虑好了，用什么语言写出程序来输入计算机呢？

1.1.2 低级语言和高级语言

● 低级语言

计算机硬件只能听从由二进制码（0 和 1）组成的指令。由二进制码组成的指令序列叫做机器语言。计算机对机器语言能识别，并能据此直接对内存中的数据进行相应操作。在计算机诞生初期，专家们是用机器语言写出程序的。

由于机器语言全是二进制码，人们难懂难记，编程费时又费力。为此，专家们研究出一种汇编语言，它的主要特征是用助记符来表示每一条机器指令。例如，采用助记符“ADD”代替二进制码“001”来表示加法操作；采用助记符“MOVE”代替二进制码“010”来表示数据移动操作，等等。这样，就让人能看懂，使编程工作容易一些。用汇编语言编程后，由一个汇编系统负责翻译成机器语言。计算机见了机器语言，就能识别并照此执行。机器语言和汇编语言称为低级语言。

● 高级语言

虽然汇编语言与人类语言近了一步，但它与人们习惯的语言还相差甚远。后经专家们不懈努力，设计出与人们习惯语言相近的程序设计语言。这种语言叫做高级语言。

20 世纪 50 年代至 70 年代， FORTRAN、BASIC、ALGOL、PASCAL、COBOL、ADA 和 C 等高级语言相继问世。它们凭借各自的优点，在程序设计中都曾占有一席之地。高级语言的出现大大促进了计算机软件的开发。

用高级语言编写出程序，经编译系统翻译成机器语言，让计算机能识别，并能遵照执行。

1.2 面向过程和面向对象的程序设计

1.2.1 面向过程的程序设计

在计算机诞生之初，科学家们是用计算机来作数学计算的，例如计算炮弹的飞行轨迹等。

这时所面临的是一个数学计算过程。设计者按数学计算的过程，设计出算法，用高级语言按算法写出程序。由于所面向的是一个过程，故名之为面向过程的程序设计。

初期的高级语言都是面向过程的程序设计语言。在面向过程的各种高级语言中，相比而言，C 语言更注重实用，注重表达式的简捷高效。C 语言中有指针变量，能直接对内存中的数据进行操作。因此，C 语言既是高级语言，同时又具有低级语言的特征。因而 C 语言受到广泛欢迎，逐渐成为软件开发的主流语言。

1.2.2 面向对象的程序设计

随着计算机应用的日益广泛，要解决的问题也日趋复杂。有许多问题，例如对一个学校学生数据的管理问题，靠一个简单的数学过程是不容易解决的。必须综合考虑每个学生的各种数据（例如学号、姓名、性别、年龄、各科成绩等）以及对数据的处理（例如计算每个学生的平均成绩、各班成绩，对成绩进行排序，判断有多少学生补考或留级，等等）。这时，必须把有关学生的各项数据及对数据的处理归为一个学生类。而一个学生类的对象（或者说，一个学生类的实例）就是某个具体的学生。就像人类的对象就是一个具体的人一样，程序设计者所面向的是要对一个类的对象的数据及数据处理作综合考虑，进行程序设计。这是面向对象的程序设计（object oriented programming，缩写为 OOP）。C++就是为面向对象程序设计而开发出来的高级语言。

1.2.3 从 C 到 C++

在面向对象程序设计语言诞生之前，C 语言业已成为程序设计的主流语言而风靡全世界。C 语言已经为软件开发打下了基础。要想摈弃 C 语言而另起炉灶，来开发面向对象程序设计语言显然是不合适的。

在 20 世纪 80 年代， AT&T Bell（贝尔）实验室的 Bjarne Stroustrup 博士及其同事们在 C 语言的基础上，开发面向对象程序设计语言获得了成功。这种语言保留了 C 语言的所有优点，增加了面向对象的机制，是 C 语言的增强版，因而名之为 C++。同时，与 C++配套的编译系统也随之诞生。此后，随着版本的不断升级，C++的功能也不断地完善。

由于 C++语言是在 C 语言的基础上发展起来的，并且与 C 语言兼容，因此，C++既可用于面向过程的程序设计，又可用于面向对象的程序设计。

1.3 C++程序的写出和运行

我们通过下面一个简例，来简单地介绍一下 C++程序的写出和运行。

1.3.1 C++程序简例

[例 1-1]试用 C++语言设计一程序，输入两任意整数，输出二者之差的绝对值。

先输入两整数，然后两数相减得差 dif。若 dif 为负，则输出-dif；若 dif 为正，则输出 dif。这就是问题的算法。根据这个算法，设计程序如下：

```
#include<iostream>                    //包含头文件 iostream
using namespace std;                  //使用命名空间 std
```

```
int   main()                          // main 是主函数。程序从主函数开始执行
{
int a,b,dif;                          //声明 a,b 和 dif 是三个整数类型(int)变量
cout<<"请输入被减数：  "<<'\n';        //输出“请输入被减数：”至屏幕
cin>>a;                               //从键盘对变量 a 输入被减数
cout<<"请输入减数：  "<<'\n';          //输出“请输入减数：”至屏幕
cin>>b;                               //从键盘对变量 b 输入减数
dif=a-b;                              //求出 a-b，赋给变量 dif。"="是赋值的意思
if(dif<0) cout<<-dif<<'\n';           //若 dif<0，则输出-dif 值
else      cout<<dif<<'\n';            //否则，输出 dif
return 0;                             //程序正常执行，返回 0
}
```

