普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机规划教材

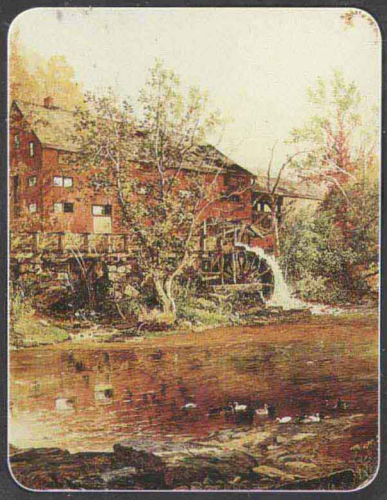
21st Century University Planned Textbook of Computer Science

C++语言程序设计教程（第2版）

The C++ Programming Language (2nd Edition)

吕凤翥 王树彬 编著

- 强调概念理解，引导读者思考
- 传授编程思想，注重编程实践
- 增加了模板、数据结构、异常处理和命名空间等内容



名家系列

人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家

21世纪高等学校计算机规划教材

21st Century University Planned Textbook of Computer Science

C++语言程序设计教程（第2版）

The C++ Programming Language (2nd Edition)

吕凤翥 王树彬 编著

名家系列

人民邮电出版社

北京

图书在版编目（CIP）数据

C++语言程序设计教程 / 吕凤翥，王树彬编著. -- 2版. -- 北京 : 人民邮电出版社，2013.5
21世纪高等学校计算机规划教材
ISBN 978-7-115-31891-6

Ⅰ. ①C… Ⅱ. ①吕… ②王… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第103490号

内 容 提 要

本书系统介绍C++语言的基础知识、基本语法和编程方法。重点讲述C++语言面向对象的重要特征，包括类和对象、继承性和派生类、多态性和虚函数、模板和C++语言实现的常用数据结构、异常处理和命名空间等重要内容。同时，还介绍C++语言对C语言的继承和改进。

本书内容系统全面，偏重应用；通过例题详细讲述C++语言具有的封装性、继承性和多态性，并阐述使用C++语言编程的方法、技巧和工具等。为了方便教学，本书每章最后都备有大量的练习题和上机题。

本书适合作为高等院校“C++语言程序设计”课程的教学用书，还可作为C++语言的自学或教学参考书。

◆ 编　　著　吕凤翥　王树彬
责任编辑　刘　博
责任印制　彭志环　焦志炜

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：21.25　　2013年5月第2版
字数：561千字　　2013年5月北京第1次印刷

定价：45.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

第 2 版前言

C++语言主要应用于面向对象的程序设计，是目前广泛使用的一种程序设计语言。为了满足广大读者渴望学习和掌握 C++语言编程的愿望，为从事计算机编程工作和深入学习计算机专业知识打下基础，作者在总结长期从事“C++语言程序设计”课程教学经验的基础上编写了此书。

本书具有下述特点。

首先，教学重点明确，语言简明，针对难点和疑点，加强了解释和分析。注重培养分析问题解决问题的能力。

其次，内容安排上强调边看书边思考，注重培养学习思考能力。本书的许多例题中都提出了思考问题，每章后边都从多方面给出练习题，引导读者去思考，去理解，去掌握学过的内容。

最后，注重实践。书中强调上机实践，每章最后都附有上机指导，用来引导读者上机实践。

本书共分 13 章。其中第 1 章至第 5 章主要复习 C 语言中讲过的一些重要内容，同时指出 C++语言对 C 语言的某些改进。这些改进对学习 C++语言是很重要的。第 6 章至第 10 章主要讲述 C++语言面向对象的特征，这是 C++语言中的重点，这部分内容是 C 语言中所没有的。学好这部分内容对学习其他面向对象语言也会有帮助。第 11 章至第 13 章主要讲述利用 C++语言实现的常用数据结构和编写大规模 C++软件的主要工具。这些内容对于提高 C++语言的编程能力很有帮助。

