21世纪 计算机基础教育系列教材

谭浩强 主编

C++语言程序设计教程

孟宪福 李盘林 编著

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

21世纪计算机基础教育系列教材

谭浩强　主编

C++语言程序设计教程

孟宪福　李盘林　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

C++语言是在C语言的基础上发展起来的面向对象程序设计语言，它不仅可以编写应用软件，而且特别适合于编写系统软件。本书共由10章组成，按照循序渐进的原则，逐步地介绍C++语言的基本概念和语法规则，特别是花费大量的篇幅来详细讲解类和继承，并利用单独的一章来专门介绍类的设计，使读者在学完本书后，能尽快应用C++语言来解决实际问题。本书是作者根据多年的C++语言教学经验编写而成的，在内容编排上尽量体现易学的特点，在文字叙述上力求条理清晰、简洁，便于读者阅读。

本书可以作为大专院校计算机专业或非计算机专业教材及教学参考书，也可作为自学用书。

图书在版编目(CIP)数据

C++语言程序设计教程/李盘林编著. 北京：电子工业出版社，2003.7
21世纪计算机基础教育系列教材
ISBN 7-5053-8696-4

Ⅰ.C… Ⅱ.李… Ⅲ.C语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆CIP数据核字(2003)第034885号

责任编辑：应月燕
印　　刷：北京四季青印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编100036
经　　销：各地新华书店
开　　本：787×1 092　1/16　印张：10.75　字数：280千字
版　　次：2003年7月第1版　2003年7月第1次印刷
印　　数：5 000册　定价：14.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。
联系电话：(010)68279077

《21世纪计算机基础教育系列教材》序

21世纪是信息时代，是科学技术高速发展的时代。计算机技术与网络技术的结合，使人类的生产方式、生活方式和思维方式发生了深刻的变化。在新世纪中，信息技术已成为当代人类文化的一个重要组成部分了。我们要将信息技术知识和应用向一切有文化的人普及。

高等学校承担着为社会培养高层次人才的任务，大学生毕业后应当成为我国各个领域中的计算机应用人才，成为向全社会推广计算机应用的积极分子。在大学里应当把计算机教育放在十分重要的位置。

我国高校的计算机基础教育起步于20世纪80年代初。20年来从无到有地迅速发展，从理工科专业发展到所有专业，从最初只开设一门语言课到按三个层次设置课程，学时也从三四十小时增加到一二百小时。计算机基础教育已经先后上了几个台阶，现在又需要上一个新的台阶。在新世纪，计算机基础教育已发展为对全体大学生的信息技术教育，我们要引导大学生学习和掌握先进的信息技术，学会在各专业领域中应用信息技术，以适应科学技术和经济发展的需要。

计算机不仅是一种工具，也是一种文化，工具是可选的，文化却是必备的。对学生来说，它还是全面素质教育的一个重要部分，通过学习计算机知识能激发学生对先进科学技术的向往，启发学生对新知识的学习热情，培养学生的创新意识，提高学生的自学能力，锻炼学生动手实践的能力。多年来的实践证明，对计算机感兴趣的学生，绝大多数都是兴趣广泛、思想活跃、善于思考、自学能力较强、喜欢动手实践的。

我们必须认真分析计算机基础教育的特点，根据教学上的需要与可能，制订出恰当的教学要求，使学生在有限的时间内能学到最多的有用的知识。计算机基础教育实际上是计算机应用的教育，应当以应用为目的，以应用为出发点。全国高等院校计算机基础教育研究会曾提出了在计算机基础教育中应当正确处理的10个关系，即：(1)理论与应用的关系，(2)深度与广度的关系，(3)当前与发展的关系，(4)硬件与软件的关系，(5)追踪先进水平与教学相对稳定的关系，(6)课内与课外的关系，(7)课程设置与统一考试的关系，(8)计算机课程与其他课程的关系，(9)要求学生动手能力强与当前设备不足的矛盾，(10)计算机技术发展迅速与师资现状的矛盾。在教学实践中，许多学校都创造了丰富的经验。

