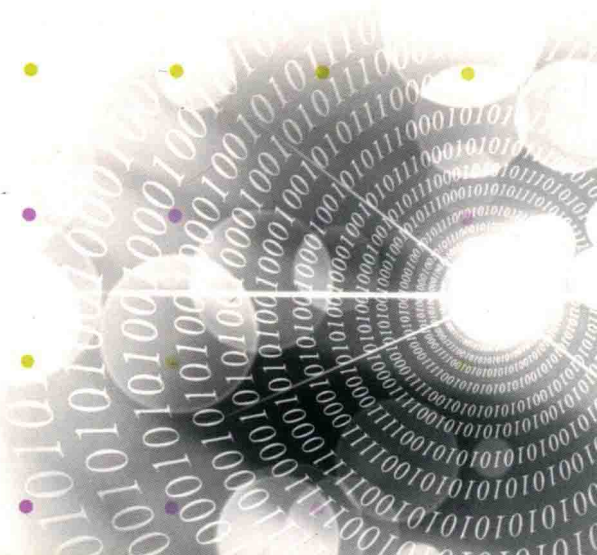




应用型本科信息大类专业“十三五”规划教材



Visual C++ 程序设计教程

主编 彭玉华 黄薇 刘艳



华中科技大学出版社

<http://www.hustp.com>



应用型本科信息类专业“十三五”规划教材

Visual C++ 程序设计教程

主 编 彭玉华 黄 薇 刘 艳

副主编 冯春华 阳小兰 吴 亮 江连海



点的理解程度；实验题是对能力的考查，要求学生在理解的基础上，通过动手操作，完成实验任务。本书不仅适合于教学，也是适合于开发人员学习和参考。本书在编写过程中得到了武昌理工学院领导和同仁的大力支持，在此表示衷心的感谢。



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

内 容 简 介

本书以 Visual Studio 2010 为开发环境,主要介绍了 C++ 面向对象程序设计和 Windows 程序开发的技巧和方法。主要内容包括:C++ 语言基础,C++ 面向对象程序设计的类、对象、继承、重载、多态、虚函数和流等,对话框、菜单、工具栏和状态栏、常用控件、图形和文本处理、文档/视图、ADO 数据库编程技术,以及学生信息管理系统项目开发实例。同时,本书每章配备了大量的例题、习题和实验题,并有项目开发案例,能帮助读者快速掌握 Visual C++ 程序设计及其应用。

本书通俗易懂,重点突出,注重实际应用,主要培养学生程序设计应用能力和项目开发综合能力。本书不仅可以作为高等院校计算机专业或相关专业的教材,也可作为用户的自学和参考书。

为了方便教学,本书还配有教学课件等教学资源包,任课教师和学生可以登录“我们爱读书”网(www.ibook4us.com)免费注册并浏览,任课教师可以发邮件至 hustpeiit@163.com 免费索取。本书教学资源包中提供的源代码全部经过精心测试,能够在 Windows XP、Windows 8 系统下编译和运行。

图书在版编目(CIP)数据

Visual C++ 程序设计教程/彭玉华,黄薇,刘艳主编. —武汉:华中科技大学出版社,2017.12
应用型本科信息大类专业“十三五”规划教材
ISBN 978-7-5680-3641-2

I. ①V… II. ①彭… ②黄… ③刘… III. ①C 语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 308430 号

Visual C++ 程序设计教程

Visual C++ Chengxu Sheji Jiaocheng

彭玉华 黄薇 刘艳 主编

策划编辑:康序

责任编辑:史永霞

责任监印:朱玢

出版发行:华中科技大学出版社(中国·武汉)

电话:(027)81321913

武汉市东湖新技术开发区华工科技园

邮编:430223

录排:武汉正风天下文化发展有限公司

印刷:武汉市籍缘印刷厂

开本:787mm×1092mm 1/16

印张:24

字数:625 千字

版次:2017 年 12 月第 1 版第 1 次印刷

定价:48.00 元



华中出版

本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前言

PREFACE

编者通过多年 C++ 面向对象程序设计教学实践和 Visual C++ 项目开发经历,结合高校应用型人才培养教学模式,突出理论性、实践性,先进性,通俗性,精选案例,侧重以 C++ 语言为基础,实现 Windows 应用程序开发为目标,培养学生项目开发的应用能力。