运行情况如下：

```
请输入被减数：                        //屏幕上显示“请输入被减数：”
123↙                                  //键盘输入 123，按一下 Enter 键
请输入减数：                          //屏幕上显示“请输入减数：”
456↙                                  //键盘输入 456，按一下 Enter 键
333                                   //屏幕上显示所计算的绝对值结果
```

通过这个简单的例子，我们说明以下几点：

（1）注释

程序的右边，以“//”开头写出的一行字是程序的注释。这种注释的写法只限于在一行之内有效。如果一个注释要写好几行（超过了一行），则注释必须改为以“/*”开头，以“*/”结尾。

注释的作用是对程序做解释，以便让人们能看懂程序，也就是提高程序的可读性。注释是不参与程序运行的。

我们把程序结合注释来看，是基本能看懂程序的。

（2）包含头文件和使用命名空间 std

按标准 C++的新要求，程序的头两行用了“#include<iostream>”和“using namespace std”。意思是包含头文件 iostream，和使用命名空间 std。这两行是为程序中所要用的输入和输出提供必要支持的。其道理暂且不去深究，后面将逐步阐明。只是要记住，每逢程序设计，开头总少不了要写上这两行。

（3）主函数

main()是主函数。程序总是从主函数 main 开始执行。int 表示函数的返回值是整数。圆括弧中空白表示函数无参数。下面花括弧所括的内容是函数体，也就是函数所要做的事。若程序正常执行完毕，就向操作系统返回数值 0（即 return 0）；否则返回数值-1（即 return -1）。

（4）标识符

我们给变量等实体所取的名字，如程序中变量名 a,b 和 dif 等，统称为标识符。标识符限定只能由大写字母、小写字母、数字或下划线组成，并且只能以下划线或字母开头。大小写字

母有区别。例如对于_B2d 和_b2d，虽然都是合法的标识符，但因有大小写字母的不同，二者不可视为同一个标识符。

（5）关键字

在 C++语言中，已经具有特定含义的词叫做关键字。例如，main、int、if、else 等等都是关键字。我们取名字所用的标识符不可与关键字相同。

（6）语句

程序中语句要用分号“;”结尾，分号是语句的组成部分。语句不一定一行只写一句。在一行中写好几句，或一句写成好几行都行。分行写和缩进写只是为求醒目，便于看懂而已。

对于程序中其他标点符号，以后随讲随看随记，逐渐积累是不难的。

（7）流程控制

if 和 else 两语句显然是对程序进行流程控制的，它把程序流程分为两个分支。括弧中的表达式就是作逻辑判断用的。若表达式(dif>0)为真，就执行 cout<<-dif<<'\n';；否则，就执行 cout<<dif<<'\n';。

（8）程序的流程图

程序流程图就是表示程序执行步骤的图。例 1-1 的程序流程可用文字表达如下：

①首先对解题中所要用到的整型(int)变量 a,b 和 dif 作一个声明。

②对变量 a 和 b 分别输入两个整数，作为被减数和减数。

③将 a 减 b 所得之差（a-b）赋给变量 dif。程序中用的运算符“=”是赋值的意思。

④判断表达式 dif<0 是真还是假。

⑤若为真，则输出-dif。

⑥若为假，则输出 dif。

程序流程各个步骤可以用直观形象的图框符号来表示。常用的图框符号见图 1.1。

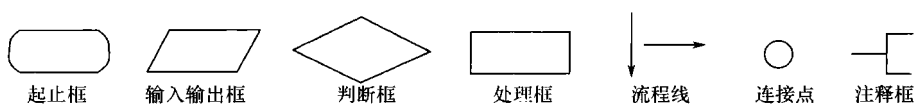

图 1.1　流程图常用的图框符号

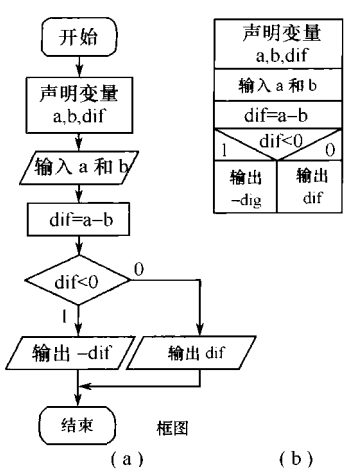

图 1.2　流程图的两种形式

上例中，程序的流程图可以表示如图 1.2。其中（a）图叫做框图。从开始框开始，图中箭头表示流程的方向，每一个框都表明了一个步骤。菱形框内的式子“dif<0”是供逻辑判断用的表达式（叫做控制表达式）。菱形尖角处“1”表示“真”,“0”表示“假”。判断“dif<0”是真还是假，由此分流出两个分支，两分支中选用一支。这就是所谓流程控制。最后用结束框表示程序结束。