在计算机基础教育的教学中，首先需要解决的问题是：准确定位，合理取舍教学内容。我们必须分清楚：哪些内容是需要的，哪些内容是不需要的；哪些内容是目前暂时可以不学而留待以后学的，哪些内容是目前不必学而以后也不必学的；哪些内容是主要的，哪些是次要的。绝不可不加分析、不问主次，贪多求全，使学生感到难以入门。

在教学方法和教材的编写上，要善于用通俗易懂的方法和语言说明复杂难懂的概念。传统的教学三部曲是：提出概念—解释概念—举例说明。我在多年教学实践中对于计算机应用课程总结了新的三部曲：提出问题—介绍解决问题的方法—归纳出必要的概念和结论。从具体到抽象，从实际到理论，从个别到一般。这是符合人们的认识规律的。实践证明，这样做已取得了很好的效果。

为了推动高校的计算机基础教育，我在1996年主编了《计算机教育丛书》，由电子工业出版社出版。编写这套丛书的指导思想是20个字："内容新颖、实用性强、概念清晰、通俗易懂、层次配套"（也可简单地概括为："新颖、实用、清晰、通俗、配套"）。先后出版的近20种供大学计算机基础教育使用的教材，受到高校广大师生的欢迎，几年内已发行近100万册，大家认为它定位准确、程度适当、内容丰富、通俗易懂，便于自学。

在进入21世纪之际，我们根据新时期的要求，按照上面所述的指导思想，重新进行规划，对原有的教材进行了筛选，淘汰了部分内容已过时的教材，同时根据计算机技术和高校计算机基础教育的发展组织编写了一些新教材，并对原有教材进行了修订和补充，以实现推陈出新、不断提高。

我们遴选了具有丰富教学经验的高校老师编写这套教材。在这套系列教材中，我们提供了多种计算机课程的教材供各校选用，其中包括必修课和选修课。不同专业、不同层次的学校都可以从中选到合用的教材，我们还将根据计算机基础教育的需要不断推出新的教材。

本系列教材是由浩强创作室策划、组织和编写的。参加工作的有：谭浩强、薛淑斌、秦建中、史济民、吴功宜、边奠英、徐士良、李盘林、孟宪福、张基温、宋国新、徐安东、毛汉书、李凤霞、许向荣、周晓玉、张玲、刘星、王兴玲、蔡翠平、訾秀玲、索梅、李宁、赵丹娅、仇芒仙等。电子工业出版社对本丛书的出版给予了大力的支持，使得本丛书得以顺利出版。

由于我们的水平和经验有限，加以计算机科学技术发展很快，本丛书肯定会有不少缺点和不足，诚恳地希望专家和读者不吝指正，我们将继续努力工作，使本丛书能尽量满足广大读者的要求。

全国高等院校计算机基础教育研究会会长
《21世纪计算机基础教育系列教材》主编
谭浩强
2003年3月1日

前　言

随着计算机技术的不断发展以及软件程序的高度复杂化，面向对象程序设计的重要性也越来越突显出来，而C++语言则是面向对象程序设计的最重要的代表性语言之一。

C++语言是在被广泛应用的C语言的基础上发展起来的。C++语言在C语言已有的功能的基础上，强化了C语言的基本功能，特别是增加了对类的处理能力，即：

C++语言＝C语言＋基本功能的扩充＋类功能

从这一公式不难看出，C++语言几乎完全继承了C语言的所有功能。从表面上看来，由于C++语言继承了C语言的所有功能，因此，只要学会了C语言，就应该很容易学会C++语言，其实不然。就类本身来讲，就包含了很多复杂的概念，而对于这些概念的正确理解则是学好面向对象程序设计语言的关键。同时，C语言是面向函数的程序设计语言，而C++语言则是面向对象的程序设计语言，这样，在程序设计过程方面就有很大差别。

本书简洁而系统地介绍了C++语言的语法现象和程序设计特点，考虑到大部分读者都学过C语言，同时也考虑到C++语言作为一门独立的课程应具有其系统性，因此，对于C语言中已有的内容，只进行简单的介绍，而把主要篇幅用于对C++语言特有的语法现象的说明上。特别是，为了使读者能够尽快利用C++语言来解决实际问题，在本书的每一章中都给出了大量的例子，这些例子对于理解C++语言的语法现象、完整掌握C++语言的特点是非常有益的。同时，考虑到面向对象程序设计语言的特点，在本书的第8章中专门利用一章的篇幅来详细介绍类的设计，所给出的几个例子都是具有代表性的并具有实用价值的，通过对这些实例的学习，能够使读者进一步掌握面向对象程序设计的要点，并能达到举一反三的目的。