本书第 1 版出版后，作者又经过了一段时间的教学实践和听取了部分读者反映的意见，在保持对原书章节不做较大改变的前提下，进行了如下修订工作。

1．模板是 C++语言的一个重要特性，模板机制通过数据类型的参数化，能够实现更为通用的程序模块，提高软件的开发效率。这次修订增加了模板的内容，作为第 11 章，主要介绍模板的基本概念、函数模板和类模板的定义及其使用方法。

2．程序设计主要包括算法和数据结构两个方面，任何语言的程序设计都会对数据进行处理。掌握基本数据结构的知识对学好 C++语言是非常必要的。这次修订增加了数据结构的内容，作为第 12 章，主要内容是采用面向对象的程序设计方法，利用 C++语言实现几种常用的数据结构，并介绍在给定数据结构中的查找和排序运算。

3．异常处理机制提供结构化的出错处理模式，能够提高程序的可读性、可维护性和容错能力。命名空间能够避免全局标识符的命名冲突。这些内容对程序设计和调试都很有帮助。这次修订增加了这两部分内容，作为第 13 章，主要内容是介绍异常处理机制的概念和使用方法、命名空间的定义及其成员的访问方法。

4．增加了附录 C，方便读者对 C++标准库有一个简单的了解。

5．改正了原书中出现的个别错误，在概念描述方面更加精炼和准确。

本书所有的程序，包括例题、练习题以及习题答案的程序都已在 Visual C++ 6.0 版本的编译系统下通过，有关资料与电子教案放在人民邮电出版社教学服务与资源网（www.ptpedu.com.cn）资源下载区中。

本书第 2 版由北京大学吕凤翥、内蒙古大学王树彬编写，其中王树彬编写了第 11 章、第 12 章、第 13 章和附录 C，并在程序的录入、调试以及校对方面做了大量的工作；吕凤翥编写其余章节并统稿。

本书中若有错漏或欠妥之处，敬请读者批评指正。

编　者

2013 年 2 月

目　录

第 1 章 C++语言概述

本章介绍面向对象的概念和C++语言的特点。通过C++程序实例，分析C++程序在结构上的特点及书写程序应注意的事项。本章还介绍使用Microsoft Visual C++ 6.0编译系统实现C++程序的方法。

1.1 面向对象的概念

C++语言是一种面向对象的程序设计语言，在介绍C++语言之前，先介绍一下有关面向对象的概念，有助于对C++语言的理解和掌握；通过对C++语言的学习又会进一步加深对面向对象方法的认识。

1.1.1 面向对象方法的由来

面向对象方法是人们开发软件的一种方法，这种方法的提出是软件研究人员对软件开发在认识上的一次飞跃，它是软件开发史上的一个里程碑。它标志着软件开发产业进入一个新阶段。

在面向对象方法出现之前，人们采用的是面向过程的方法。面向过程方法是一种传统的求解问题的方法。该方法将整个待解决问题按其功能划分为若干个相对独立的小问题，每个小问题还可以按其功能划分为若干个相对独立的更小的问题，依此类推，直到将所划分的小问题可以容易用程序模块实现为止。在面向过程的程序设计中，每个程序模块具有相对独立的功能，由小模块组成大模块，最后组成一个完整的程序。整个程序的功能是通过模块之间相互调用来实现的。这种面向过程的方法具有很多的弊病。第一，该方法将数据和数据处理过程分离成为相互独立的实体，当数据结构一旦发生变化时，所有相关的处理过程都要进行相应的修改，因此，程序代码的可重用性较差。第二，该方法对于图形界面的应用开发起来比较困难，而图形界面越来越被人们广泛使用。第三，面向过程的程序设计中，模块之间有较大的依赖性，这对调试程序和修改程序带来一定的难度。