本书以 Visual Studio 2010 为开发环境。第 1 至 3 章介绍 C++ 面向对象程序设计,第 1 章 C++ 语言基础,第 2 章 C++ 面向对象程序设计,第 3 章多态性与虚函数,选例完整,突出重点。第 4 至 7 章介绍 Windows 图形界面资源的创建和编程方法,重点介绍 MFC 应用程序的编程。第 4 章对话框,第 5 章菜单、工具栏和状态栏设计,第 6 章常用控件,第 7 章图形和文本处理。第 8 章文档/视图程序设计,让读者加深对 Windows 应用程序的来龙去脉的理解。第 9 章数据库应用及项目开发实例,设计思路清晰,设计过程完整,所有实例代码均调试通过。

本书各章提供了一定数量的习题和实验题,习题主要考查学生对基本知识的理解程度;实验题是对能力的考查,要求学生具有一定的设计能力。实验题有完整的操作步骤和代码,便于学生掌握操作过程和代码编写。

本书不仅适合于教学,也适合于用 Visual C++ 编程和开发应用程序的开发人员学习和参考。

本书在编写过程中得到了武昌理工学院信息工程学院魏绍炎教授指导,由武昌理工学院彭玉华和黄薇、武汉工程科技学院刘艳担任主编。武汉工程科技学院冯春华、武昌理工学院阳小兰和吴亮、青岛理工大学琴岛学院江连海担任副主编。其中第 2、4、9 章由彭玉华编写,第 6 章由黄薇编写,第 7 章由刘艳编写,第 8 章由冯春华编写,第 1 章由阳小兰编写,第 3 章由吴亮编写,第 5 章由江连海编写。高翠芬、赵永霞、温静、李娟、向紫欣、毛玲、谢文亮、陶枫、汪潇、朱逢园、曾秀莲、李婵飞、陈静为本书编写提供了不少素材。全书由彭玉华统稿。

为了方便教学,本书还配有教学课件等教学资源包,任课教师和学生可以登

录“我们爱读书”网(www.ibook4us.com)免费注册并浏览,任课教师可以发邮件至 hustpeiit@163.com 免费索取。本书教学资源包中提供的源代码全部经过精心测试,能够在 Windows XP、Windows 8 系统下编译和运行。

在本书编写、修改过程中,作者参考了参考文献中列举的书籍及资料,在此向这些资料的作者表示诚挚的谢意!鉴于作者水平有限,书中难免有不当和错误之处,敬请广大读者批评、指正。

编者

2017 年 9 月

目录

CONTENTS

第 1 章 C++ 语言基础	(1)
1.1 C++ 概述	(1)
1.2 C++ 的输入与输出	(5)
1.3 C++ 的函数	(16)
1.4 指针和引用	(26)
1.5 C++ 新增运算符	(31)
1.6 Visual Studio. NET 集成开发环境	(34)
习题 1	(36)
实验 1 学生成绩计算	(41)
第 2 章 C++ 面向对象程序设计	(43)
2.1 类和对象	(44)
2.2 类的共享	(61)
2.3 类的继承和派生	(69)
习题 2	(87)
实验 2 使用面向对象方法实现学生成绩计算	(94)
第 3 章 多态性与虚函数	(103)
3.1 类的多重继承	(103)
3.2 类的多态性	(106)
3.3 运算符的重载	(111)
3.4 模板	(119)
习题 3	(123)
实验 3 使用面向对象方法实现学生成绩计算	(131)