本书共由10章组成，按照循序渐进的原则，逐步地介绍C++语言的基本概念和语法规则，特别是花费大量的篇幅来详细讲解面向对象程序设计的两个基本概念——类和继承。书中的所有例题都在Visual C++环境下测试完成。每章的最后都附有一定量的习题，这些习题对于读者巩固已学的内容是大有益处的。

我们认为，要学好C++语言，除了掌握C++语言的基本理论之外，还必须加强实践环节，读者可以边学习边上机。刚开始时可以调试本书中的例题，待学习一段时间之后，就可以调试自己编写的程序了。只有这样，才能加快学习进度，提高学习效率。

本书在编写过程中，一直得到谭浩强教授的支持和帮助，在此表示深深的谢意。

限于作者水平，书中难免还有一定不足之处，敬请有关老师、计算机工作者和广大读者批评指正。

编　者

2003年3月于大连

目　录

第1章 绪 言

随着计算机技术的不断发展以及软件设计规模的不断扩大，计算机软件的开发面临着两大难题：一是如何越过程序设计的复杂性障碍问题；另一个是如何利用软件来自然地表示客观世界，也就是对象模型问题。面向对象的程序设计技术很好地解决了上述问题，而C++语言正是面向对象程序设计技术的具体实现。

C++语言是在C语言的基础上发展起来的，它既融合了面向对象程序设计技术，又保留了C语言的特征。C++语言在提供了面向对象的设计能力的同时，又保持了与C语言的高度兼容性，使C程序设计人员能够比较容易地转向C++语言。

1.1 面向对象程序设计的特点

世界上的任何事物都可以被看做为对象，对象是对现实世界的抽象。一本书可以是一个对象，而一个图书馆也可以是一个对象。从软件设计的角度来讲，一个对象就是一个高度抽象的模块，该模块中既包含了相应的数据结构，又提供了对数据结构进行操作的方法。正是由于这种高度抽象的结果，使得面向对象程序设计具有许多面向过程程序设计所无法比拟的特点。

归纳起来，面向对象程序设计有如下一些主要特点。

1. 模块化

采用面向对象技术设计出来的程序都是由一个一个的对象组成的，在每一个对象中，既定义了相应的数据结构，同时又定义了操作这些数据结构的方法，这样，每一个对象都是一个完整的功能模块。因此，由面向对象程序设计语言所设计出来的程序，其模块化程度高，易于扩充和维护。

2. 数据隐藏

在面向过程的程序设计中，一般注重的是程序的代码，而把数据放在次要位置上。实际上，只有将数据和操作这些数据的方法结合起来，才能完整地描述一个程序。数据隐藏的目的是将对象的设计和对象的使用分开，使用者不必了解对象实现的细节，只要利用设计者所提供的接口来访问该对象即可。同时，由于为对象中的成员加上了访问权限控制信息，使得外部对对象中成员的访问是有限制的：对于那些私有成员来讲，外部就无法进行访问，从而起到了数据隐藏的作用。

面向对象程序设计通过定义类把数据和方法集成在一个对象中，外部程序只能通过类所规定的接口和类进行通信。

3. 继承

在面向对象程序设计中，对象是程序的基本单位，复杂的对象可由简单的对象来组成，而继承操作正是用于完成这一工作的。它允许从已经存在的类中继承相应的成员（数据和方

法),只要告诉编译器你的新类是由另一个类继承而来的,它就会把另一个类中除私有成员之外的所有成员都赋给你的新类,就如同你重新定义的一样。显然,继承操作能够减轻程序设计的工作量,提高程序的可靠性。

