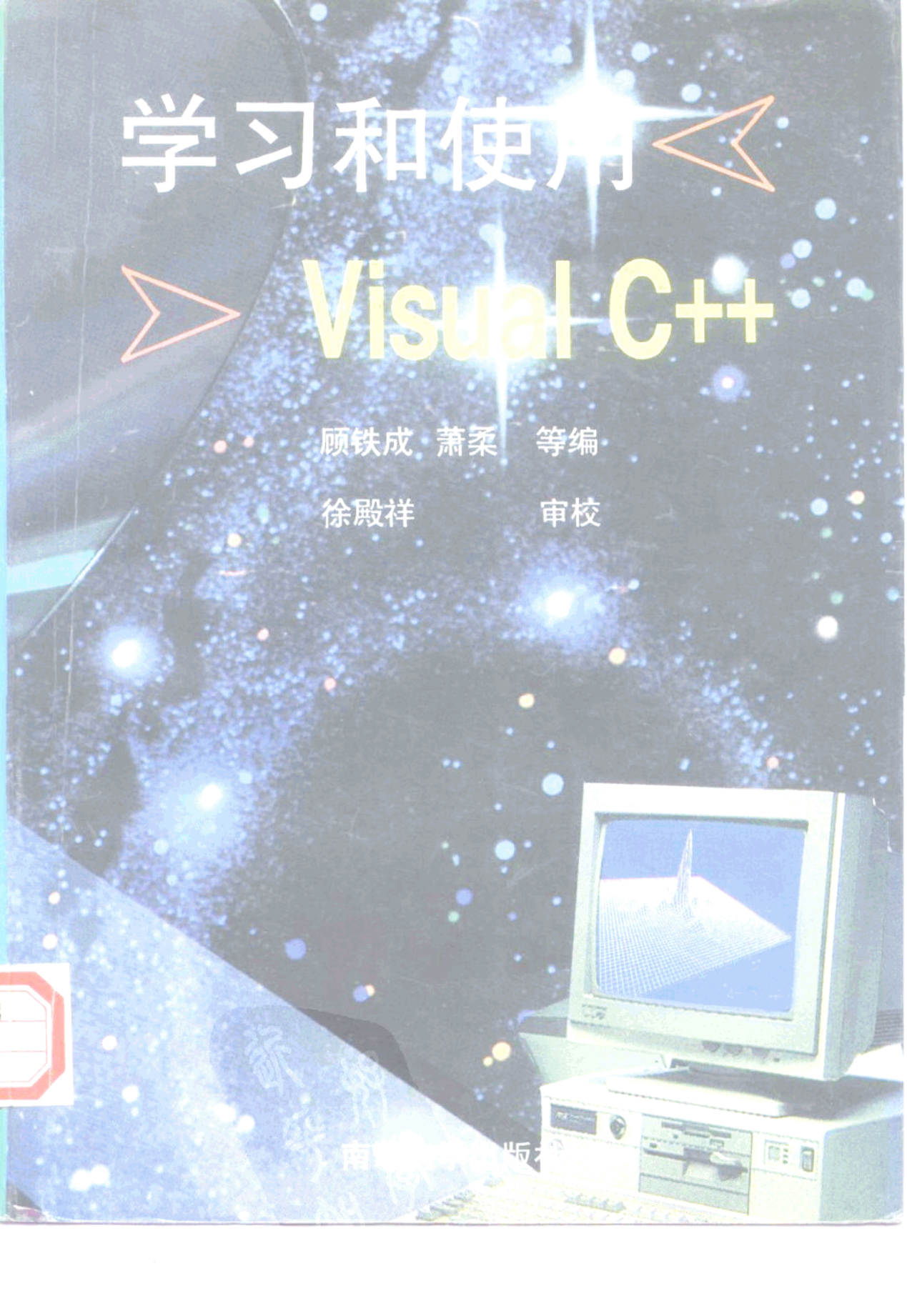
学习和使用 Visual C++

顾铁成 萧柔 等编

徐殿祥 审校



OOP 技术丛书

学习和使用 Visual C++

顾铁成　萧　柔　等编著

徐殿祥　审校

南京大学出版社

1995·南京

(苏)新登字 011 号

内 容 简 介

Microsoft Visual C++是一种编程工具，用于在集成 Windows 环境中构造和调试基于 Windows 的应用程序和库。Visual C++综合了高级C++应用程序框架类和集成的 Windows 宿主开发工具，因而使复杂的 Windows 应用程序开发任务大为简化。

Visual C++中用到的 Windows 宿主开发工具包括 Visual Workbench、App Studio、AppWizard、ClassWizard，以及其他一些可从 Visual Workbench 中(或由它本身)访问的实用程序。