第 4 章 对话框	(138)
4.1 MFC 应用程序	(138)
4.2 对话框的使用	(140)
4.3 消息对话框	(155)
4.4 通用对话框	(158)
习题 4	(165)
实验 4 登录对话框	(165)
第 5 章 菜单、工具栏和状态栏设计	(169)
5.1 菜单设计	(169)
5.2 工具栏	(180)
5.3 状态栏	(184)
5.4 菜单、工具栏和状态栏综合实例	(187)
习题 5	(190)
实验 5 菜单、工具栏和状态栏综合应用的单文档程序	(191)
第 6 章 常用控件	(193)
6.1 控件概述	(193)
6.2 静态控件和按钮	(196)
6.3 编辑框和旋转按钮	(201)
6.4 列表框	(209)
6.5 组合框	(215)
6.6 滚动条	(220)
6.7 滑动条	(225)
6.8 进度条	(229)
6.9 日期时间控件、图像列表和标签控件	(234)
6.10 文件系统控件	(242)
习题 6	(245)
实验 6 常用控件使用	(246)
第 7 章 图形和文本处理	(252)
7.1 图形设备接口概述	(252)
7.2 简单图形绘制	(259)
7.3 文本处理	(263)
7.4 图像处理	(269)
习题 7	(270)

实验7 绘图对象使用	(271)
第8章 文档/视图程序设计	(276)
8.1 文档/视图结构	(276)
8.2 文档和视图应用示例	(287)
习题8	(291)
实验8 文档/视图打印的应用程序	(292)
第9章 数据库应用及项目开发实例	(298)
9.1 数据库应用技术概述	(298)
9.2 使用ADO访问数据库	(299)
9.3 学生信息管理系统项目开发实例	(318)
习题9	(374)
参考文献	(375)

1.1 C++概述

C++语言是一种新型的、面向对象的计算机程序设计语言,既适合用来编写系统程序,也适合用来编写应用程序软件。自20世纪80年代由ALG-T及理查德·斯托曼(Richard Stallman)博士创建C++以来,C++日益受到重视并得到广泛应用,已成为程序设计最主要语言之一。

1.1.1 C++语言的起源

C++语言是从C语言发展而来的。C语言是一种面向过程的程序设计语言,实现了模块化和结构化,在处理小规模程序时,显得比较方便,但是在处理大规模、复杂的程序时,就显得明显不足。因为C语言对程序员的要求比较高,要求程序设计人员必须全面、细致地设计程序的每一个步骤和细节,整体规划程序的各个环节,显得烦琐,不适合设计大型软件。针对这个问题,在20世纪80年代提出了面向对象的程序设计方法,客观模仿事物被组合在一起的方式,将人们的习惯思维和表达方式运用在程序设计中,程序设计人员可以按照人们通常习惯的思维方式进行程序设计,从而设计出更可靠、更容易理解、可重用性更强的程序。C++就是在这种情况下产生的,自1983年C++诞生以来,就受到了人们普遍重视,特别是在1998年推出了C++国际标准版本以后,C++更是得到了飞速发展,成为当代程序设计的主流语言。

1.1.2 C++程序的特点

C++是C语言的扩充和延伸,C++既保留了C语言原有的许多优点,又扩充了C语言的一些新功能,增加了面向对象的机制。C++拥有丰富的运算符和数据结构,采用结构化程序设计方法,具有极好的移植性和高效的机器代码。程序设计更加简单灵活,无类型,C++既

第1章

C++语言基础

本章要点

- C++输入和输出操作
- C++函数的定义和调用
- C++新增的运算符
- Visual C++.NET 编程环境的使用

C++是一种面向对象的程序设计语言,也是目前最受欢迎的程序设计语言之一。面向对象程序设计思想来源于客观世界,符合人们的思维习惯,是未来最重要的编程思想。目前C++已经成为学习面向对象程序设计的基础语言,学习C++,可以快速掌握面向对象程序设计的思想。本章首先简要概述C++语言的起源、C++程序的特点、C++程序与C程序的比较及C++程序的基本结构等基础知识,然后介绍C++程序中的输入与输出流、C++程序中的函数、C++新增的运算符等基础知识。