面向对象方法是求解问题的一种新方法，它把求解问题中客观存在的事物看做各自不同的对象，这符合人们习惯的思维方式，再把具有相同特性的一些对象归属为一个类，每个类是对该类对象的抽象描述。对象之间可以进行通信。类之间可以有继承关系，函数和运算符可以重载，这样可以提高程序的可重用性，便于软件开发和维护。

总之，面向对象方法是计算机科学发展的要求。随着人们对信息的需求量越来越大，软件开发的规模也越来越大，对软件可靠性和代码的重用性的要求越来越高。这时，面向过程的方法使

得分析结果不能直接映射待解决的问题，并且分析和设计的不一致给在编程、调试、维护等诸方面造成不便和困难。在这种情况下，面向对象方法应运而生。由于面向对象方法具有封装、继承和多态等特性，与面向过程方法相比，它较好地克服了在编程、调试和维护等方面的不便和困难，提供了代码的重用率，使得软件开发变得更为容易和方便。

1.1.2 面向对象的基本概念

面向对象是一种由对象、类、封装、继承和多态性等概念来构造系统的软件开发方法。这些新的概念描述了面向对象这种新方法。

1. 对象

对象是现实世界中客观存在的某种事物，它可以是有形的，也可以是无形的。对象是一种相对独立的实体，它具有静态特性和动态特性，通常通过一组数据来描述对象的静态特性，使用一组行为或功能来表示对象的动态特性。

对象是系统中用来描述客观事物的一个实体，它是软件系统的基本构成单位。对象是由一组属性和一组行为构成的。属性是描述对象的静态特性的数据项，行为是描述对象动态特性的操作。

2. 类

类是人们对于客观事物的高度抽象。抽象是忽略事物的非本质特性，只抓住与当前相关的特性，从而找出其共性，把具有共同特性的事物划分为一类，得到一个抽象的概念。例如，在生活中经常遇到的抽象出来的概念有桌子、房屋、汽车和足球等。

面向对象方法中的类是一种类型，它是具有相同属性和行为的对象的集合。类是具有相同属性和行为的若干对象的模板。类为属于该类的全部对象提供了抽象的描述，这种描述包括了属性和行为两大部分。类与对象的关系就像模具和铸件的关系。某个类的对象又称为该类的一个实例。

3. 封装

封装是指把对象的属性和行为结合成一个独立的单位，又称为封装体。对象的属性通常用一组数据项来表示，对象的行为又称为服务，通常用方法或函数来表示。封装体具有独立性和隐藏性。独立性表现在封装体内所包含的属性和行为形成了一个不可分割的独立单位；隐藏性表现在封装体内的有些成员在封装体外是不可见的，这部分成员被隐藏了，具有一定的安全性。一个封装体与外部联系只能通过有限的接口。

4. 继承

继承是面向对象方法提高重用性的重要措施，继承表现了特殊类与一般类之间的关系。当特殊类包含了一般类的所有属性和行为，并且特殊类还可以有自己的属性和行为时，称作特殊类继承了一般类。特殊类又称为派生类，一般类称为基类。

继承的重要性就在于它大大地简化了对于客观事物的描述。例如，已经描述汽车这个类属性和行为，由于小轿车是汽车类的特殊类，它具有汽车类的所有属性和行为，在描述小轿车类时，只需描述小轿车本身的属性和行为，而汽车类的属性和行为不必再重复了，因为小轿车类继承了汽车类。

5. 多态性

多态性指的是一种行为对应着多种不同的实现。在同一个类中，同一种行为可对应着不同的实现，例如，函数重载和运算符重载都属于多态性。同一种行为在一般类和它的各个特殊类中可以具有不同的实现，例如，动态联编是属于这类多态性。在一般类中说明了一种求几何图形面积的行为，这种行为不具有具体含义，因为并没有确定具体的几何图形，又定义一些特殊类，如“三