4. 多态性

所谓多态,是指一个名字可具有多种不同的语义,或者说多个函数具有相同的名字但具有不同的作用。

在继承操作中,如果父类和子类中的某个函数具有相同的名字,且被定义为虚函数,则利用指向父类的指针,根据所赋给的对象不同(父类对象或子类对象),就可以调用不同的函数。

面向对象程序设计中的多态性,给程序设计带来了很大的灵活性,使我们既能够重用已有的类,又能够满足新类的需要。

5. 重载

在面向过程的程序设计语言中,每个函数必须有一个惟一的名字,也就是说函数不能重名。而在像C++这样的面向对象程序设计语言中,多个函数可以共用一个名字,只要它们的参数个数不同或参数类型不同即可,这就是函数的重载。在C++语言中,不但函数可以重载,运算符也可以重载,可以根据需要为已有的运算符赋予不同的意义。显然,增加了重载功能以后,将提高程序的可理解性,使程序功能实现起来更自然、更流畅。

1.2 C++语言程序的开发过程

开发一个C++语言程序的基本过程如图1.1所示。

1. 编辑

选择适当的编辑程序,将C++语言源程序通过键盘输入到计算机中,并以文件的形式存入到磁盘中。经过编辑后得到的源程序文件是以.CPP为其文件扩展名的。

2. 编译

通过编辑程序将源程序输入到计算机后,需要经过C++语言编译器将其编译成目标程序。在对源程序的编译过程中,可能会发现程序中的一些词法或语法错误,这时就需要重新利用编辑程序来修改源程序,然后再重新编译。经过编译后得到的目标文件是以.OBJ为其文件扩展名的。

3. 连接

经过编译后生成的目标文件是不能直接执行的,它需要经过连接之后才能生成可执行的代码。连接后所得到的可执行文件是以.EXE为其文件扩展名。

4. 执行

经过编译、连接之后,源程序文件就可以生成可执行的文件,这时就可以执行了。在DOS系统下,只要键入可执行的文件名,并按"回车"键后,就可执行文件了。在Windows系统下,

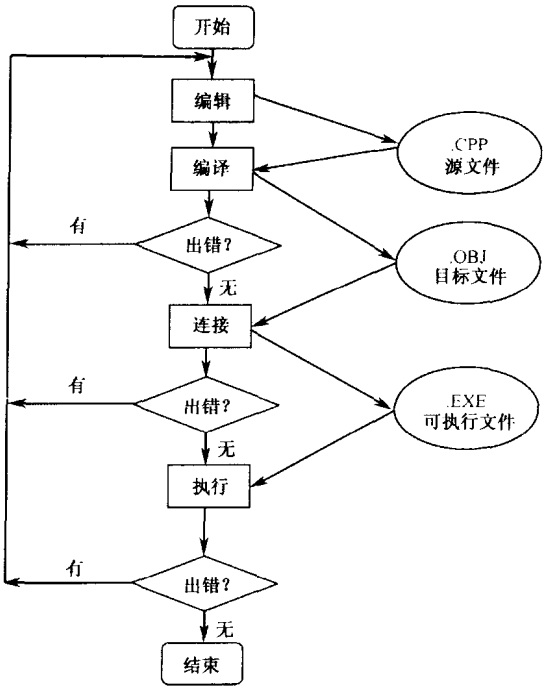

图 1.1　C++语言程序的开发过程

只要双击可执行文件名即可执行。

现在有许多厂家都推出一种集成环境来处理 C++语言程序，如 Turbo C++、Visual C++等，在这种集成环境下，对程序的编辑、编译和连接等操作，都可以在一个窗口下进行，使用起来非常方便。

1.3　C++语言程序的结构

从程序结构的角度来讲，C++语言同 C 语言基本上是相同的。在这一节中，我们通过编写几个简单的 C++语言程序，来阐述 C++语言的程序结构，同时，也对 C++语言的基本语法成分进行相应的说明，以便使读者对 C++语言程序有一个概括的了解，为以后的学习打下基础。

【例 1.1】　编写一个 C++程序，用于显示字符串“Hello, World !”。

```
#include <iostream.h>
void main( )
{
    cout << "Hello, World! \n";
}
```

这是一个简单而完整的 C++语言程序，经过编辑、编译和连接后，其执行结果是在屏幕的当前光标位置上显示如下字符串：

Hello, World!