1.1 C++概述

C++语言是一种新型的、面向对象的计算机程序设计语言,既适合用来编写系统程序,也适合用来编写应用程序软件。自20世纪80年代由AT&T贝尔实验室的Bjarne Stroustrup博士创建C++以来,C++日益受到重视并得到广泛应用,已经成为程序设计最主要的语言之一。

1.1.1 C++语言的起源

C++语言是从C语言发展而来的。C语言是一种面向过程的程序设计语言,实现了结构化和模块化,在处理小规模程序时,显得比较方便,但是在处理大规模、复杂的程序时,就显得明显不足。因为C语言对程序人员的要求比较高,要求程序设计人员必须全面、细致地设计程序的每一个步骤和细节,整体规划程序的各个环节,显得很烦琐,不适合设计大型软件。针对这个问题,在20世纪80年代提出了面向对象的程序设计方法,客观模仿事物被组合在一起的方式,将人们的习惯思维和表达方式运用在程序设计中,程序设计人员可以按照人们通常习惯的思维方式进行程序设计,从而设计出更可靠、更容易理解、可重用性更强的程序。C++就是在这种情况下产生的,自1983年C++诞生以来,就受到了人们普遍重视,特别是在1998年推出了C++国际标准版本以后,C++更是得到了飞速发展,成为当代程序设计的主流语言。

1.1.2 C++程序的特点

C++是C语言的扩充和延伸,C++既保留了C语言原有的许多优点,又扩充了C语言的许多功能,增加了面向对象的机制。C++拥有丰富的运算符和数据结构,采用结构化程序设计方法,具有良好的移植性和高效的机器代码,程序设计更加简单灵活、方便快捷。C++既

支持传统的面向过程程序设计,又支持当前的面向对象程序设计(object oriented programming,简称 OOP),是一种新型的程序设计语言,完全兼容 C 语言,用 C 语言编写的程序可以不加修改地在 C++ 中使用。C++ 提出了将数据和数据操作方法封装在一起,作为一个整体,称之为“对象”,每个对象都属于一个类型,称之为“类”,类中的数据用本类的方法进行处理,并通过继承、派生、重载和多态性等特征,实现了程序代码的重复应用和程序自动生成。

1.1.3 C++ 程序与 C 程序的比较

C++ 程序和 C 程序有许多相同之处,可以通过下面三个实例来进行比较。

例 1.1 用 C 语言编程求一个边长为 6 的正方形的面积。

```
#include<stdio.h>

void main()
{
    int L,S;
    scanf("%d",&L);           //输入边长 L 的值
    S=L*L;
    printf("%d\n",S);         //输出面积 s 的值
}
```

运行程序时输入 6,结果为:

36

例 1.2 用 C++ 语言编程求一个边长为 6 的正方形的面积。

```
#include<iostream>           //头文件
using namespace std;         //使用命名空间 std
void main()
{
    int L,S;
    cin>>L;                   //输入边长 L 的值
    S=L*L;
    cout<<S<<endl;           //输出面积 S
}
```

运行程序时输入 6,结果为:

36

这两个程序分别是用 C 和 C++ 来编写的,可以看出,程序在结构上基本是相同的,都是以 main 函数作为入口函数,以一对大括号“{}”把函数中的语句括起来,以分号作为语句结束标志,因为两者都是采用面向过程的方法。但是,也有一定的区别,如表 1-1 所示。

表 1-1 C 和 C++ 的区别

语 言	输入输出头文件	输出流 cout	输入流 cin
C++	iostream. h	“<<”插入符	“>>”提取符
C	stdio. h	printf()	scanf()

#include<iostream>与传统的#include<iostream. h>相比少了. h 后缀,因为 Visual C++ .NET 删除了旧的 iostream 库,而 std 命名空间仍有 C++ 标准库的定义,但没有对应头文

件.h 后缀,所以在 `#include<iostream>` 后要加入语句“`using namespace std;`”命名空间。

“`using namespace std;`”,即使用命名空间 `std`,其作用就是规定该文件中使用的标准库函数都是在标准命名空间 `std` 中定义的。命名空间就是将多个变量和函数等包含在内,使其不会与命名空间外的任何变量和函数等发生重命名的冲突。

