



普通高等教育“十一五”国家级规划教材
国家教学成果奖配套教材

高等学校计算机程序设计课程系列教材

C/C++程序设计教程

龚沛曾 杨志强 主编
高枚 许兰兰 雷新贤 编



高等教育出版社
Higher Education Press

内容提要

普通高等教育“十一五”国家级规划教材

国家教学成果奖配套教材

高等学校计算机程序设计课程系列教材

C/C++ 程序设计教程

龚沛曾 杨志强 主编

高 枚 许兰兰 雷新贤 编

图书在版编目(CIP)数据

C/C++ 程序设计教程 / 龚沛曾等主编. — 北京: 高等教育出版社, 2003.2

ISBN 7-04-010539-2

I. C... II. ①龚... III. C语言—教材 IV. TP312

设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2003)第047372号

责任编辑 黄英

封面设计 王大为

责任印制 余国珍

责任印制 余国珍

责任印制 余国珍

责任印制 余国珍

责任印制 余国珍

010-28281118 编辑 龚沛曾

400-810-0308 编辑 高枚

http://www.tup.com.cn 网址

http://www.tup.com.cn 网址

http://www.tup.com.cn 网址

http://www.tup.com.cn 网址

http://www.tup.com.cn 网址

出版发行 高等教育出版社

社址 北京市西城区德胜大街4号

邮政编码 100120

总机 010-28281000

发行 蓝色印刷图书发行有限公司

印刷 北京北航印刷厂

开本 787×1092 1/16

印张 18

字数 430 000

2003年2月第1版

2003年2月第1次印刷

24.00元

高等教育出版社

版次 第1版

印次 第1次

内容提要

本书是普通高等教育“十一五”国家级规划教材,国家教学成果奖配套教材。本书根据教育部《关于进一步加强高等学校计算机基础教学的意见》中有关计算机程序设计基础课程的教学要求而编写。

本书以 Visual C++ 为开发环境,主要包括:C/C++ 概述,基本数据类型、运算符和表达式,结构化程序设计,数组和指针,函数,结构和链表,文件,面向对象程序设计基础,并针对课程内容设置了 10 个实验。

本书按照“夯实基础、面向应用、培养创新”的指导思想更加重了教材的基础性、应用性和创新性,成功地将传统的面向过程程序设计和现代的面向对象程序设计两部分有机地结合在一起,体现了教材的先进性。

本书可作为高等学校程序设计课程教材,也可作为培训或自学教材。

图书在版编目(CIP)数据

C/C++ 程序设计教程/龚沛曾,杨志强主编. —北京:
高等教育出版社,2009.5

ISBN 978-7-04-026299-5

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

中国版本图书馆 CIP 数据核字(2009)第 047375 号

策划编辑	耿芳	责任编辑	俞丽莎	封面设计	于文燕	责任绘图	黄建英
版式设计	陆瑞红	责任校对	金辉	责任印制	毛斯璐		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京北苑印刷有限责任公司

开 本 787×1092 1/16
印 张 18
字 数 430 000

购书热线 010-58581118
免费咨询 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2009 年 5 月第 1 版
印 次 2009 年 5 月第 1 次印刷
定 价 24.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 26299-00

作者介绍



龚沛曾 1982年毕业于华东理工大学自动控制系。现为同济大学计算机科学与技术系教授,任教育部高等学校计算机基础课程教学指导委员会委员、全国高校计算机教育研究会副理事长、上海市计算机基础教育协会副理事长。长期从事计算机基础教学与研究工作,主持多项教育部教改项目。承担的国家“七五”攻关项目获上海市科技进步三等奖;主编的《Visual Basic 程序设计教程》、《大学计算机基础》于2003年、2007年连续两届获上海市优秀教材一等奖;主讲的“Visual Basic 程序设计”、“大学计算机基础”于2003年和2005年被评为国家精品课程;主持的教改项目获国家级教学成果二等奖、2001年和2005年上海市教学成果一等奖;享受国务院特殊津贴,获上海市优秀教育工作者、上海市首届高校教学名师奖等荣誉称号。



杨志强 1985年毕业于南京大学数理逻辑专业,工学博士,现为同济大学计算机科学与技术系教授。长期从事计算机基础教学工作,先后主讲了“Visual Basic 程序设计”、“大学计算机基础”、“数据库技术与应用”、“C/C++ 程序设计”等课程,参与编写了18部教材,先后获得上海市育才奖、同济大学优秀青年教师等荣誉称号。