角形”、“圆形”、“正方形”、“矩形”、“梯形”等，它们都继承了一般类。在不同的特殊类中都继承了一般类的“求面积”行为，可以根据具体的不同几何图形使用求面积的公式，重新定义“求面积”行为的不同实现，使之分别实现求“三角形”、“圆形”、“正方形”、“矩形”和“梯形”等面积的功能。这就是面向对象方法的重要的多态性。

1.2　C++语言的特点

C++语言是 20 世纪 80 年代初期由美国贝尔实验室的科研人员提出的，它是一种继承了 C 语言的面向对象的程序设计语言。

1.2.1　C++语言是面向对象的程序设计语言

C++语言支持面向对象的程序设计，主要表现在它支持面向对象方法中的 3 个主要特性。

1. 支持封装性

C++语言允许使用类和对象。类是支持数据封装的工具，对象是数据封装的实现。类中成员有不同的访问权限。类中的私有成员仅由该类体内的成员函数访问，因此，私有成员具有隐藏性，类体外是不可见的。类中的公有成员是类体与外界的一个接口，类体外面的函数可以访问类体中的公有成员。类中还有一种保护成员，它具有公有成员和私有成员的双重特性，它具体使用在类的继承中。

2. 支持继承性

C++语言支持面向对象方法中的继承性，它不仅支持单重继承，而且支持多重继承。继承性给 C++语言编程带来了方便，增强了程序的扩展性和可重用性，提高了软件开发的效率。继承是两个类之间的关系，基类和派生类是继承中的重要概念。派生类继承了基类中的所有成员，并且还可以定义自身的新成员，继承实现了抽象和共享的机制。继承和封装是衡量一种语言是否是面向对象的程序设计语言的两个重要标准。C++语言支持封装，又支持继承，因此，可以断定 C++语言是面向对象的程序设计语言。

3. 支持多态性

多态性是在继承性基础上的面向对象方法中的重要特性之一。C++语言支持多态性主要表现如下两个方面。

① 支持函数重载和运算符重载。重载是指一个函数名可以有多种实现，即同一个行为对应不同实现，这便是多态性。

② 支持动态联编。动态联编反映了基类和派生类中同名函数的多态性。动态联编是在公有继承的前提下，通过虚函数来实现的。动态联编虽然没有静态联编运行效率高，但它可以通过高度抽象，提高程序的灵活性和扩充性。

1.2.2　C++语言继承了 C 语言

C++语言与 C 语言兼容，C 语言是 C++语言的一个子集。C 语言的词法、语法和其他规则都可以用到 C++语言中。例如，C 语言中的类型、运算符和表达式在 C++语言中都可以使用；C 语言中的语句也是 C++语言的语句；C 语言中的函数定义和调用也适用于 C++语言，只是 C++语言对此稍有改进和扩充；C 语言中的预处理命令也可用于 C++语言，只是宏定义命令在 C++语言中

较少使用；C语言中的构造类型，如数组、结构和联合在C++语言中也是适用的，只不过C++语言中较多是使用类类型；C语言中的指针在C++语言中也可使用，但是C++语言中引入了引用概念，在某些情况下减少了指针的使用；还有C语言中作用域的规则，存储类的规定等在C++语言中也都适用。

由于C++语言继承了C语言，使得已经掌握了C语言的人们学习C++语言比较容易，这也是C++语言得以广泛使用的原因之一。已经学会了C语言的人们学习C++语言时，要做到一种思维方式的转变，即从面向过程的思维方式转变到面向对象的思维方式，没有这个转变是学不好C++语言的。

由于C++语言继承了C语言，因此，C++语言仍旧具有C语言的简练明了的风格，同时还不得不保留某些C语言的面向过程的特性。所以，有人说C++语言是一种不完全的面向对象的程序设计语言。在C++程序中，允许类体外的函数存在，这便是保留面向过程的特征。