C++ 语言除了支持面向过程的程序设计方法外,还支持面向对象的程序设计方法。下面的实例采用的是面向对象的程序设计方法来实现这一功能的。

例 1.3 用 C++ 语言编程求一个边长为 6 的正方形面积的程序。

```
#include<iostream>
using namespace std;
class Area                //声明一个名为 Area 的类
{
public:                   //声明公有访问权限的成员
    Area(int newLength);  //声明 Area 类的构造函数
    ~Area();              //声明 Area 类的析构函数
private:                 //声明私有访问权限的成员
    int Length, Square;
};
Area::Area(int newLength) //Area 类构造函数的具体实现
{
    Length=newLength;
}
Area::~~Area()            //Area 类析构函数的具体实现
{
    Square=Length*Length;
    cout<<Square<<endl;
}
void main()               //主函数
{
    Area myArea(6);       //声明一个 Area 类的对象 myArea,自动调用构造函数
}                          //退出 main() 函数时自动调用析构函数释放对象 myArea
```

运行结果为:

36

例 1.3 采用面向对象的程序设计方法,程序的功能与例 1.1、例 1.2 完全一样。首先声明了一个类名为 `Area` 的类,在类中又声明了公有的构造函数 `Area()` 与析构函数 `~Area()`,以及私有变量 `Length` 和 `Square`,在主函数中直接声明了 `Area` 类的对象 `myArea` 并赋予参数值 6。该对象会调用类中的公有构造函数,通过参数传递将值 6 传递给私有变量 `Length`,在主函数运行结束时释放对象 `myArea`,并调用类中的公有析构函数,在析构函数中计算私有变量 `Square` 的值并输出。通过这样的方法最后计算出正方形的面积。有关该程序中的公有构造函数与析构函数、私有变量、类和对象的具体概念及声明方式将在第 2 章具体讲述。读者可能觉得这种方法比前面的两种方法要麻烦,但在实际开发软件时,会逐渐感受到采用面向对象的方法给程序开发带来的好处。

通过上面的三个实例可以看出:用 C++ 语言编写的程序和 C 语言编写的程序在程序结构上基本是相同的,例如程序由函数构成,并且都是从 `main()` 函数开始执行的,语句分为

说明语句和执行语句,用分号“;”作为语句结束的标志;但是两者之间又不完全相同,说明语句就是说明定义变量、函数、结构等,执行语句则通知计算机完成一定的操作。

1.1.4 C++程序的基本结构

完整的C++程序一般包含类、普通函数和主函数。其中,普通函数和主函数与C语言中的类似,类是C++新增加的一个概念,也正是因为有了它,才使得C++可以进行面向对象的程序设计。在C++类中的成员一般有函数成员和数据成员,将数据和数据操作方法封装在一起,采用信息屏蔽的原则,减少了成员与外界的联系,提高了数据的安全性,使程序设计更加方便、灵活。类中成员的访问属性通常包括公有成员、私有成员和保护成员,使类中的成员具有不同的访问权限,从而实现类的封装性、继承性和多态性。

通过下面一个实例来了解C++程序的基本结构。

例 1.4 计算长方形的面积。

```
#include<iostream>
using namespace std;
class RectArea                                //声明一个名为 RectArea 的类
{
public:                                        //公有类型函数成员
    void input(int a,int b)
    {
        Length=a;
        Width=b;
    }
    void output()
    {
        int Square=Length*Width;
        cout<<Square<<endl;                //输出 Square 值并换行
    }
private:                                    //私有类型数据成员
    int Length,Width;
};                                           //“;”为类声明的结束标志
void show()                                  //普通函数
{
    cout<<"The area is:"<<endl; //输出一行提示文本
}
void main()                                  //主函数
{
    RectArea myRectArea;                    //声明 RectArea 类的对象 myRectArea
    int a,b;
    cin>>a>>b;                              //输入 a,b 的值
    show();                                  //调用普通函数 show()
                                           //通过对象 myRectArea 来访问类中的成员函数
    myRectArea.input(a,b);
    myRectArea.output();
}
```