从开始，一个箭头进入流程，至最后一个箭头进入结束，这种情况叫做单线进单线出，简称单进单出。

如果把框图中的箭头和线条省略，就可构成图（b）的形式。图（b）是表示流程的另一种简示图。它较紧凑、简明、好懂，并且在程序复杂时，可免得

线条多，看花了眼。它的含义是和（a）图一样的，以后见到它要能知道它的含义。

1.3.2 C++程序从写出到运行的几个步骤

（1）用C++语言写出的程序，叫源程序。经编辑输入计算机后形成扩展名为cpp的源文件。cpp是C plus plus的缩写，即C++的意思。

（2）源程序经过编译程序翻译，成为用机器语言表示的目标程序。目标程序又称目标代码，或程序代码。由此形成扩展名为obj的目标文件。obj是object的缩写。

（3）把各相关文件链接起来形成扩展名为exe的可执行文件。

在以上每一个步骤中，如果有错就必须及时修正。待每一步中的错误都修正好后，最后在运行中进行调试、修正并完善。

1.3.3 本书内容的安排

本书内容的安排是顺着程序设计的发展过程，先讲面向过程的程序设计，然后进展到面向对象的程序设计。

从上面介绍的一个简单程序可看出，编写程序的目的是为了要让计算机按我们设定的算法步骤，为我们进行数据计算或处理。程序虽简单，但涉及到数据类型、运算符和表达式、输入和输出、流程控制、函数以及编译预处理（例如#include）等方面的知识。而这些内容正是第2、第3、第4、第5章的内容。是我们即将逐章展开作详细介绍的内容。随后内容拓展到自定义数据，以及面向对象程序设计。

C++编译系统经历了一个发展过程。ANSI C++标准（以下简称C++标准）正式通过并发布后，在一些编译系统及教材中尚未全面执行。但最新推出的集成开发环境VC++ 2008已严格要求按C++标准编写程序，否则就不予通过。本书按C++标准编写程序。

1.4 习题

1. 进行注释可有哪几种方法？
2. 下列标识符哪些合法，哪些不合法？为什么？

（1）_My3pen （2）teA$her （3）-2abc

（4）34pp （5）A2.85st （6）start_

3. 语句以什么符号结尾？
4. C++程序从写出到运行分几个步骤？各形成什么文件？

第2章　基本数据、别名和指针

2.1　基本数据

数据是程序处理的对象而存储于内存空间中。用于存储某数据的内存空间，本书简称之为存间。存间的大小与数据类型有关。数据类型分基本数据类型和导出数据类型。所谓基本数据类型就是指的一般常用的整数、实数（带小数点的数）和字符等类型。所谓导出数据类型就是以基本数据类型为基础，根据程序的需要而导出的数据类型，实际是由编程者自己定义的数据类型。故导出数据类型又叫自定义数据类型。这里先介绍基本数据类型及其操作处理。从第6章开始，再谈导出数据类型及其操作处理。对数据进行操作处理会用到别名和指针。

一般而言，随着程序的运行，数据之值可能会有所变化。会变化的数据称之为数据变量，简称变量。不变的情况可视作变量的特殊情况。

存储变量的内存空间就是变量的存间。变量存间用作随时存储变量之值。这样，变量存间的内容就是该变量的值。变量存间的大小视该变量类型而定。以下变量存间简称为存间。

要对数据进行处理，必然会涉及到存间，涉及到数据类型。下面我们就先从数据类型谈起。

2.1.1　基本数据类型

C++基本数据类型主要有char（字符型）、int（整数型）、float（浮点型，表示实数）、double（双精度浮点型，简称双精度型）。

其中，char 型数据在计算机中是以其ASCII码来表示和存储的。每一个ASCII码实际就是一个整数，与字符一一对应。故从本质上说，char型也可看成是整数型。

按它们带不带正负号又可修饰为signed（有符号）或unsigned（无符号）。

按它们的存间所占内存单元数（即字节数）又可修饰为short（短）或long（长）

基本数据类型与这些修饰词组合后，综合如表2-1所示。实际上，不同的C++系统有可能会有所区别。

表2-1　基本数据类型

基本数据类型名			存间大小（字节数）	长度（位）	取值范围
字符		char(signed char 的简称)	1	8	-128～127
		unsigned char	1	8	0～255
整数	短整型	short(signed short int 的简称)	2	16	-32768～32767
		unsigned short	2	16	0～65535
	整型	int(signed int 的简称)	4	32	-2^{31}～$(2^{31}-1)$
		unsigned int	4	32	0～$(2^{32}-1)$
	长整型	long(signed long int 的简称)	4	32	-2^{31}～$(2^{31}-1)$
		unsigned long	4	32	0～$(2^{32}-1)$
实数		float	4	32	-3.4×10^{-38}～3.4×10^{38}
		double	8	64	-1.7×10^{-308}～1.7×10^{308}
		long double	8	64	-1.7×10^{-308}～1.7×10^{308}