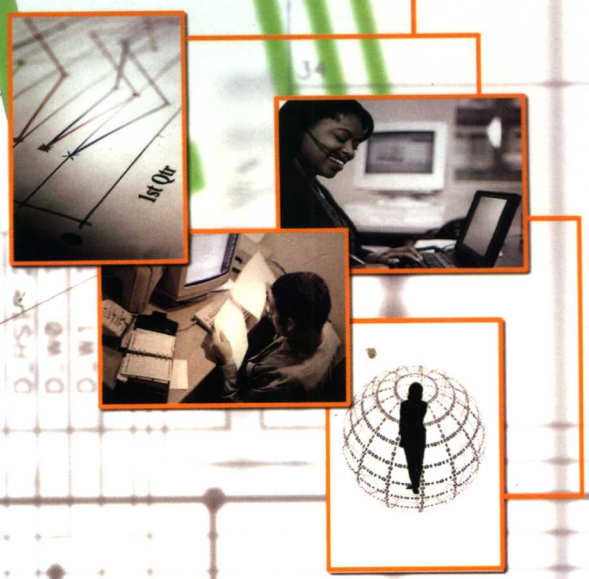
高职高专系列教材

C++程序设计实用教程

主　编：李美村

副主编：朱永玲　胡　多
邱启声

参　编：张国强　谭咏梅
邬厚民



广东高等教育出版社

高职高专系列教材

C++程序设计实用教程

主　编　李美村

副主编　朱永玲　胡　多　邱启声

参　编　张国强　谭咏梅　邬厚民

广东高等教育出版社

·广州·

图书在版编目（CIP）数据

C++程序设计实用教程/李美村主编. —广州：广东高等教育出版社，2005.8
（高职高专系列教材）
ISBN 7-5361-3193-3

Ⅰ. C… Ⅱ. 李… 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（2005）第084418号

出版发行	广东高等教育出版社 地址：广州市天河区林和西横路 邮政编码：510500 电话：（020）87557232
印　刷	广州市新明光印刷有限公司
开　本	787毫米×1 092毫米 1/16
印　张	17.25
字　数	442千字
版　次	2005年8月第1版
印　次	2005年8月第1次印刷
印　数	1~3 000册
定　价	27.00元

前　言

《C++程序设计实用教程》是为适应当前高职高专教学需要编写的实训型教材，本书主要针对高职高专学生的特点，着重培养学生的分析问题和解决问题的能力，内容由浅入深，简明易懂，力求将复杂的概念用简洁的语言描述出来，并且体现了“教材与实验指导书”有机结合的实训型教材新体系的特点。在本教程中，编入了大量实例和操作题，着重实操技能的培养，使读者学习此书后即能用C++语言编写实际应用程序。该书既适合作为高职高专计算机及相关专业的教材，也可供其他高校计算机和相关专业及计算机应用人员学习参考。

本教程由广州大学科技贸易技术学院李美村、朱永玲、谭咏梅、邬厚民，广州大学纺织技术学院胡多、张国强，广州大学城建技术学院邱启声编写。李美村任主编，朱永玲、胡多、邱启声任副主编，张国强、谭咏梅、邬厚民参加编写。其中第一章由谭咏梅编写；第二章由邬厚民编写；第三章、第七章由朱永玲编写；第四章、第六章由李美村编写；第五章由李美村、朱永玲编写；第八章、第九章由邱启声编写；第十章由张国强编写；第十一章由胡多编写。

由于时间紧迫，成书仓促，书中难免有欠缺和错漏之处，敬请读者批评指正。

编　者

2005年6月2日

目　录

第一章　C++语言概述

C++语言是一种应用较广的面向对象的程序设计语言，面向对象的设计与面向过程的设计有很大的区别，面向过程的程序设计是功能的分解，而面向对象的程序设计是对象加消息。

1.1　面向对象的基本概念

面向对象方法是利用抽象、封装等机制，借助于对象、类、继承、消息等概念进行程序的设计。对象是现实世界中客观存在的事物，对象由对象名、对象属性和对象操作构成，在面向对象程序设计中，用类来定义一种对象类型，类描述了属于该类型的所有对象的属性和操作，并且通过封装将一部分属性和操作隐藏起来，另一部分提供为用户的访问接口。继承所表达的是类之间的相关关系，这种关系使得某类对象可以继承另外一类对象的特征和能力。面向对象程序中的一切操作，都是通过向对象发送消息来实现的，对象接到消息后，启动相关方法，完成相应的操作。