本书比较详细、系统地介绍 Visual C++程序设计的方法与技术。

本书及其姊妹篇《Visual C++开发工具 Visual Workbench 和 AppStudio 指南》是一套学习和使用 C++和 Visual C++极好的教材和教学参考书，可供从事计算机系统软件及应用软件开发的工程技术人员使用。

OOP 技术丛书

学习和使用 Visual C++

顾铁成 萧 柔 等编著

徐殿祥 审校

*

南京大学出版社出版

(南京大学校内 邮政编码 210093)

江苏省新华书店发行 陆军指挥学院印刷厂印刷

*

开本：787×1092 1/16 印张：20.5 字数：499 千

1995 年 9 月第 1 版 1995 年 9 月第 1 次印刷

印数：1—5000

ISBN 7—305—02873—8/TP・136

定价：25.00 元

前　言

Microsoft Visual C++是目前正在流行的编程工具，用于在 MS Windows 环境中构造和调试基于 Windows 的应用程序和库。Visual C++综合了高级 C++应用程序框架类和 MS Windows 宿主开发工具，因而使复杂的 Windows 应用程序开发任务大为简化。

Visual C++中用到的 Windows 宿主开发工具包括集成开发环境 Microsoft Visual Workbench，App Studio，AppWizard，ClassWizard，以及其他一些可以从 Visual Workbench 中（或由它本身）访问的实用程序。

在开发过程中，可以利用 AppWizard 来为项目创建 C++ Microsoft Foundation Class Library 源文件。如果要创建和编辑对话框、菜单、工具条等控件资源，可以使用 AppStudio。ClassWizard 则可以用来为资源或 View 和 Document 类增加 C++类框架代码和消息图。

Visual C++有两种版本：Visual C++ Standard Edition（标准版本）和 Visual C++ Professional Edition（专业版本）。两种版本使用相同的、但略有些区别的集成开发环境。

本书分为上下两篇。上篇包括第一章到第十章，较详细地、系统地介绍 C++语言的基础；下篇包括第十一章到第十九章，介绍了 Visual C++程序设计的方法与技术。

参加本书编写工作的人员有顾铁成、萧柔、潘金贵、高青、薛仲清等十二位同志。并由潘金贵和顾铁成同志作了修改和统编。徐殿祥博士对全书进行了仔细的审阅和校改，值此表示深深的谢意。

由于水平、时间所限，不妥之处在所难免，欢迎广大读者批评指正。

编著者

1995 年 8 月于南京

目 录

第一篇 学 习 C++

第一部分 C++ 介 绍

第三部分 面向对象的设计

第二篇 程序设计技术

第一部分 提高程序的性能

第二部分 特 殊 环 境

第一篇
学　习　C++

第一部分
C++介绍

第一章　C++初步

C++是由C派生而来的。它是C的一个超集(只有几点例外),意即C中有的C++中都有。C++对C中固有的特性作了一些简单的增强,另外还增加了C中所没有的一些新的重要特性。

本章将介绍C和C++约定之间的一些不同之处。首先介绍C++处理输入和输出的新方式,这对设计一个实用的程序来说是必须了解的。接着,本章还将介绍C++的注释和函数原型。

1.1　使用流进行输入和输出

下面是一个非常简单而又典型的C++程序HELLO.CPP:

```
#include <iostream.h>

void main()
{
    cout<<"Hello,World\n";
}
```

这个程序是有名的C程序HELLO.C的C++版本。程序中包含的不是STDIO.H,而是IOSTREAM.H;它没有调用printf,而是采用了一种我们所不熟悉的语法,其中出现了一个未

定义的名为 cout 的变量，按位左移操作符(<<)和一个串。

1.2 标准的输出流

在C++中，执行输入和输出的设施称为“流”(stream)。本书中的示例程序都采用流来读取和显示信息。名 cout 代表标准的输出流，它可以用来显示信息：

```
cout<<"Hello,World\n";
```

串“Hello,World\n”被送至标准输出设备，即屏幕上。<<操作符称为“插入”(*insertion*)操作符。它从要发送的内容(即串)指向它要去向的地方(即屏幕)。

假定我们希望打印的是一个整数，而不是串。在 C 中，可使用一个 printf 语句，该语句有一个描述参数的格式串：

```
printf("%d",amount);
```

在C++中，不需要格式串：

```
#include <iostream.h>

void main()
{
    int amount=123;
    cout<<amount;
}
```

程序输出 123。

无需使用格式串，就可以将任何基本数据类型的数据发送至输出流中。cout 流会识别不同的数据类型，并对它们作出正确的解释。

下面的例子说明如何用一条语句将一个串、一个整数和一个字符常量送至输出流。

```
#include <iostream.h>

void main()
{
    int amount=123;
    cout<<"the value of amount is"<<amount<<'.';
}
```