例 1.4 程序运行时需要输入两个整型数,分别表示长方形的长与宽,以空格分开。假如输入 6 和 2,则运行结果为:

The area is:

12

事实上,C++程序通常由编译预处理命令、函数、语句、对象、变量、类、输入、输出及注释等几个部分组成。

(1) 编译预处理命令。在 C++程序的开头的行,以“#”作为标记的命令项(该命令不以“;”做语句的结束标志),称为编译预处理命令。编译预处理命令是在程序被正常编译之前执行的,故又称为预处理命令。C++提供的编译预处理命令主要有宏定义命令、文件包含命令和条件编译命令三种。

(2) 函数。函数是构成 C++程序的基本单位,一个 C++程序由若干个函数构成。函数包含函数说明部分和函数体,函数体一般由说明语句和执行语句构成。C++的函数分为两类:由 C++系统提供的标准库函数和用户根据需要自己定义的函数。另外,一个 C++程序中必须有一个而且只能有一个主函数。主函数名是 main(),它是程序的入口,主函数可以调用其他函数,其他函数可以相互调用,但是其他函数不能调用主函数。

(3) 语句。语句是构成 C++程序的基本单元,用来描述程序的执行过程,以分号“;”作为语句结束标志。一条语句可以占用多行,多条语句可以安排在一行中。

(4) 对象和变量。大多数 C++程序都有对象和变量,变量的类型很多,基本类型有整型、实型、字符型等,对象是一种“类”类型的变量。

(5) 类。类是 C++中特殊的数据类型,类包含数据成员和成员函数。类典型的特征是将数据和数据操作方法封装在一起,作为一个整体使用。

(6) 输入与输出。C++不仅保留了 C 语言的输入、输出系统,还增加了标准输入、输出流。使用标准输入、输出流时,用 #include 将其头文件包含进来。

(7) 注释。C++允许在程序中使用注释来提高程序的可读性。注释方法有以下两种:

① 以“//”开头的文字一直到行尾,称为行注释;

② 兼容 C 语言的注释,即把注释内容放在一对符号“/*”与“*/”之间,可以占用多行,称为段落注释。



1.2 C++的输入与输出

1.2.1 C 语言中的 printf 和 scanf 的缺陷

C 语言程序一般采用 scanf 函数和 printf 函数来进行数据的输入和输出操作。例如在下面这段程序中,采用 scanf 函数实现从键盘输入两个十进制整数,采用 printf 函数将输入的两个十进制数输出到终端(显示器)。

例 1.5 输入两个十进制整数并输出。

```
#include<stdio.h>
void main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d,%d\n",a,b);
}
```

在 C 语言中要执行输入和输出操作,必须指定输入或输出数据的格式,这是 C 语言的缺陷。在例 1.5 的程序中,分别输入和输出两个十进制整数,都使用了“%d”格式控制。如果要输入、输出不同类型的数据,必须指定不同的数据格式控制。这给编写程序带来了一定的难度。在 C++ 中,这个问题得到了很好的解决。

1.2.2 标准输入/输出(I/O)流类

C++ 保留了 C 语言的输入、输出系统,可以直接使用 C 语言输入、输出方式进行数据的输入、输出。另外,在 C++ 的编译系统中,有一个 I/O 流类库,数据的输入与输出可以通过 I/O 流来处理。“流”是 C++ 的一个重要概念,数据从一个对象到另一个对象的流动抽象为“流”,它负责在数据的生产者与数据的消费者之间建立联系并管理数据的流动。当数据从外部对象流向程序时称为“输入流”,当数据从程序流向外部对象时称为“输出流”。使用 C++ 的 I/O 标准流进行数据的输入、输出操作时,系统能自动完成数据类型的转换。表 1-2 列出了 C++ 的 I/O 标准输入、输出流类。