1.2.3 C++语言对C语言进行了改进

C++语言虽然保留了C语言的风格和特点，但又针对C语言的某些不足做了较大的改进。改进后的C++语言与C语言相比，在数据类型方面更加严格了，使用更加方便了。

下面简单扼要地列举一些C++语言对C语言的改进内容，更详细的介绍参见本书第2章至第5章。

① C++语言中规定所有函数定义时必须指出数据类型，不允许默认数据类型。无返回值的函数使用void进行说明，返回值为整型的函数使用int进行说明。

② C++语言规定函数说明必须使用原型说明，不得用简单说明。

③ C++语言规定凡是从高类型向低类型转换时都需加强制转换。

④ C++语言中符号常量建议使用const关键字来定义，这种方法可以指出常量类型，使用简单宏定义命令定义符号常量没有类型说明。

⑤ C++语言中引进了内联函数，建议使用内联函数取代带参数的宏定义命令，这也是增加了对参数的类型说明。

⑥ C++语言允许设置函数参数的默认值，提高了程序运行的效率。

⑦ C++语言引进了函数重载和运算符重载的规则，为编程带来了方便。

⑧ C++语言引进了引用概念，使用引用作函数的参数和返回值，比使用指针作函数参数和返回值更加方便，并且二者具有相同的特点。这就使得C++程序中减少了对指针的使用，避免由于指针使用不当造成的麻烦。

⑨ C++语言提供了与C语言不同的I/O流类库，方便了输入/输出操作。

⑩ C++语言为方便操作还采取了其他措施。例如，使用运算符new和delete代替函数进行动态存储分配；增添了行注释符（//），为行注释信息提供了方便；取消了C语言中在函数体和分程序中说明语句必须放在执行语句的前边的规定等。

1.3 C++程序在结构上的特点

本节介绍C++语言编写的程序在结构上的特点和在书写上应注意的事项。为此，首先列举两个C++语言的程序，从这两个程序实例中分析C++程序的特点。

1.3.1　C++程序举例

【例 1.1】从键盘上输入两个 int 型数，编程求这两个 int 型数之和。

程序内容如下：

```
#include <iostream.h>
int add(int ,int);
void main()
{
    int a,b;
    cout<<"Enter a b: ";
    cin>>a>>b;
    int c=add(a,b);
    cout<<"a+b="<<c<<endl;
}
int add(int x,int y)
{
    return x+y;
}
```

运行该程序后，显示下述提示信息：

```
Enter a b:
```

这时，在键盘上输入 18␣36↙后，输出显示下述结果：

```
a+b=54
```

其中，“␣”表示空格符，“↙”表示按 Enter 键。

程序分析：

例 1.1 是一个 C++语言的程序，初看上去该程序从结构形式和书写规则上与 C 语言程序很相似。该程序由一个文件组成，该文件有两个函数，一个是主函数 main()，另一个是被调用函数 add()。

再仔细看会发现该程序与 C 语言程序有如下的区别。

① 该程序开头包含了 iostream.h 文件，该文件中包含了 C++语言的输入/输出操作中的相关内容。例如，该程序中使用的插入符（<<）和提取符（>>）都被重载定义在 iostream.h 文件中，还有 endl 也定义在该头文件中，它与换行符（'\n'）功能相同。

② 该程序中第 2 条语句是函数说明语句，这里使用的是原型说明，不仅要说明函数名字和类型，还要说明函数参数的个数和类型。

③ 主函数 main()和被调用函数 add()在定义时都加了类型说明符 void 和 int，这是不可省略的。

④ 在主函数中出现了如下所示的输出和输入语句：

```
cout <<"Enter a b: ";
cout <<"a+b="<<c<<endl;
cin>>a>>b;
```

其中，前边两条是输出语句，后边一条是输入语句。

下面结合该例程序介绍 C++语言的标准文件的输出语句和输入语句。关于输入/输出操作的内容详见本书第 10 章。由于标准文件输入/输出操作在一开始的程序中就不可避免地要出现，因此在这里仅就使用重载运算符进行标准文件的输入/输出介绍如下。