1.1.1　对象与类

1. 对象

对象是现实世界中客观存在的事物，是我们认识世界的基本单元，它既可以是一种有形的具体存在的人或物，也可以是一种无形存在的事件。例如，一个学生、一张桌子、一本书、一间教室都是一个对象；一堂课、一场球赛、一次毕业答辩也都可以是对象。

每个对象都有自己的属性和行为。对象的属性又称为对象的状态，如学生的学号、姓名、性别、出生年月、身高、体重等都是一个学生对象的属性；对象的行为又称对象的操作，如上学、做作业、吃饭、跑步、睡觉等都是一个学生对象的行为。

在C++中，对象是一个状态和操作的封装体。对象实现了信息隐藏。

对象属性	对象操作
姓名：王冠 性别：男 班级：计算机1班 C++成绩：80	显示姓名 显示性别 显示班级 显示成绩

对象属性	对象操作
姓名：张扬 性别：男 班级：计算机1班 C++成绩：60	显示姓名 显示性别 显示班级 显示成绩

对象属性	对象操作
姓名：李丽 性别：女 班级：计算机1班 C++成绩：70	显示姓名 显示性别 显示班级 显示成绩

图1-1　学生信息

2. 类

类是对多个对象进行综合抽象的结果，类是创建对象的样板，即一个类可以创建多个具有相同属性和操作的对象，一个对象是一个类的实例。

例如，图1-1中的3名学生，都是用同样的属性来描述，可以进行同样的操作，他们可以构成一个学生类，而王冠、张扬、李丽都是这个类的一个实例。

1.1.2 抽象与封装

1. 抽象

C++中的抽象是指对每一类对象进行概括，抽出这类对象的公共性质（包括属性和操作），然后用计算机语言加以描述的过程。例如，图1-1中的王冠、张扬、李丽3名同学都用相同的属性描述，都有相同的操作，就可抽象为一个类，而王冠、张扬、李丽3名同学都是这个类的一个对象。

2. 封装

在面向对象程序设计中，封装实现的是信息隐藏，封装可以使得一部分属性和操作隐藏起来，不允许操作者访问，而另一部分则作为类的外部接口，提供给操作者访问。

封装一般可以理解为：

（1）有一个清楚的边界，对象的属性和操作被限定在此边界内。

（2）有一个外部接口，接口规定能向特定的对象发出什么请求（如图1-2）。

（3）需保护的属性和操作要隐藏。其他对象不能随便修改。

类名	属性	接口
Student	No Name Sex Score	Disp ( ) FindNo ( ) FindName ( ) FindScore ( )

图1-2

从图1-2中可以看到，Student类的所有对象的属性和操作都被封装起来了，一般用户不能直接修改对象的属性，但可通过提供的接口函数访问对象的属性。

1.1.3 继承性

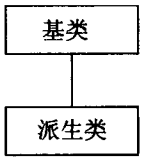


图1-3

继承是面向对象程序设计中用于创建新类的一种方法，即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。新类共享已有类的行为，本身还可添加一些自己的行为。继承的本质特征是行为共享。新类被称为已有类的子类或派生类，而已有类被称为父类或基类（如图1-3）。

1.1.4 多态性

多态性也是面向对象程序设计的一个重要特性，多态性是指不同的对象收到相同的消息时产生多种不同的行为方式。C++中多态性是通过重载和虚函数来实现的。例如，求最大值，要求两个数的最大值，也要求三个数的最大值，可以用同一函数Max ( )来实现，这种实现称为函数重载。

1.2 一个简单的C++程序

例1-1是一个用C++编写的例子，其功能是在屏幕上显示“I am a student.”，其程序

代码如下。

〖例 1-1〗

```
# include < iostream. h >          /* 包含头文件 */
void main ( )                      // 程序入口函数
{                                  //程序开始
    cout <<" I am a student.";     //在屏幕上输出
}                                  // 程序结束
```

运行结果为：

I am a student.