表 1-2 I/O 标准输入、输出流类

类 名	说 明	包含文件
抽象流基类	ios	流基类 iostream. h
输入流类	istream	通用输入流类和其他输入流的基类 iostream. h
	ifstream	输入文件流类 fstream. h
	istream_withassign	cin 的输入流类 iostream. h
	istrstream	输入字符串流类 strstrea. h
输出流类	ostream	通用输出流类和其他输出流的基类 iostream. h
	ofstream	输出文件流类 fstream. h
	ostream_withassign	cout、cerr 和 clog 的输出流类 iostream. h
	ostrstream	输出字符串流类 strstrea. h
输入/输出流类	iostream	通用输入/输出流类和其他输入/输出流的基类 iostream. h
	fstream	输入/输出文件流类 fstream. h
	strstream	输入/输出字符串流类 strstrea. h
	stdiostream	标准 I/O 文件的输入/输出类 swtdiostr. h
流缓冲区类	streambuf	抽象流缓冲区基类 iostream. h
	filebuf	磁盘文件的流缓冲区类 fstream. h
	strstreambuf	字符串的流缓冲区类 strstrea. h
	stdiobuf	标准 I/O 文件的流缓冲区类 stdiostr. h
预先定义的流初始化类	istream_init	预先定义的流初始化类 iostream. h

为了方便程序设计,C++ 系统提供了 4 个预定义的标准 I/O 流对象,即 cin、cout、cerr 和 clog,如表 1-3 所示。使用这些对象时,必须在程序的开头用 # include 将其头文件 iostream. h 包含进来。

表 1-3 C++系统预定义的标准 I/O 流对象

流对象名	含 义	缺省设备
cin	标准输入	键盘
cout	标准输出	显示器
cerr	标准错误输出	显示器
clog	cerr 的缓冲形式	显示器

1. 数据的输入

数据的输入可以使用输入流对象 cin, cin 代表标准输入设备(键盘), 使用提取符“>>”, 表示从设备(键盘)获取数据送到输入流对象 cin 中, 然后送到内存中的变量。使用 cin 可以获得多个从键盘的输入值, 其具体使用格式如下:

cin>>表达式 1>>表达式 2>>...>>表达式 n;

其中, 提取符后面的表达式通常是获取输入数据的变量或对象。例如:

```
int x,y;
double m,n;
cin>>x>>y>>m>>n;
```

此时, 要求从键盘输入两个整数和两个实数, 输入时, 数值之间用空格(空格数量不限)或回车隔开。执行上述语句, 用户可以输入:

```
6 8 7.5 8.9 //输入时中间加空格
```

或

```
6 //输入时按回车
8
7.5
8.9
```

变量 x 获取值为 6, 变量 y 获取值为 8, 变量 m 获取值为 7.5, 变量 n 获取值为 8.9。

注意: 使用输入流对象 cin 可以输入任何类型的数据, 且不需要在提取符后面指定数据的类型。

2. 数据的输出

数据的输出可以使用输出流对象 cout, cout 代表标准输出设备(显示器), 使用插入符“<<”, 将数据输出到流对象 cout(显示器)中, 其具体使用格式如下:

cout<<表达式 1<<表达式 2<<...<<表达式 n;

其中, 插入符后面的表达式通常是要输出到屏幕的常量、变量、字符串、转义字符、对象或其他表达式。

例如, 将 x, y 的值以及 x+y 的值在屏幕上显示出来, 可用如下命令:

```
int x=3,y=4;
cout<<"x="<<x<<"\t"<<"y="<<y;
cout<<"\nx+y="<<x+y<<endl; //其中字符串"x+y="作为提示文本原样输出
```

执行上述语句, 输出结果为:

```
x=3 y=4
x+y=7
```

对比一下 C 语言中的输入和输出,不难看出 C++ 语言要方便得多。

注意:使用输出流对象 cout 可以输出任何类型的数据,且不需要在插入符后面指定数据的类型,系统会自动按数据本身的类型进行输出。

1.2.3 I/O 格式控制符

使用 cin 和 cout 进行输入与输出操作时,不管输入或输出的数据是什么类型,它们都能够自动地按照正确的默认格式处理。但有时需要设置特殊的格式,这时就要用到格式控制符。