- 使用插入符进行输出操作的格式如下：

〈操作数 1〉 << 〈操作数 2〉 << 〈操作数 3〉…

其中，〈操作数 1〉是输出流对象名，C++语言规定标准输出设备屏幕的对象名为 cout。〈操作数

2>，<操作数 3>…是待输出的表达式。这里，<<是被系统重载的左移运算符，重载后的功能是将右操作数的值输出到左操作数指定对象上。下列输出语句：

```
cout <<"Enter a b:";
```

将字符串常量“Enter a b:”输出到屏幕的当前光标处。

下列输出语句：

```
cout <<"a+b="<<c<<endl;
```

先将字符串常量“a+b=”输出到屏幕的当前光标处，接着再将变量 c 的值输出到屏幕的当前光标处，最后，再输出一个换行符到屏幕的当前光标处。于是，输出结果为：

```
a+b=54
```

- 使用提取符进行输入操作的格式如下：

```
<操作数 1> >> <操作数 2> >> <操作数 3> …
```

其中，<操作数 1>是输入流对象名，C++语言规定标准输入设备键盘的对象名为 cin。<操作数 2>，<操作数 3>…是用来接收从输入流中提取的输入项的变量名。这里，>>是被系统重载的右移运算符，重载后的功能是将从左操作数的输入流对象中提取的输入项数据赋值给右操作数的变量，要求具有相同的类型。下列输入语句：

```
cin >>a>>b;
```

将从键盘上读取的第一个 int 型数赋值给变量 a，再将从键盘上读取的第二个 int 型数赋值给变量 b。于是通过该输入语句使得变量 a 和 b 从键盘上获取了值。

通过以上分析可以看出该程序实现了将从键盘上输入的两个整型数求和后，输出显示在屏幕上的功能。

【例 1.2】一个带有类和对象的 C++程序。关于类和对象将在本书第 6 章介绍，这里只是给出带有类和对象的 C++程序结构。

程序内容如下：

```
#include <iostream.h>
class A
{
  public:
     A(int i)
     {  a=i; }
     int fun1()
     {  return a+a; }
     int fun2()
     {  return a*a;  }
  private:
     int a;
};
void main()
{
   A x(5);
   cout<<x.fun1()<<endl;
   cout<<x.fun2()<<endl;
}
```

运行该程序后，输出结果显示如下：

```
10
25
```

程序分析：

该程序是由一个类的定义和一个主函数构成的。

程序开始，定义了一个类 A，关于类的定义格式及其相关规则在本书第 6 章中介绍，这里，只是简单进行描述。类 A 中定义了 3 个公有的成员函数，其中一个是带有一个参数的构造函数，其名字同类名，另外两个成员函数分别是 fun1()和 fun2()，其功能分别是求其变量 a 的 2 倍值和 a 的平方值。类 A 中还有一个私有的数据成员 a，它是一个 int 型变量。

在主函数 main()中，先定义一个类 A 的对象 x，并对它进行了初始化，使对象 x 的数据成员 a 的值为 5。主函数中另外两条输出语句，使用了插入符向屏幕上显示输出表达式 x.fun1()和 x.fun2()的值，每个占一行。这里，x.fun1()是通过 A 类对象 x 调用类中的成员函数 fun1()，于是其值为 10，同样地，x.fun2()的值为 25。

1.3.2　C++程序结构上的特点

通过例 1.1 和例 1.2 中的 C++程序分析，可以看出 C++程序在结构形式上的特点，基本上与 C 语言程序相似。具体地讲，例 1.1 中的程序由两个函数组成，其中有一个主函数，它同 C 语言程序结构完全一样；例 1.2 中的程序由一个类和一个主函数组成，不同于 C 语言程序的是类成为了组成 C++语言程序的成分。