C++中是以 iostream. h 文件作为标准输入输出头文件，采用符号“ <<”作为标准输出。

1.3 C++的词法符号及规则

1.3.1 词法符号

C++的字符集由下列字符组成：

1. 大小写英文字母

a ~ z 和 A ~ Z

2. 数字字符

0 ~ 9

3. 特殊字符

空格 ! # % ^ & * _（下划线）

- + [ ] = ~ < > / \ |

. , : ; ? ' " ( ) [ ]

{ }

1.3.2 词法规则

C++的字符集可用于组成 C++的词法单词。一般有以下六种。

1. 标识符

标识符是由程序员定义的，用以命名程序中的一些实体。常用于命名，如函数名、类名、数组名、变量名、常量名、对象名、标号名、类型名等。

C++规定，标识符是由大小写字母、数字字符（0 ~ 9）和下划线组成，且必须以字母或下划线开始，其后跟零个或多个字母、数字字符或下划线。

定义标识符时应注意：标识符区分大小写，如 a 与 A，XYZ 与 Xyz 等都是不同的标识符；定义标识符时，不要采用系统的保留字，保留字是指系统已预定义的标识符，它包含关键字和设备字等。例如，合法标识符：Year，Day，ATOK，x1，_CWS，_change，_to。不合法标识符：# 123，. CUM，$ 100，1996Y，1 - 2 - 3 - ，int。

2. **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，用户不可再重新定义。例如：

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	operator	private	protected	public	register	return	short
signed	sizeof	static	static _cast	struct	switch	this	typedef
union	unsigned	virtual	void	while			

3. **运算符**

运算符实际上是系统预定义的，作用于被操作的对象的函数名字。根据运算符所操作的对象个数不同，可分为单目运算符、双目运算符和三目运算符。

4. **分隔符**

分隔符是用来分隔单词或程序正文的。在 C++中，常用的分隔符有空格符、逗号、分号、冒号等。这些分隔符都是用来构造程序的。

5. **常量**

常量是在程序中直接使用符号表示的数据。在 C++中，常量有数字常量、字符常量、字符串常量等。

6. **注释符**

注释在程序中仅仅起到对程序的注解和说明的作用，注释的目的是为了便于阅读程序。在 C++中，采用两种注释方法：一种是使用“/*”和“*/”括起来进行注释。在“/*”和“*/”之间的所有字符都被作为注释符处理，这种方法适用于有多行注释信息的情况。例如：/* 包含头文件 */（见例 1-2）。另一种是使用“//”注释。从“//”开始，直到它所在的行尾，所有字符都被作为注释处理。这种方法用来注释一行信息。例如：//程序入口函数（见例 1-2）。

1.4 C++程序结构的特点

1.4.1 C++程序的组成

用一个程序来求两个浮点数之和，这两个浮点数是从键盘上输入的，求得的和由屏幕输出显示。该程序源代码如下。

〖**例 1-2**〗

```
/*这是我们第二个 C++程序                    } 程序注释
 *输入任意三个整数，输出其中的最大值
 */

    #include <iostream.h>                   } 预处理命令
```

```
int max (int a, int b, int c)
{
int MAX;
    MAX =a;
    if (MAX <b)
       MAX =b;
    if (MAX <c)
       MAX =c;
    return MAX;
}
void main ( )
{
    int a, b, c;
    cin >>a >>b >>c;
    cout <<"MAX =" << max(a, b, c) <<endl;
}
```

函数定义

主程序

执行该程序，输入以下三个数字：

8 9 10

按回车键后，输出如下结果：

max = 10

从例 1 - 2 的执行程序可了解 C++程序的组成部分包括：

（1）一个 C++程序可由若干个文件组成，每个文件又是由一个或若干个函数组成，但必须包含一个且只能包含一个名为 main 的函数（又称主函数），程序总是从 main 函数开始执行。

（2）函数与函数之间是相对独立且是并行的，函数之间可以互相调用，函数的调用可以嵌套，但任何函数不能调用主函数。

（3）C++程序中的函数可分为两大类，一类是用户自定义函数，另一类是标准函数。

（4）函数名前的标识符规定了函数返回值的类型。void 表示不带回返回值。

（5）由一对花括号括起来的部分是函数体，其中语句实现程序的预定功能。

（6）函数可由一个或多个语句构成，C++的每个基本语句都以“;”结束。

（7）cout 是 C++中的标准输出流对象（显示器），“ <<”是 cout 中的运算符，表示将其后的参数输出到计算机显示器。

（8）cin 是 C++中的标准输入流对象（键盘），“ >>”是 cin 中的运算符，表示从键盘读入数据存放到其后的参数中。

（9）endl 表示回车换行。

（10）# include 语句是编译预处理语句，格式有两种：# include < 文件名. 扩展名 > 或 # include “文件名 . 扩展名”。

（11）有两种注释方法：/ *…… * /一般用于多行注释；//注释内容一般用于单行注释。

（12）一个面向对象的 C++程序，一般是由类的声明和类的使用两大部分组成。

1.4.2 C++程序的书写格式

C++程序的书写格式基本上与C语言程序书写格式相同。其基本原则为：一是一行一般写一条语句。短语句可以一行写多个。长语句可以一条写多行。二是要尽量提高可读性。应采用适当的缩格书写方式。表示同一类内容的语句行要对齐；例如，一个循环的循环体中各语句要对齐，同一个if语句中的if体内的若干语句或else体内的若干语句都要对齐。