说明：

(1)一个 C++程序可以由多个函数组成，但任何一个完整的 C++程序，都必须包含一

个且只能包含一个名为 main( )的函数，程序总是从 main( )函数开始执行的。

(2)由左、右花括号括起来的部分是函数体，函数体中的语句将实现程序的预定功能。在本例中，main( )函数的函数体中只有一个语句，其功能是将字符串"Hello, World !"显示到屏幕上。

(3)cout 是标准输出流，它意味着进行标准输出，运算符＜＜表示将数据送入 cout 的意思，为了使用 cout 进行标准输出，必须在程序的开始处包含进 iostream.h 头文件。

【例 1.2】 从键盘输入两个数，并将这两个数的差显示出来。

```
#include <iostream.h>
int SUBab(int a, int b)      //子函数
{
    int c;
    c=a-b;                   //求 a 和 b 的差
    return c;
}
void main( )                 /*主函数*/
{
    int x, y;
    cin >> x;                /*输入 x*/
    cin >> y;                /*输入 y*/
    int z;
    z=SUBab(x, y);
    cout << z;
}
```

说明：

(1)在 C++语言程序中，既可以使用 C 语言中使用的/*和*/来进行注释，也可以使用//符号来进行注释。/*和*/表示由其括起来的部分是注释部分，而//表示从//符号开始到此行末的所有内容都是注释。

从下面的两行语句中，可以看出由/*和*/以及//所表示的注释之间的差异：

```
a=1; /* a=a+10; */     a=a+100;      //结果为 a=101
a=1; // a=a+10;       a=a+100;      //结果为 a=1
```

(2)C++语言程序中的所有变量都必须定义为某种数据类型，同时必须遵循"先定义、后使用"的原则，但并不要求所有变量一定在函数体的开始处(即可执行语句的前面)进行定义，只要在变量的使用之前进行定义即可。例如，此程序中的变量 z 的定义位置就是位于此函数体的可执行语句的中间部分。

(3)一个 C++程序可以由多个函数组成，通过函数之间的调用来实现相应的功能。程序中所使用的函数，既可以是系统提供的库函数，也可以是用户根据需要自己定义的函数。例如，此程序中的 SUBab( )就是用户自己定义的用于求两个数的差的函数。

(4)cin 是标准输入流，它意味着进行标准输入，运算符＞＞表示从 cin 读数据的意思，同 cout 一样，为了使用 cin 进行标准输入，必须在程序的开始处包含进 iostream.h 头文件。

从表面上来看，C++语言只是在 C 语言的基础上增加了对类的处理功能，而实际上，除了类之外，C++语言还在很多细小的方面进行了扩充和完善，同时，即便是类本身也包含了很多复杂的内容。因此，要想学好 C++语言，最根本的方法就是多练习、多上机。由于C++

语言是在C语言的基础发展起来的，因此，几乎C语言中的所有功能在C++语言中都可以直接使用，这为C++语言的初学者提供了方便。

习题

1.1 根据所学的知识，请总结一下什么是面向对象程序设计以及面向对象程序设计的特点。

1.2 编一程序，利用标准输出流对象cout，将"C++ Programming Language"字符串显示到屏幕上。

1.3 编一程序，利用标准输入流对象cin和标准输出流对象cout，从键盘输入一字符串，并将其显示到屏幕上。

1.4 试分析下列程序，并给出其运行结果。

```
#include <iostream.h>
int Mulab(int a, int b)
{
    int Mvaluc;
    Mvalue=a * b;
    return Mvalue;
}
void main( )
{
    int x, y;
    cin >> x;
    cin >> y;
    int result;
    result=Mulab(x, y);
    cout << result;
}
```

1.5 请指出下列程序的错误。

```
#include <stdio.h>
void main( )
{
    int x, y, z;        //变量定义
    cin >> x
    cin >> y
    z=x+y;
    printf("Result=%d", z);
    }
```

