面向21世纪高等院校计算机教材系列

C++程序设计教程

郑 莉 刘慧宁 孟 威 编著

机械工业出版社
China Machine Press

面向21世纪高等院校计算机教材系列

C++程序设计教程

郑莉　刘慧宁　孟威　编著

机械工业出版社

本书介绍了 C++ 语言和面向对象的程序设计方法、常用的群体数据及其处理方法、标准 C++ 库。本书内容全面，例题丰富，讲述简明清晰。由于书中所有的概念和语法都以丰富的例题为背景来讲解，因此易读易懂，实用性强。

本书适合高等学校和培训机构用于 C++ 语言程序设计课程教学，也适合具有一些高级语言编程基础的读者用于自学。

图书在版编目（CIP）数据

C++ 程序设计教程/郑莉等编著 .—北京：机械工业出版社，2001.7

面向 21 世纪高等院校计算机教材系列

ISBN 7-111-08318-0

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

中国版本图书馆 CIP 数据核字（2001）第 033220 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：胡毓坚

责任编辑：王琼先

责任印制：郭景龙

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2001 年 9 月第 1 版·第 2 次印刷

787mm × 1092mm 1/16 · 19.75 印张 · 485 千字

5 001—10 000 册

定价：28.00 元

出版说明

随着计算机技术的飞速发展，计算机在经济与社会发展中的地位日益重要。在高等院校的培养目标中，都将计算机知识与应用能力作为其重要的组成部分。为此，国家教育部根据高等院校非计算机专业的计算机培养目标，提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次教育的课程体系。根据计算机科学发展迅速的学科特点，计算机教育应面向社会，面向潮流，与社会接轨，与时代同行。随着计算机软硬件的不断更新换代，计算机教学内容也必须随之不断更新。

为满足高等院校计算机教材的需求，机械工业出版社聘请了清华大学、北方交通大学、北京邮电大学等院校的老师，经过反复研讨，结合当前计算机发展需要和编者长期从事计算机教学的经验精心编写出“面向21世纪高等院校计算机教材”。

本套教材理论教学和实践教学相结合，图文并茂，内容实用、层次分明、讲解清晰、系统全面，其中溶入了老师大量的教学经验，是各类高等院校、高等职业学校及相关院校的最佳教材，也可作为培训班和自学使用。

前　言

C++语言是目前使用最为广泛的一种面向对象的程序设计语言，它是从C语言发展演变而来的，其主要特点表现在两个方面，一是全面兼容C语言，二是支持面向对象的方法。面向对象的方法使程序能够比较直接地反应问题的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发，并且可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

C++语言是在C语言的基础上发展起来的，但它更是一次变革。两者的本质差别在于C++语言支持面向对象的程序设计，而C语言仅仅支持面向过程的程序设计。掌握好面向对象的程序设计思想是学好C++语言的关键。C++是一个完整的语言，没有编程经验的读者完全可以直接学习C++语言。当然，读者如果具有C语言的基础，或许会更快地成为一个好的C++程序员。

本书作为一本C++语言的入门教材，不仅详细介绍了C++语言本身，而且深入讲述了面向对象的方法。本书的主要特点是语言流畅，简洁易懂，例题丰富，实用性强。这使得读者不仅可以学会一门程序设计语言，还能初步掌握面向对象的程序设计方法。其中丰富的便题使得初学者可以在学习的同时就开始积累初步的编程经验，以尽快达到学以致用的目的。

我们几位作者均为清华大学的副教授或博士，常年从事计算机基础课教学和应用开发工作。从工作的实践中我们体会到，基础课教材必须要兼顾理论性与实用性。本书一方面具有大学教材理论严谨、概念准确、逻辑性强的特点，同时又具有应用培训教材实用性强的优点。因此既适合用作大中专院校的程序设计课程教材，也适合各类培训机构用作培训教材。

为了使读者更顺利地学习C++语言程序设计，我们对不同类型的读者有一些建议：如果您已经熟悉C语言或JAVA语言，您可以使用本书顺利地学会C++语言；如果您只熟悉除C和JAVA以外的其他编程语言，在学习本书时您还需要另外选择一本简单介绍C语言的书，已备随时查阅。如果您没有任何程序设计基础，也可以使用本书学习C++语言，只是我们建议您最好在教师指导下学习，并且应该参考一本较详细介绍C语言的教材，这样可以达到事半功倍的效果。当然，学习的方法和过程因人而异，这些建议只是我们在教学工作实践中得出的一些经验，并不一定符合每位读者的实际情况，写在这里仅供参考。

