面向21世纪 交通版

高等学校教材

仝秋红　编著

C++语言教程

人民交通出版社
China Communications Press

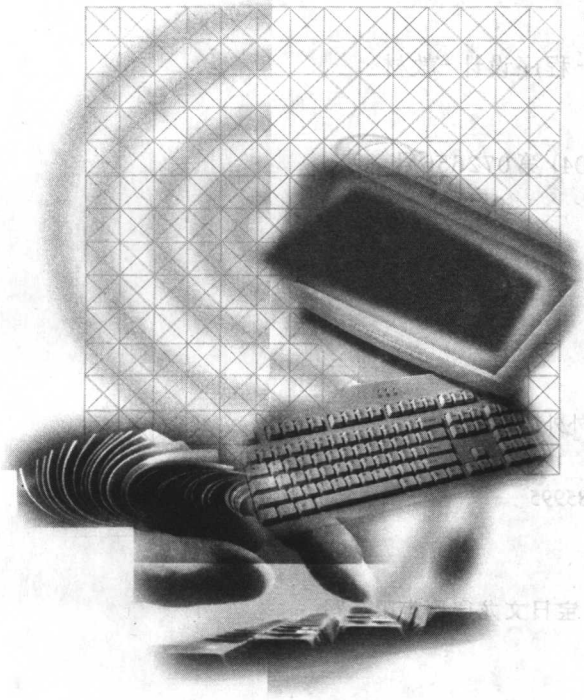
面向21世纪交通版

高等学校教材

C++语言教程

C++ YUYAN JIAOCHENG

仝秋红　　编著



人民交通出版社
China Communications Press

内 容 提 要

本书详细介绍了C++的基本语法，包括类与对象、类的继承性、多态性与虚函数、运算符重载、类对象的复制与转换、I/O流、C++模板、C++异常处理等。

本书介绍理论的同时也注重应用，其中包括一些MFC应用程序的设计举例，并备有作业题和练习题。

本书可作为高等院校理工科本科生、研究生的教材，也可作为软件编程人员的参考用书。

图书在版编目(CIP)数据

C++语言教程/仝秋红编著.—北京：人民交通出版社，2004.8
ISBN 7-114-05166-2

Ⅰ.C… Ⅱ.仝… Ⅲ.C语言-程序设计-教材
Ⅳ.TP312

中国版本图书馆CIP数据核字(2004)第072656号

书　　名：C++语言教程
编 著 者：仝秋红
责任编辑：刘敏嘉
出版发行：人民交通出版社
地　　址：(100011)北京市朝阳区安定门外外馆斜街3号
网　　址：http://www.ccpress.com.cn
销售电话：(010)85285656，85285838，85285995
总 经 销：北京中交盛世书刊有限公司
经　　销：各地新华书店
印　　刷：三河市海波印务有限公司－宝日文龙印刷有限公司
开　　本：787×1092　1/16
印　　张：15.25
字　　数：378千
版　　次：2004年8月第1版
印　　次：2004年8月第1版第1次印刷
书　　号：ISBN 7-114-05166-2
印　　数：0001－3000册
定　　价：24.00元
(如有印刷、装订质量问题的图书由本社负责调换)

前　言

面向对象的程序设计技术是目前最热门、最实用的软件开发技术。与传统的面向过程的程序设计技术不同，面向对象程序设计技术对现实世界的对象进行建模。它应用了类关系，在这种关系中，特定类的所有对象都有相同的特征。它还应用了继承关系、多重继承关系，新对象类是利用已有类的特征建立起来的，同时还增加了自身的特征。专家预测，至少在今后的几十年，面向对象编程都会是一种最重要的编程手段。

本书在编写方法上，力图适应各种具有不同计算机软件基础的读者。如果读者没有任何的软件编程经验，则应从本书的“绪论”开始阅读，先熟悉程序设计思想，再了解C++的语法，最后再接触面向对象程序设计。如果读者已经掌握一二种除了C、C++之外的其它高级语言，可以大致了解一下本书中前9章的内容，转入C++的语法，之后从第11章开始，介入面向对象程序设计。而如果读者已经掌握了C语言，可以直接从本书的第10章开始，了解C++的新特点，之后进入面向对象程序设计。