这个程序向 cout 中发送三种不同类型的数据：一个串常量，一个整型变量 amount，以及一个字符常量‘.’，它给句子加上一个(英文)句号。程序打印出下面的信息：

```
The value of amount is 123.
```

注意多个值是如何用一条语句来显示的：<<操作符重复作用于每个值。

1.3 格式化输出

到目前为止，我们所给出的示例程序都还没有向 cout 中发送格式化的输出。假设我们希望以十六进制(而不是十进制)形式来显示一个整数。printf 函数可以很好地完成这一任务，cout 又是怎么做的呢？

注意：每当想到如何用C++来完成 C 能完成的任务时，记住整个 C 语言都是C++的一部分。如果找不到一种更好的办法，可以回过去用 C。

C++将一组"操纵子"(manipulator)与输出流联系起来。它们可改变整型参数的缺省格式。为了改变格式，要将操纵子插入流中。这些操纵子的名称为 dec，oct 和 hex。

下一个程序例子说明了如何以三种不同的形式来显示一个整数值。

```
#include <iostream.h>

main()
{
    int amount=123;
    cout<<dec<<amount<<' '
        <<oct<<amount<<' '
        <<hex<<amount<<;
}
```

程序插入了各操纵子(dec，oct 和 hex)，用来将 amount 的值转换为不同的表示形式。

这个程序的输出为：

```
123  173  76
```

变量 amount 的十进制值为 123，上面的各个值分别是其不同数制下的表现形式。

1.4 标准的错误流

如果要将输出送到标准错误设备上，就要用 cerr(而不能用 cout)了。利用这一技术，可以从标准输出重定向到另一个文件或设备的程序中，将信息送到屏幕上。

1.5 标准输入流

除了显示信息外，有时还需要从键盘中读取数据。C++标准输入版本的形式为 cin。下面的例子说明了如何利用 cin 从键盘读取一个整数。

```
#include <iostream.h>

void main()
```

```
{
    int amount;
    cout<<"Enter an amount...\n";
    cin>>amount;
    cout<<"The amount you entered was "<<amount;
}
```

这个例子程序提示用户输入一个量值。然后，cin 将输入的值送给变量 amount。下一条语句利用 cout 来显示该量值，从而说明 cin 操作是有效的。

cin 还可以用来读取其他类型的数据。下一个例子说明了如何通过键盘读一个串。

```
#include <iostream.h>

void main()
{
    char name[20];
    cout<<"Enter an amount...\n";
    cin>>name;
    cout<<"The name you entered was "<<name;
}
```

这个程序例子所说明的做法有一个严重的缺陷。字符数组只有 20 个字符长。如果键入了过多的字符，栈就会上溢，并出现奇怪的现象。get 函数可解决这个问题。它在有关 iostream 类库参考的内容中解释。至于目前，例子程序假定读者不会键入过多的字符。

注意：回忆一下，printf 和 scanf 并不是 C 语言本身的一部分，而是运行时刻库中定义的函数。类似地，cin 和 cout 流也不是C++语言的一部分，而是在流库中定义的，这就是为什么在使用它们时要包含 IOSTREAM.H 的缘故。此外，<<和>>操作符的含义与它们所处于的上下文有关。仅当与 cout 和 cin 一起使用时，它们才能显示或读取数据。

1.6 C++ 注 释

C++支持以/*开始、以*/结束的 C 注释形式。此外，它还有另外一种注释形式，很多编程人员都比较爱用这一种形式，即//序列。每当此序列出现时(除非它是在一个串中)，//后到当前行末间的所有内容都是注释。

下面的程序例子在前一个例子中加了一些注释：

```
// C++注释
#include <iostream.h>

void main()
{
```

```
    char name[20];                                    // 声明一个串
    cout<<"Enter an amount...\n";                     // 请求一个名
    cin>>name;                                        // 读取名
    cout<<"The name you entered was "<<name
}
```

1.7 函数原型

在标准的C中,可以在一个函数被定义之前先声明它。函数声明刻划了函数的名,它的返回值,以及它的参数数目及类型。这个特性称为"函数原型"(function prototype),它使得编译程序可以将函数调用与原型比较,并强制做类型检查。

C并不要求每个函数都有原型。如果没有给出函数原型,最多也就是出现一条警告。但是,C++却要求每个函数都有一个原型。

下面的示例程序利用一个名为display的函数来打印"Hello,world"。

```
// 一个没有函数原型的程序
// 注意:它将通不过编译
#include <iostream.h>

void main()
{
    display("Hello,World\n");
}

void display(char *s)
{
    cout<<s;
}
```

由于display函数没有原型,故这个程序通不过C++编译程序的语法检查。

在前一个例子的基础之上,下一个例子加了个函数原型。该程序能通过编译,不出现错误。

```
// 一个带有函数原型的程序
#include <iostream.h>

void display(char *s);

void main()
{
    display("Hello,World");
}
```