要学好编程，重在实践，为了方便读者巩固所学的知识，在实践中提高编程能力，我们还编写了与本书配套的《C++语言程序设计习题集》。希望读者在学习过程中不仅要勤于思考，还要勇于实践。

参加本书编写工作的还有张智海同志，在此深表感谢。

书中如有错误和不足之处，希望各位读者和专业人士不吝赐教，我们将非常感激并及时更正。

作者

目　录

第1章　C++语言编程入门

C++语言是一种已得到广泛使用的面向对象的程序设计语言。本章首先回顾C++语言的发展历史，讨论C++语言对面向对象程序设计的支持；然后，介绍用C++语言开发程序的步骤及一个简单C++程序的组成；最后，详细地介绍C++语言支持的数据类型、运算符、表达式和控制语句等。利用这些知识，我们就可以开发出简单的C++程序。

1.1　C++语言的产生

计算机语言是计算机软件的基础，自第一台电子计算机诞生以来，伴随着计算机应用的日益广泛，计算机语言也得到不断的充实和发展。

20世纪60年代，Martin Richards设计出BCPL语言，它被用于在开发软件系统时作为记述语言。1970年Ken Thompson在BCPL的基础上开发了B语言，美国DEC公司的PDP-7计算机中的UNIX操作系统就是用B语言开发的。其后的1972年，Dennis Ritchie等在为PDP-11计算机开发UNIX操作系统时，对B语言做了进一步的改进，推出了更加通用的C语言。

尽管C语言同时具有高级语言与汇编语言的优点：如语言简洁、程序执行效率高、可直接访问物理地址、具有良好的可读性和可移植性等，并得到了广泛使用。但它毕竟是一种面向过程的编程语言，已经无法满足运用面向对象方法开发软件的需要。为此AT&T贝尔实验室的Bjarne Stroustrup博士在C语言的基础上对其进行了改进和扩充，开发出支持面向对象程序设计的C++语言。

1.2　C++语言是一种面向对象的程序设计语言

1.2.1　C++语言和面向对象的程序设计

面向对象的程序设计是在吸取结构化程序设计的一切优点的基础上发展起来的一种新的程序设计方法，其本质是把数据和处理数据的过程抽象成一个具有特定身份和某些属性的自包含实体——对象。面向对象系统最突出特点是封装性、继承性和多态性。

1. 封装

我们知道，当一台计算机的内存条损坏后，并不需要用原料重新做一个。只要按照这个组件的接口规范，找一个成品替代它即可。这里，我们所关心的内存条只是一个“黑盒子”，只要符合规范就行，并不关心它是如何工作的。因此，在维修一台计算机时，我们并不需要看到各组件的内部构造。也就是说，各组件对于我们来说是一个自包含实体，是封装的。这种无需知道封装单元是如何工作，就能使用的思想称为数据隐藏。在这里计算机内存条的所有属性都封装在内存条对象内，用户只需按照它的引脚情况就可以使用它，无需知道内存条的工作原

理。

为了更好地模拟现实世界，在计算机编程中，引入了面向对象的思想。在面向对象的程序设计中，封装是一种数据隐藏技术，它通过把一组数据和与数据有关的操作集合放在一起形成对象来实现。对象通过操作接口与外部发生联系，而内部的具体细节则被隐藏起来，对外是不可见的。封装的目的就是防止非法访问，用户只能通过对象的操作接口利用对象提供的服务，但看不到其中的具体实现细节。

C++语言通过类来支持封装性。类是对象的抽象及描述，对象是类的实例，一个类中所有的对象都具有相同的数据结构和操作代码。在C++语言中，类是数据和其相关操作的封装体，可以作为一个整体来使用。类中的具体操作细节被封装起来，用户在使用一个已定义类的对象时，无须了解类内部的实际工作流程，只要知道如何通过其对外接口使用它即可。

2. 继承

继承这一概念在现实世界中无处不在，子女会继承父母的大部分特点，新的产品会在老型号的基础上进行改进。因为有继承，人类的文明史才会不断地延续和发展。

在面向对象的程序设计中，同样有着继承的机制，通过继承可以支持重用。通过继承，程序可以在扩展现有类的基础上声明新类。即新类是从原有类的基础上派生出来的，新类将共享原有类的属性，并且还可以添加新的特性。其中新类被称作原有类的子类或派生类，原有类称作基类，又叫父类。

C++语言支持单继承和多继承，因而具有继承这一特性所带来的优势，大大增加了程序的重用性。

3. 多态性

日常生活中我们去买水果吃，有可能会买苹果、橘子，或者其他水果，这随个人的爱好而不同。

在面向对象的程序设计中，通过多态性来支持这一思想。多态性是指相同的消息为不同的对象接收到时，可能导致不同的动作。它使得属于同一类的不同对象可以按各自的需要对同一消息做出适当的响应。