本书是在作者多年从事C及C++教学的基础上编写的，由于C++是C语言的扩充，所以本书中第9章以前的内容，重点介绍了C++从C语言继承而来的面向过程部分，包括第2章的数据类型、运算符与表达式；第3章的逻辑运算和判断选取控制；第4章的数组；第5章的函数；第6章的编译预处理；第7章的指针；第8章的结构体、共用体、枚举类型；第9章的位运算。对于已经扎实掌握了C语言的读者可以跳过这些章节，直接从第10章开始阅读。第10章——C++新特点，着重介绍了一些在C++中新引入的概念，包括引用、常量类型、内联函数、new与delete运算符等。第10章以前的内容是从事C++面向对象程序设计之前必须要具备的基础知识，而其后的内容，包括第11章的定义C++类；第12章的派生C++类；第13章的虚函数与多态性；第14章的重载、复制和转换；第15章的C++ I/O流；第16章的C++模板以及第17章的C++异常处理则全面介绍了C++的有关知识。

书中的所有程序均在Visual C++6.0上调试通过，一些例题中的MFC应用程序在附录列出。

在本书中第15章、第16章、第17章、第11章第10节链表的编写中，陈兴旺提供了许多宝贵的资料，在此表示感谢！本书的习题部分由陈兴旺编写。

本书既可作为大专院校C++语言教材，亦可作为从事C++语言软件开发者、计算机应用人员参考书。

本书在编写过程中，查阅了许多的中、外参考书目及资料，在此对这些书的作者和译者表示感谢。由于编者水平有限，书中难免有错，希望广大读者提出宝贵意见。

谢谢阅读本书的读者！

编　者

2004年5月

目　录

第1章 绪 论

相信大家对“面向对象”这个词已经非常熟悉了。目前,最为流行的程序设计技术就是面向对象的程序设计技术,而C++语言之所以受到如此广泛的欢迎,其中一个重要的原因就是C++是一种面向对象的程序设计语言。

在C语言出现之前,几乎所有的程序设计语言都是面向过程的(如那时的FORTRAN语言、BASIC语言)。过程性语言的一个最大问题就是,程序在执行一条语句时,和前后的过程非常相关。比如,那时我们常常遇到的一个问题就是,在定义变量时,常常担心是否会和程序中的另一个变量重名;另外,在程序调试中,修改了一条语句,就会影响到整个程序。之后的C语言,实现了结构化的程序设计,程序是由一个个函数组成的,而每个函数都是独立的代码块,这样在函数中定义变量时,不用担心与另一个函数中的变量重名,而每一时刻只有被调用的函数在内存中。另外,C语言还有一个非常重要的特点就是可以访问内存。因此,C语言一产生,就马上得到了较广泛的应用,但C语言仍然是面向过程的程序设计语言。令人欣慰的是,C语言的这些优点,都继承到了C++中,而更重要的是,C++是面向对象的程序设计语言。

在C++中,所有的任务可以封装在类中,这包括完成任务所需要的变量,以及对这些变量的操作。由于具有继承的特点,相关的任务可以组合在一个树形的目录下,而用一条指令去完成不同的任务,这就是它的多态性。对于软件开发者来说,无论是在软件的整体设计规划,还是在软件的开发过程中,面向对象的程序设计都带来了很大的便利。采用面向对象程序设计的C++写出来的程序更容易理解,更加规范,更容易维护、修改和调试。