归纳 C++语言程序的特点如下。

① C++语言程序是由若干个类和函数组成的。这些类和函数可以放在一个文件中，也可以放在多个文件中。

② C++语言程序中的函数有两个种类，一个种类是类体内的成员函数，另一个种类是类体外的一般函数。

③ C++语言程序中有且仅有一个主函数 main()，其他的一般函数都是由主函数或主函数所调用的函数所调用。C++语言程序是从主函数开始执行的。

④ C++语言程序中可具有若干个类，类之间可以是继承关系，也可以是包含关系。一个类可以被嵌套在另一个类中，也可以定义在一个函数体中。

⑤ C++语言程序中的成员函数和一般函数都是由函数头和函数体构成的，函数体由若干条语句组成的；函数头中包括函数名、函数类型和函数参数。

⑥ C++语言程序与 C 语言程序一样，可以使用预处理命令，也可以使用注释信息。

另外，C++语言程序与 C 语言程序一样，可读性比较差，因此要求在书写上要遵照习惯格式和方法，这样可以提高程序的可读性。例如一行写一个语句，大括号采用统一格式，书写时采用缩格方法，适当使用注释信息等，这些在书写 C++语言程序中都适用。

1.4　C++程序的实现

本节介绍如何实现 C++程序。介绍使用 Microsoft Visual C++ 6.0 版本编译系统实现 C++语言的单文件程序和多文件程序的方法。

1.4.1　C++程序的编辑、编译和运行

C++程序和其他高级语言源程序一样，实现 C++程序应有如下 3 个步骤。

1. 编辑

编辑是将编写好的C++语言源程序通过输入设备录入到计算机中，生成磁盘文件加以保存。录入程序可采用两种方法，一种是使用机器中装有的文本编辑器，将源程序通过选定编辑器录入生成磁盘文件，并加扩展名为.cpp；另一种是选用C++编译系统提供的编辑器，编辑C++语言源程序，这是常用的方法。例如，Visual C++ 6.0编译系统提供一个全屏幕编辑器，可使用它来编辑C++语言源程序。

2. 编译

C++语言的源程序编辑完成后，存放在磁盘上，运行前必须先经过编译。编译操作是由系统提供的编译器来实现的。编译器的功能是将源代码转换成为目标代码，再将目标代码进行连接，生成可执行文件。

整个编译过程可分为如下3个子过程。

① 预处理过程。程序编译时，先执行程序中的预处理命令，然后再进行正常的编译过程。

② 编译过程。编译过程主要进行词法分析和语法分析。在分析过程中，发现有不符合词法和语法规则的错误，及时报告用户，将其错误信息显示在屏幕上。在该过程中还要生成一个符号表，用来映射程序中的各种符号及其属性。

③ 连接过程。将编译生成的目标代码中加入某些系统提供的库文件代码，进行必要的地址链接，最后生成能运行的可执行文件。

3. 运行

运行可执行文件的方法很多，最常用的方法是选择编译系统的菜单命令或工具栏中的按钮命令来运行可执行文件。

运行可执行文件也可以在MS-DOS系统下，在DOS提示符后，直接键入可执行文件名，如有参数还可键入参数，再按回车便可运行。

可执行文件被运行后，在屏幕上输出显示其运行结果。

1.4.2 C++程序实现举例

下面分别介绍单文件程序的实现方法和多文件程序的实现方法。

1. 单文件程序的实现方法

下面通过一个具体的C++应用程序讲述单文件程序的实现方法。

（1）编辑C++源程序并存入磁盘

【例1.3】分析下列程序的输出结果。

```
#include <iostream.h>
void main()
{
    int a,b,s;
    a=15;
    b=18;
    s=a+b;
    cout<<"s="<<s<<endl;
}
```

编辑方法如下：

单击主窗口菜单栏中的“File”菜单项，弹出其下拉菜单。在该下拉菜单中选择“New”子菜单项，弹出“New”对话框。在该对话框中选择“Files”标签后，出现如图1.1所示的“New”对