C++语言通过提供函数重载和运算符重载等来实现多态性，从而达到不同的对象各自按照自身的需求对同一消息进行正确处理的目的。

1.2.2 C++语言与C语言的关系

首先，C++语言是一个更好的C语言，因为C++语言根除了C语言中存在的问题，并保持与C语言相兼容。事实上，C++语言就是C语言的一个超集，绝大多数C语言代码无须修改就可以直接在C++程序中使用，这使得C程序员能够更容易地学习C++语言。

其次，C++语言是支持面向对象的程序设计的语言。但为了保持与C的兼容，C++语言也支持面向过程的编程，因此，C++语言并不是一个纯粹的面向对象设计语言。尽管如此，我们在用C++编程时，还是应注意采用面向对象的思维方法。C程序员需要从全新的面向对象的角度来学习C++语言，利用新技术。

C++语言是在C语言的基础上发展起来的，但它更是一次变革。两者的本质差别在于

C++语言支持面向对象的程序设计，而C语言仅仅支持面向过程的程序设计。掌握好面向对象的程序设计思想是学好C++语言的关键。

学习C++语言必须有C语言基础吗？不是的。C++语言是一个完整的语言，读者完全可以直接学习C++语言。当然，读者如果具有C语言的基础，或许会更快地成为一个好的C++程序员。

1.3 C++程序的开发步骤

C++语言是一个计算机编程语言，利用它编写的程序并不能直接在计算机上运行，而是要经过编辑、编译和链接三步生成可执行文件。

1. 编辑

是指把按C++语法规则编写的程序代码输入计算机，并存盘。在存盘时，文件的扩展名习惯上取为CPP。编辑成的文件叫源文件。

许多编辑器可用来编辑C++语言的源文件。如Windows Notepad、DOS Edit等。

2. 编译

将编辑好的C++源程序通过编译器转换成目标文件(OBJ文件)。不同编译器的使用方法各不相同，因此在使用某种编译器前应了解其正确的使用方法。

3. 链接

目标文件要和相应的C++库文件相链接才能生成可执行文件(EXE文件)。

简单地说，开发一个C++程序至少需要编写程序、编译、链接和运行程序4个步骤。但是在实际的编程过程中，编写源代码时常常会犯这样或那样的错误，有些错误会导致编译失败，有些会导致链接失败，有些错误甚至只有在运行时才表现出来。不管发现何种错误，程序员都必须改正它。这就必须重新经过编辑、编译和链接生成新的执行文件。

本书中所有的例题都是在Microsoft Visual C++ 6.0集成环境下开发的，有关这一集成环境的使用方法，读者可以参阅相关文献，这里限于篇幅就不详细介绍了。

1.4 一个简单C++程序的组成

首先，我们来看一个简单的C++程序。

【例1-1】 一个简单的C++程序。

```
#include < iostream.h >
void main()
{
        cout << "This is my first C++ program!\n";
}
```

这个程序经编辑、编译和链接便产生可运行的后缀为EXE的文件。运行这个程序，将在

屏幕上输出：

```
This is my first C++ program!
```

下面我们来比较详细地分析一下例【1-1】。

程序的第一行 #include < iostream.h > 是一条预处理指令，指示编译器将文件 iostream.h 中的代码嵌入到程序中该指令所在之处。关于预处理器的内容，本书后面将有详细的阐述，这里只须知道为了能进行输入/输出操作，程序中必须有这一行。

第二行定义了一个 main()的函数。main 是函数名，void 表示该函数没有返回值。它是程序的开始执行点，也即在程序生成可执行文件后，将在此处开始运行。其后的一对“{}”包括了该函数的全部内容，其中左大括号表示函数的开始，右大括号表示函数的结束。括号中的内容被称作函数体，它是由一系列 C++ 语句组成，这些语句描述了函数的功能实现。每个 C++ 程序中有且只能有一个名为 main()的函数，它是程序的主函数。

第四行是程序的主要部分。cout 用来向显示器输出数据，它后跟“<<”(两个“<”号)，表示将“<<”号后的数据在显示器上显示出来，这里是一个字符串。字符串必须用双引号括起来，最后的两个字符“\n”告诉 cout 在输出 This is my first C++ program! 后换行。在 C++ 语立中语句必须以“;”结尾。