1.1 面向对象程序设计

面向对象程序设计系统主要包含三个要素:对象、类和继承性。

1.1.1 对象

一般认为,对象就是一种事物,一个实体。

从概念上讲,对象代表着正在创建的系统中的一个实体。

从实现形式上讲,对象是一个状态和操作(或方法)的封装体。

1.1.2 类

类是创建对象的样板,它包含着所创建对象的状态描述和方法的定义。

类的完整描述包含了外部接口和内部算法以及数据结构的形式。

实际上,类就是对自然界的某一事物其共同特点的抽象描述,所以,类就是抽象。而对象则是这一事物的具体实现,因此,对象就是具体。

如:“矩形”,这就是一个抽象的概念,它表示有长、宽,平面的二维图形。“矩形”的概念可以用类来描述。

而“长20米,宽10米的矩形”,就是一个具体的矩形,可以用对象来实现。

定义了类以后,就可以用类来定义具体的对象,因此,我们把对象也称做类实例。

1.1.3 继承

继承提供了创建新类的一种方法，一个新类可以通过对已有类进行修改或扩充来满足新类的要求。

如果新类是从已有类继承过来的，则新类称做派生类，原来已有的类称做基类。派生类中包含了基类中的所有元素，除此之外，还有自己新的特点。

如："圆角矩形"可以由"矩形"继承而来，它除了具有"长"和"宽"这样的特点外，还包括"四个圆角"这样的特点。

1.2 C++是面向对象的程序设计语言

1.C++支持数据封装。

类是支持数据封装的工具，对象则是数据封装的实现。

类将数据和对该数据进行合法操作的函数封装在一起。

2.C++的类中包含私有、共有和保护成员。

私有成员，只有类中的函数可以访问。共有成员，类内类外的函数都可以访问，它是该类的接口。保护成员，只有该类的派生类可以访问 。

3.C++中通过发送消息来处理对象。

对象的初值由构造函数来设置，不同的构造函数向对象传递不同的类型消息。

4.C++中允许友元破坏数据的封装性。

定义友元函数和友元类可以访问类中的私有成员。

5.C++的多态性。

C++的多态性体现在，用虚函数实现函数的动态联编、函数的重载、运算符的重载。

6.C++的继承性。

由基类继承，产生派生类。类的结构可以很复杂，可以是单继承的树形结构，也可以是多继承。

7.C++支持动态联编。

实现动态联编的函数的实际地址是在程序运行时确定的。

1.3 C++中保留的关键字

关键字是系统已预定义的单词，它们在程序上都有着不同的用途。

C++中保留了C语言中的32个关键字，除此之外，还增加了新的关键字。下面为C++中常用的关键字：

auto	break	case	char	class	const	continue	default
delete	do	double	else	enum	explicit	extern	float
for	friend	goto	if	inline	int	long	mutable
new	operater	private	protected	public	register	return	short
signed	sizeof	static	static _ cast	struct	switch	this	typedef
union	unsigned	virtual	void	while			

关键字都属于保留字，用户不可再重新定义。

1.4 C++程序结构的特点

首先,我们看一个简单的C++程序。

```
#include "iostream.h"                          //文件包含命令
int sum(int a,int b,int c);                    //函数说明语句
void main()                                    //主函数
{
  int x,y,z,s;                                 //变量定义
  cin >> x >> y >> z;                          //变量的输入
  s = sum(x,y,z);                              //函数的调用
  cout <<"the sum of" << x <<"," << y <<"," << z <<"is:" << s << endl; //变量的输出
}
int sum(int a,int b,int c)                     //函数定义
{
  int s;
  s = a + b + c;
  return s;
}
```

运行程序,先输入:

2

3

4

程序输出:

the sum of 2,3,4is:9

程序中,//……为注释语句。

在C++程序的开头,常常是预处理命令,这些命令以“#”开头。

#include "iostream.h"是预处理命令中的文件包含命令,表示将文件 iostream.h 包含到本文件中。iostream.h 是一个和输入输出流有关的文件,只要程序中有输入输出该文件都要包含进来。

main()函数也叫作主函数,程序从 main()开始,也从 main()函数结束。

```
{  …
   …
}
```

花括号中的内容是函数体的内容,函数体由语句组成,每条语句以“;”结束。

cin >> 、cout << 代表对变量的输入、输出,而 endl 表示换行。

cin、cout 为流操纵算子,>>、<<分别称做流提取与流插入运算符。

程序经过编译、连接,生成 .exe 文件后,就可运行了。

第2章　数据类型、运算符和表达式

C++具有丰富的数据类型(图2-1),使得C++在处理各种数据类型时非常方便。除了基本数据类型可以非常方便地互相转换,基本类型也可通过构造函数转换为类对象(第13章)。