第2章　数据类型和运算符

C++语言程序中所用到的每一个常量、变量以及函数等都是程序的基本操作对象，它们都隐式地或显式地与一种数据类型相联系，每种数据类型都表示了它的可能取值范围以及能在其上所进行的运算。C++语言中提供了丰富的数据类型和运算符，运用这些数据类型和运算符能够进行复杂的数学运算。

本章主要讨论C++语言中的一些基本概念，如基本数据类型、指针和引用、运算符以及利用这些运算符来构成相应表达式的一些规则等。

2.1　基本概念

2.1.1　标识符

在计算机语言中，标识符的概念经常被用到。所谓标识符，是指用来标识程序中所用到的变量名、函数名、类型名、数组名、文件名以及符号常量名等的有效字符序列。

在C++语言中，标识符的命名规则是：由字母（大、小写皆可）、数字及下划线组成，且第一个字符必须是字母或下划线。

由上述标识符的命名规则可知，下面的标识符名是合法的：

year，Day，ATOK，x1，_CWS，_change_to

而下面的标识符名是不合法的：

#123，.COM，$100，2002Y，1_2_3

在C++语言中，大写字母和小写字母是有区别的，即作为不同的字母来看待。如标识符RAN、Ran和ran分别表示3个不同的标识符，这一点同有的程序设计语言是有区别的，应引起注意。

2.1.2　常量

常量又称常数，是指在程序运行过程中其值不能被改变的量，如：100，3.14等。常量也分为不同的类型，这是由常量本身隐含决定的（将在下面详细介绍）。为了增加程序的可读性，可以用一个名字（字符序列）来代表一个常量，此时的常量被称为符号常量。有关符号常量的使用，将在“编译预处理”一章中详细介绍。

2.1.3　变量

变量是指在程序运行过程中其值可以被改变的量。变量被区分为不同的类型，不同类型的变量在内存中占用不同的存储单元，以便用来存放相应变量的值。

组成变量名（标识符）的有效字符数随C++语言的编译系统而定。有的编译系统允许使用长达31个字符的变量名，而有的编译系统只取变量名的前8个字符作为有效字符，后面的字符无效，不被识别，这样，只要变量名的前8个字符相同，就被认为是同一个变量。因此，在

进行程序设计之前，应首先了解所使用的编译系统中对变量名长度的规定，以免造成变量使用上的混乱。

2.1.4 关键字

关键字，又被称为保留字或保留关键字，也是C++语言中的一种标识符，它用来命名C++语言程序中的语句、数据类型和变量属性等。每个关键字都有固定的含义，不能另作其他用途，C++语言中的所有关键字都是用小写字母来表示的。

2.2 基本数据类型

在C++语言中，共有5种基本数据类型，它们分别由如下关键字来进行定义：

- bool　布尔型
- int　整型
- char　字符型
- float　单精度浮点型
- double　双精度浮点型
- void　void型

C++语言规定，对程序中用到的所有变量，都必须先定义后使用，每个变量只能与一种数据类型相联系。在定义变量时，不能把C++语言中具有固定含义的关键字（如int、char等）作为变量名，同时，同一个函数内所定义的变量不能同名。

2.2.1 整型变量及其常量

整型变量可用来存放整型数据（即不带小数点的数），其定义方式如下：

```
int  i1, i2;
```

其中，i1和i2即被定义为整型变量。

在C++语言中，整型常量可以用3种数制来表示。

(1) 十进制整型常量：例如319，－200等，其每个数字位可以是0～9。

(2) 十六进制整型常量：如果整型常量是以0x或0X开头，那么，这就是用十六进制形式表示的整型常量。例如，十进制数的129，用十六进制形式表示为0x81或0X81，其每位数字可以是0～9，A～F。

(3) 八进制整型常量：如果整型常量的最高位为0，那么它就是以八进制形式表示的整型常量。例如，十进制数的128，用八进制表示为0200。需注意的是，八进制表示中的每个数字位必须是0～7。

2.2.2 浮点型变量及其常量

在C++语言中，把带有小数点的数称为浮点数。浮点型变量又被称为实型变量，按其能够表示的数的精度，又被分为单精度浮点型变量和双精度浮点型变量。

单精度浮点型变量的定义方式如下：

```
float  f1, f2;
```

其中，f1和f2即被定义为单精度浮点型变量。