与 cout 相对应，我们可以利用 cin 后跟“>>”(两个“>”号)来给“>>”右边的数据输入值。下面的程序说明了如何进行输入操作。

【例 1-2】 cin 的使用。

```
#include < iostream.h >
void main()
{
    int x,y,z;
    cout << "Please input two integers:";
    cin >> x >> y;
    cout << "x = " << x << "    y = " << y << endl;
    z = x + y;
    cout << "x + y = " << z << endl;
}
```

运行这个程序，显示如下信息：

```
Please input two integers:
```

输入两个整数 1 和 2，1 与 2 之间以空格符分隔，回车，输出如下结果：

```
x = 1      y = 2
x + y = 3
```

在这个程序中，我们还使用了 endl。在这个程序中它的作用等同于“\n”，不过“\n”是 C 语言的换行符，在 C++ 语言中保留它，仅是为了与 C 语言相兼容。有关 endl 的详细内容本书后面将有介绍，现在我们只需知道在 C++ 中它可以替代“\n”。“int”说明数据 x,y,z 是整数，有关整型“int”的使用，本章稍后将有说明。

注释信息也是 C++ 程序的一部分,较大或较复杂的程序都应加上相应的注释信息,以解释该程序,提高程序的可读性。C++ 程序注释有两种类型。双斜线"//"注释是 C++ 型注释,从"//"开始,本行中所有字符都被作为注释处理。另一种注释是 C 型注释,"/*"表示注释的开始,"*/"表示注释的结束。在"/*"、"*/"之间的字符均被视作注释。C 型注释主要用于大块注释,而且 C++ 型注释可以嵌在 C 型注释中。【例 1-2】中的程序加上注释后如下。

【例 1-3】 注释的使用。

```
/**********************************
This program adds two integers that are
entered by us, then writes the sum to screen
**********************************/
#include < iostream.h >
void main()
{
    int x,y,z;                                  //x、y、z are integer
    cout << "Please input two integers:";
    cin >> x >> y;                              //enter x,y
    cout << "x = " << x << "y = " << y << endl; //writes x,y to screen
    z = x + y;
    cout << "x + y = " << z << endl;            //writes the sum to screen
}
```

另外,需要指出的是:用 C++ 语言编程时,程序的书写非常自由,比如上例完全可以改写成如下形式:

```
#include < iostream.h >
void main() {int x,y,z;cout << "Please input two integers:";cin >> x >> y;
cout << "x = " << x << "y = " << y << endl;z = x + y;cout << "x + y = " << z << endl;}
```

甚至可以把整个函数体全部书写在一行上。但这样做,对人来说,如果程序稍长一点,就难理解了。因此,在书写 C++ 程序时,一般采用比较整齐美观的"缩进"格式来书写。这一点在以后的学习中应该注意。

1.5 C++ 语言的词法记号

词法记号是 C++ 语言的最小语法单位。C++ 语言中共有 5 种词法记号:关键字、标识符、常量、运算符和标点符号。由于字符是程序中可以区别的最小符号,在讲解词法记号前,我们应先了解一下 C++ 语言的字符集。

1.5.1 字符集

字符是构成 C++ 语言的基本要素。用 C++ 语言编程时,除字符型数据外,其他所有成分都只能由字符集中的字符构成。C++ 语言的字符集由下述字符构成:

- 英文字母:A ~ Z,a ~ z

- 数字字符:0~9
- 下列特殊字符:

空格 ! # % ^ & * _ (下划线)
+ = - ~ < > / \ |
' " ; . , () [] {} : ?

1.5.2 词法记号

1. 关键字

关键字是C++语言的保留字,好比是已经赋予特殊含义的专用单词。它们各自有不同的使用目的,在程序中不能把它们用作别的用途。下面列出了些常用的关键字,至于这些关键字的意义和用法,我们将在以后逐渐介绍。

auto	bool	break	case	catch	char	class
const	const_cast	continue	default	delete	do	double
else	dynamic_cast	enum	explicit	extern	false	float
for	friend	goto	if	inline	int	long
mutable	namespace	new	operator	private	protected	public
register	reinterpret_cast	return	short	signed	sizeof	static
struct	static_cast	switch	template	this	throw	true
try	typedef	typeid	typename	union	unsigned	using
virtual	void	volatile	while			

2. 标识符

标识符是程序员为命名程序中的一些要素所定义的单词,如变量名、函数名等。C++语言中标识符的命名规则如下:

- 由字母、数字和下划线组成。
- 以字母或下划线作为第一个字符,其后跟零个或多个字母、数字、下划线。
- 大写字母与小写字母分别代表不同的标识符。
- 不能与关键字相同。

例如:sum、Draw_rectangle_myfriend 都是正确的标识符。而下面的标识符都是非法的:

```
1_people        //起始字符非法
operator        //是关键字
my $            //含有非法字符
```