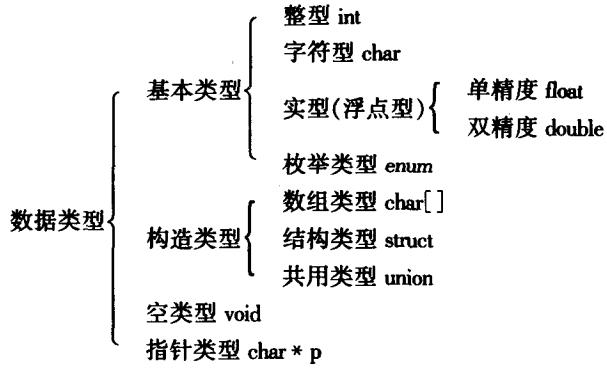


图2-1　数据类型

2.1　整型数据

2.1.1　整型常量

常量:其值不能被改变的量。例如:

1.十进制:123。

2.八进制:以"0"开头,0123。

3.十六进制:以"0x"开头,0x123。

2.1.2　变量

变量:其值能发生变化的量。

在设计程序之前,首先对程序中要使用的变量进行定义,在定义变量时要注意以下三点:

1.变量名由字母、数字、下划线三种字符组成。第一字符必须为字母或下划线,如:agreet _ 1。

2.变量名一般用小写母表示。有些系统将大小写认为是两个不同的变量名,例如,Sum和sum为不同变量名。

3.对变量"先定义,后使用"。

定义为某种类型后,则在编译时按照其类型为其分配相应的存储单元,未被定义的变量名,不能使用。

2.1.3　整型变量

1.基本型:用int表示,占4个字节,32位。

2.短整型:用short int或short表示,占2个字节,16位。

3.长整型:用long int或long表示,占4个字节,32位。长整形常量的表示方法为,数字后加上一个字母l或L,例:1231,423L,0l。

4.无符号型:用 unsigned int,unsigned long, unsigned 表示,存放不带符号的整数。存放数的范围比一般整型变量中数的范围扩大一倍。

注:有符号与无符号的区别是对整数最高位的解释,有符号型数编译时生成的代码将整数最高位作为符号标志,0 为正,1 为负。无符号型数在存放时最高位为数据位,用来存放数据。

2.2 实数数据

2.2.1 实型常量

实型常量有两种形式:

1.十进制数形式:如 0.123, 1.23。

2.指数形式:如 123e3。

2.2.2 实型变量

1.单精度型:float,占 4 个字节,6 位有效数字。

2.双精度型:double,占 8 个字节,12 位有效数字。

3.长精度型:long double,占 10 个字节,15 位有效数字。

然而,实型常数无论是十进制形式还是指数形式,都把它作为 double 形式,若要表示 float 型实常量,可在后加 f,如:3.1415926f。

2.2.3 实数在内存中的存储形式

实数亦称为浮点数,其在内存中的存放是以浮点的形式存放的。以 float 为例,在 4 个字节 32 位中,包括符号部分、指数部分、小数部分,如图 2-2 所示。

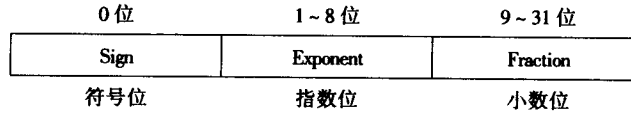


图 2-2 实数的存储形式

2.3 字符型数据

2.3.1 字符常量(单字符常量)

1.用单引号括起来的一个字符。例如:

'a','b','?','$'。

2.以"\"开头的字母或数字序列,反斜线代码(见表 2-1)。例如:

'\n' /换行

'\t' /横向跳格

'\101' /字符'A'

C++ 中常用的反斜线代码 表 2-1

符 号	含 义	符 号	含 义
\a	响铃	\\	反斜线
\n	换行符	\'	单引号
\r	回车符	\"	双引号
\t	水平制表符(Tab)	\0	空字符
\b	退格符(Backspace)		

2.3.2 字符变量

字符变量在定义时的标识符为 char,占一个字节的空间。例如:

```
char c1,c2;
c1 = 'a'; c2 = 'b';
```

注:只能放一个字符。

C++中,字符型变量中只能存放单字符常量,不能存放字符串,字符串是由字符数组来处理的。

2.3.3 字符数据在内存中的存储形式

字符型变量在内存中存放时,将字符所对应的 ASCII 码以二进制的形式存放在内存中,占一个字节。

如:char c1,c2;

c1 = 'a'; c2 = 'b';

则其存储形式如图 2-3 所示。

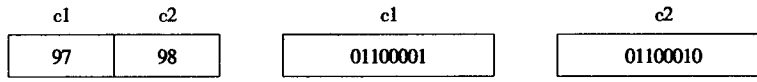


图 2-3 字符在内存中的存储形式

由于字符型数据是这样的存储方式,故字符数据可进行算术运算。进行算术运算时要先将字符转化为对应的 ASCII 码,再做运算。

例 2.1

```
#include"iostream.h"
void main()
{
    char c1 = 'a',c2 = 'b';
    c1 = c1 - 32;c2 = c2 - 32;
    cout << c1 << c2 << endl;
}
```

输出:A B

2.3.4 字符串常量

用双引号括起来的字符序列,如:

"how do you do"、"china"。

字符串存放在字符数组中,与普通字符数组不同的是,在每一个字符串的末尾,都隐含有一个字符'\0'。

\0:字符串结束标志,隐含在字符串的末尾。故:'a'与"a"不同。

2.3.5 简单变量赋初值

简单变量赋初值的方法有:

1.在定义变量时赋值。如:

```
int a = 3;或 int a(3);
```

2.先定义,后赋值。如:

```
int a;
a = 3;
```

未被初始化的变量在第一次被赋值前是不定的。

3.一些赋值,一些不赋值。如:

```
int a,b,c=5;
```

2.3.6 数据类型运算时的转换

不同类型的数据在运算时,其运算结果遵循以下转换规则。

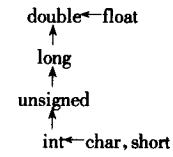


图 2-4 数据类型运算时的转换

从图 2-4 可以看出,double 类型级别最高,char 类型级别最低。当 float 与 long 、unsigned 、int、char 等相遇时,转换为 double。

例 2.2

```
#include"iostream.h"
void main()
{
    char ch='a';
    int i=2;
    float f=2.1f;
    double d=3.5;
    double result;
    result=(ch/i)+(f*d)-(f+i);
    cout <<result;
}
```

例 2.2 中,ch/i 为 int 型,f*d 的结果为 double 型,f+i 的结果为 double 型,故表达式(ch/i)+(f*d)-(f+i)的结果为 double 型,故在定义变量 result 时应将其定义为 double,这样,才能保证正确的将表达式的值赋值给 result。

2.4 算术运算符和算术表达式

运算符的范围很广,除控制语句和输入输出外所有的基本的操作都作为运算符处理。并且,在面向对象的程序设计中,运算符还可以重载,例如,位运算符 << 、>> ,赋值运算符 =。

C++有三大类运算符,即算术运算符、关系与逻辑运算符、位运算符。另外还有一些特殊的运算符和表达式。本节将介绍算术运算符和其它一些常用的运算符。

2.4.1 算术运算符和算述表达式

2.4.1.1 算术运算符

算术运算符是做各种基本的算术运算时所用到的运算符,共有以下几种:

+,-,*,/,%(求余运算符,模运算符)。

注:%两侧要求为整型。例如 7%4 结果为 3。

例 2.3

```
#include"iostream.h"
void main()
{
    int x,y;
```

```
    x = 1;
    y = 3;
    cout << x/y << '/' << x%y;
}
```

结果:0/1

注意:除法'/'用于整数和字符时结果取整。

例 2.4

```
# include"iostream.h"
void main()
{
    int a = 100;
    char b = 'a';
    int c;
    c = a/b;
    cout << c;
}
```

结果为 1。

2.4.1.2 算术表达式、运算符优先级与结合性

算术表达式:用算术运算符和括号将运算对象连接起来的式子。

算术表达式是C ++ 中使用最多的表达式之一。例:a + b - 5 * 3 + 'b'。

优先级:先乘除,后加减。

结合性:从左至右(结合方向,代表表达式执行的顺序)。

注意:括号的作用为强迫升高某个运算符或某组运算的优先级。

2.4.2 强制类型转换

当某一变量定义后,其类型就不能改变,但在程序运行当中,需要临时将一表达式或变量转换成需要的类型时,可使用强制类型转换。使用的形式为:

(类型名)(表达式)

```
(double)a ;
(int)(x + y);
(float)(5% 3);
```

注:强制类型转换是运算符,它得到的是一个中间变量,而原变量类型并未发生变化。

例 2.5 用一整数控制循环,但在执行时又要保存小数部分。

```
# include"iostream.h"
void main( )                          //print i and i/2 with fractions.
{
    int i ;
    for(i = 1;i <= 100;i ++ )
    cout << i << ' ' << (float)i/2 << ' ';
}
```

上面例子中将 i 临时强制转换为 float,就可保留除法运算小数点后面的数字。

强制类型转换虽不经常使用,但相当实用。

2.4.3 自增、自减运算符

自增、自减运算符为 ++ 、— 运算符,可放在变量前,也可放在变量后,但放在前后的作用不同。

++ i, — i:先使 i 的值加(减)1,之后再使用 i。

i ++ , i — : 先使用 i,之后再使 i 的值加(减)1。

++ 、— 运算符在使用时要注意以下几点:

1.设:i = 3;

则:

y = ++ i;结果为 y = 4 ,i = 4

y = i ++ ;结果为 y = 3,i = 4

2.不能用于常量和表达式,或 float、double 型变量。例如,下列均为错误语句:

5 ++ ,(a + b) ++ ,float a = 2.1;a ++ ;

3.结合方向自右向左

cout << - i ++ ;

若 i = 3,输出 - 3,后 i = 4。

4.常用于循环语句,指针变量。

2.5 赋值运算符和赋值表达式

2.5.1 赋值运算符

赋值运算符: =

例如,语句　　a = 3,c = a + b;

左边必须是变量,不能为函数或常量。

C ++ 中,赋值操作是作为运算符来处理的,并没有专为赋值而规定的语句。因而赋值号可以出现在表达式中。例如:3 + (i = 5)/6 ,是合法的表达式。

2.5.2 类型转换

当赋值运算符在操作时,赋值号左右两端会出现类型不一致的情况,左右两端类型不一致,则代表数据在内存中的存放方式不一样,这样,在赋值时会有一些特殊的情况。下面将常见的几种情况说明一下。

1.int 型数据 = float 型数据:舍去 float 中小数部分。

2.float,double 型数据 = int 型数据:数值不变,将 int 型数据转换成浮点形式赋值到左边变量中。

3.int 型数据 = char 型数据:

1)字符为无符号量,将字符 8 位放到整型变量低 8 位,高 24 位补零。

2)字符为有符号量:

(1)字符的最高位为 0,将字符 8 位放到整型变量低 8 位,高 24 位全补零。

(2)字符的最高位为 1,将字符 8 位放到整型变量低 8 位,高 24 位全补 1。

这种操作称做符号位的扩展。

4.long、int 型数据 = short 型数据:

(1)short 型数据为无符号量,则将 short 型数据 16 位放到 long 、int 型变量低 16 位,高 16 位