在 C++ 的 I/O 流库中提供了一些格式控制符,这些控制符可直接嵌入到 I/O 语句中,实现 I/O 的格式控制。表 1-4 给出了一些常用的格式控制符。但必须注意的是,在使用 setprecision(int p)、setw(int w) 和 setfill(char ch) 这些控制符时,在程序的开头除了包含“iostream.h”头文件外,还必须包含“iomanip.h”头文件。

表 1-4 C++ 常用的 I/O 格式控制符

控制符	含义	控制符	含义
dec	数值用十进制表示	ends	插入空字符
hex	数值用十六进制表示	flush	刷新流,不插入换行符
oct	数值用八进制表示	setprecision(int p)	设置实数输出的有效位数为 p
ws	提取空白字符	setw(int w)	设置输出域宽为 w
endl	插入换行符,并刷新	setfill(char ch)	设置填充字符为 ch

例 1.6 使用 cout 格式控制符输出。

```
#include<iostream>
#include<iomanip>
using namespace std;
void main()
{
    int nNum=123;
    double fNum=1.2345678,dNum=9.876543;
    cout<<"1234567890"<<endl;
    cout<<setw(10)<<nNum<<endl;           //整型数据默认靠右对齐,左边以空格填充
    cout<<setw(2)<<nNum<<endl;             //设置域宽小于实际长度时无效
    cout<<fNum<<endl;                     //实型数据默认有效数字为 6 位,靠左对齐
    cout<<setw(10)<<fNum<<endl;
    cout<<setw(3)<<fNum<<endl;
    cout<<setw(10)<<setfill('*')<<nNum<<endl;
    cout<<setw(10)<<fNum<<endl;           //设置域宽为 10
    cout<<ends<<setw(3)<<setprecision(2)<<fNum<<endl;
    cout<<dNum<<endl;
    cout<<setfill('#')<<setw(5)<<dNum<<endl; //设置域宽为 5,不足部分以“#”填充
    cout<<ends<<setw(4)<<setprecision(3)<<fNum<<endl;
}
```

运行结果如下：

```
1234567890
      123
123
1.23457
      1.23457
1.23457
*****123
***1.23457
      1.2
9.9
##9.9
1.23
```



注意：①使用 `setw(int w)` 用于设置数据输出占用的宽度，当其设置的宽度小于实际宽度时，该设置无效，数据按实际长度输出；`setw(int w)` 只对紧跟其后的数据有效，如果要设置多个数据的域宽，必须要多次使用 `setw(int w)`。

② 使用 `setprecision(int p)` 设置实型数据输出的有效数字个数，其有效位不包括小数点在内。`setprecision(int p)` 的作用域从使用开始一直到重新设置或程序结束为止。

③ `setfill(char ch)` 设置输出数据时填充字符，只能是单个字符，必须用单引号引起来。

在有些情况下，程序中数据需要以八进制和十六进制的形式输出。在输出流中插入控制符 `dec`（十进制）、`oct`（八进制）、`hex`（十六进制）即可。

例 1.7 不同进制数输出。

```
#include<iostream>
using namespace std;
void main ()
{
    int num;
    cout<<"请输入一个整型数据:";
    cin>> dec>> num;
    cout<<"八进制数是:"<<oct<<num<<endl;
    cout<<"十进制数是:"<<dec<<num<<endl;
    cout<<"十六进制数是:"<<hex<<num<<endl;
}
```

运行结果如下：

```
请输入一个整型数据:2010✓
八进制数是:3732
十进制数是:2010
十六进制数是:7da
```

1.2.4 输入输出操作的成员函数

`cin` 是 `istream` 类的对象，“>>”是提取操作符；`cout` 是 `ostream` 类的对象，“<<”是插