3. 常量

又称文字,是指在程序中直接使用符号表示的数据,包括数字、字符、字符串等。在本章中将有专门的篇幅介绍其详细内容。

4. 运算符

运算符是C++语言实现加、减等各种运算的符号。如:+、-、*、/等。本章也有相关的

详细说明。

5. 分隔符

在编写程序时，分隔符用于分隔词法记号或程序正文。运算符就可以用做分隔符，其他分隔符还有

() {} , : ;和空白字符。

这些分隔符不表示任何实际的操作，仅用于构造程序。其中空白字符是指空格(空格键所产生的字符)、制表符(Tab 键所产生的字符)、换行符(Enter 键所产生的字符)。由于 C++ 编译器将注释也当作空白对待，所以，注释也可用作分隔符。

1.6 数据类型

数据是程序处理的对象，在 C++ 语言中，所有的对象都属于某种数据类型。C++ 语言有着丰富的数据类型，大致可分为基本数据类型和自定义数据类型。

1.6.1 基本数据类型

基本数据类型是 C++ 语言预定义的类型，共有 4 种：布尔型(bool)、字符型(char)、整型(int)和浮点型(float、double)。其中布尔型(bool)数据的取值只能是 false 和 true，事实上，在计算机内，编译系统将 false 表示成整数 0，true 表示成整数 1，因此，也可以把布尔型看成是一个取值受限的整型。

另外，为了满足不同情况下的需要，C++ 语言还提供了 4 个关键字：signed(表示有符号)、unsigned(表示无符号)、short(表示短型)和 long(表示长型)，用来作为前缀修饰字符型、整型、浮点型。这 4 个关键字又被称作修饰符。

上述的 4 个修饰符对 int 均适用，但只有 signed、unsigned 适用于 char，long 适用于 double。在用 short 或 long 修饰 int 时，int 三个字符可省略。表 1-1 列出了 C++ 语言所提供的基本数据类型。

每种类型所占字节数随着所采用的 C++ 编译系统的不同，而有所不同，表 1-1 中所列的是采用 VC++6.0 时的情况。

表 1-1 C++ 语言的基本数据类型

类 型	长 度(字节)	取 值 范 围
bool	1	false(0) true(1)
char	1	-128 ~ 127
signed char	1	-128 ~ 127
unsigned char	1	0 ~ 255
short int(short)	2	-32768 ~ 32767
signed short int(signed short)	2	-32768 ~ 32767
unsigned short int(unsigned short)	2	0 ~ 65535
int	4	-2147483648 ~ 2147483647

（续）

类　型	长度(字节)	取值范围
signed int	4	-2147483648 ~ 2147483647
unsigned int	4	0 ~ 4294967295
long int(long)	4	-2147483648 ~ 2147483647
signed long int(signed long)	4	-2147483648 ~ 2147483647
unsigned long int(unsigned int)	4	0 ~ 4294967295
float	4	3.4E-38 ~ 3.4E+38
double	8	1.7E-308 ~ 1.7E+308
long double	8	1.7E-308 ~ 1.7E+308

1.6.2 变量

程序中的数据运行时需要存放在内存中，占据着若干字节的内存。有些数据在程序运行过程中是可以改变的，我们称之为变量。由此可见，一个变量对应着计算机中的一组内存单元，这组内存单元在C++语言中用一个标识符来标识，即变量名。在程序中能很容易地通过变量名来操作相应的内存单元，存取或修改其中的数据，而不需要知道它的实际内存地址。

C++语言中使用变量前必须对变量进行声明。变量声明的格式如下：

数据类型 变量名1,变量名2,…,变量名n;

其中数据类型是指C++语言中的任一合法类型，每个变量都必须属于一种类型。变量名的命名应遵照1.5节中标识符的命名规则，另外，在命名变量时应尽量做到“见名知意”，以增加程序的可读性。例如：

```
int i;
```

就声明了一个整型变量i，程序在运行时将需要安排4个字节的内存单元来存放它。

下面再举两个例子：

```
char c;
float width,length,area;
```

它们都是合法的变量声明。

在声明一个变量的同时，我们也可以给它赋初值。例如：

```
int i=1;
```

就在声明整型变量i的同时，把它的值初始化为1。

正如一次可以同时声明多个变量，C++语言在一个语句中也可以初始化多个变量，而且一个语句中的变量也没有必要全部初始化。如下语句都是合法的。

```
int i=1,j=2;
double width=5.6,length,area=50.8;
```

在声明变量的同时赋初值还有另外一种形式，如：