〖**例1-3**〗分析下列程序的输出结果。

```
# include < iostream. h >
void
main ( )
{
    int i,
sum = 0;
    for (i = 1;; i ++) {
    if (i > 100)
    break;
    sum + = i;}
    cout << " sum = "
 << sum
 << endl;}
```

例1-3的程序很简单，但是由于书写方法不当，使得阅读起来比较困难，难以分析输出结果。将同样这个程序书写成例1-4中的程序，分析起来就容易多了。

〖**例1-4**〗分析下列程序的输出结果。

```
# include < iostream. h >
void main ( )
{
    int i, sum = 0;
    for (i = 1;; i ++)
        {
        if (i > 100)
            break;
        sum + = i;
        }
    cout << " sum = " << sum << endl;
}
```

该程序是用来求得自然数1～100的和，执行结果输出：

sum = 5050

通过例1-3和例1-4对同样一个程序作了两种不同形式的书写，可以看出正确的书写方法会提高程序的可读性。因此，C++程序的书写要尽量提高可读性。

1.5 C++程序的实现

C++语言有多种，如 Visual C++(VC++)、Borland C++等。C++源程序的实现是和其他高级语言源程序实现的原理一样的。一般都要经过图 1-4 所示的过程。

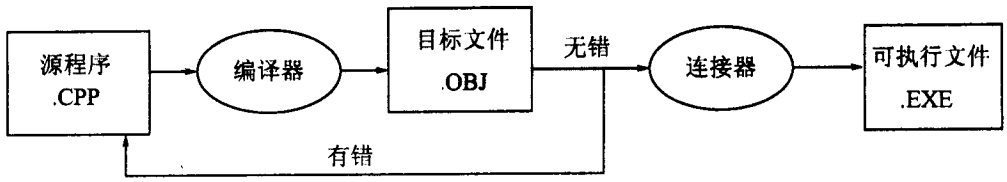


图 1-4

1. 编辑

编辑是将编写好的 C++源程序输入到计算机中，生成磁盘文件的过程。C++程序的编辑可以使用计算机软件所提供的某种编辑器进行编辑，将 C++程序的源代码录入到磁盘文件中。磁盘文件的名字要用扩展名 .CPP。

2. 编译

编译主要进行的是对程序的语法结构的分析。分析过程中，发现有不符合要求的语法错误，及时报告给用户，显示在屏幕上，最终生成目标代码，将这些代码以 .OBJ 为扩展名存在磁盘文件中，称为目标代码文件。

3. 连接

编译后的目标代码文件还不能由计算机直接执行。一个程序可能有多个源文件，编译后这些源文件的目标代码文件还分布在不同的地方，因此需要把它们连接到一块。即使该程序只有一个源文件，这个源文件生成的目标代码文件还需要系统提供的库文件中的一些代码，因此，也需要把它们连接起来。连接器将由编译器生成的目标代码文件和库中的某些文件连接处理，生成一个可执行文件，存储这个可执行文件的扩展名为 .EXE。

4. 运行

一个 C++的源程序经过编译和连接后生成了可执行文件。运行可执行文件的方法很多，可在操作系统环境下单独运行，也可在 C++语言环境下运行。

1.6 应 用 举 例

1. 编辑 C++源程序

首先，系统要安装 VC++6.0（中文版），然后启动 VC++6.0，启动成功后如图 1-5。

编辑 C++源程序时，选择“文件”菜单项，出现一个下拉式菜单，再选择该菜单中的“新建”选项（热键为 Ctrl + N _），这时又出现一个“新建”对话框，见图 1-6，该框中有 4 个标签，选择“文件”标签，在它的下拉式菜单中，选择“C++source File”菜单项，出现编辑屏幕，见图 1-7。在编辑屏幕上，可以键入 C++的源程序。

〖**例 1-5**〗键入如下源程序。

```
# include < iostream. h >
```