前言

本书是普通高等教育“十一五”国家级规划教材,是根据教育部《关于进一步加强高等学校计算机基础教学的意见》的基本要求、为普通高等学校非计算机专业学生编写的教材。

随着计算机应用的不断深入,对大学生的程序设计能力有了更高的要求。大学生不仅要能够编写面向过程的结构化程序,而且还要掌握面向对象程序设计技术。为了解决内容多、课时少的矛盾,作者对“C/C++ 程序设计”课程进行了大胆的改革和尝试,对实际应用中很少使用的内容进行淡化或删除,并将传统的面向过程程序设计和现代的面向对象程序设计两部分内容有机地融合成一门课程。希望学生通过该课程的学习,一方面为后继计算机应用课程夯实基础;另一方面初步具备利用现代编程技术解决各自专业领域中计算机应用问题的能力。

本书共 8 章。前 7 章是传统的面向过程程序设计,介绍了基本概念和基础知识、结构化程序设计、数组、函数和文件等内容,这部分是传统 C 语言的内容,也是程序设计的基础;第 8 章面向对象程序设计基础,介绍了类和对象、继承和派生以及多态性等内容。

对于本书的学时数,作者建议课堂教学 36~54 学时,上机实践 36~54 学时。为克服内容多、课时少的矛盾,作者建议在课堂教学上强调培养学生的程序设计能力,语法细节让学生课后自学,对于复杂问题通过案例教学化繁为简。

本书由龚沛曾、杨志强主编。第 1 章由杨志强编写;第 2 章由雷新贤编写;第 3、4 章由龚沛曾编写;第 5 章由许兰兰编写;第 6、7、8 章由高枚编写。相应的实验由各自完成。

使用本书的学校可与作者联系索取相关的教学资源,作者邮件地址为 yzq98k@163.com。

由于作者水平所限,加之编写时间紧迫,因此错误和疏漏在所难免,恳请读者批评指正。

作者

2009 年 2 月于同济大学

目 录

第一篇 教 学 篇

第 1 章 C/C++ 概述 3

1.1 C/C++ 发展简史 3

1.2 简单的 C/C++ 程序实例 4

1.3 基本语法成分 6

1.4 数据的输入和输出 7

1.4.1 cout 和 cin 7

1.4.2 printf 和 scanf 函数 10

1.5 编译预处理 13

1.5.1 文件包含 13

1.5.2 宏定义 14

1.6 Visual C++ 简介 14

1.6.1 Visual C++ 集成开发环境 14

1.6.2 简单 C/C++ 程序的编写、 运行过程 16

1.6.3 程序调试 19

习题 21

第 2 章 基本数据类型、运算符

和表达式 23

2.1 数据类型 23

2.1.1 数据类型分类 23

2.1.2 基本数据类型的存储 24

2.2 变量和常量 25

2.2.1 变量 25

2.2.2 常量 26

2.3 运算符和表达式 28

2.3.1 运算符 29

2.3.2 表达式 33

2.3.3 数据类型转换 33

2.3.4 内部函数 35

2.4 指针 35

2.4.1 指针概念的引入 35

2.4.2 指针运算 37

2.4.3 new 和 delete 39

2.5 typedef 语句 41

2.6 应用举例 41

习题 43

第 3 章 结构化程序设计 46

3.1 顺序结构 46

3.1.1 表达式语句 46

3.1.2 空语句 47

3.1.3 复合语句 47

3.2 选择结构 47

3.2.1 if 语句 47

3.2.2 switch 语句 52

3.3 循环结构 54

3.3.1 while 语句 54

3.3.2 do-while 语句 55

3.3.3 for 语句 56

3.3.4 循环语句的嵌套 58

3.4 其他控制语句 59

3.4.1 goto 语句 59

3.4.2 break 语句和 continue 语句 60

3.5 应用举例 61

习题 67

第 4 章 数组和指针 72

4.1 一维数组 72

4.1.1 引例 72

4.1.2 一维数组的定义、存储和初始化	74	5.7.2 静态局部变量	135
4.1.3 数组元素的引用和基本操作	75	5.7.3 全局变量	136
4.1.4 数组排序	77	5.8 应用举例	139
4.2 二维数组	79	习题	143
4.2.1 二维数组的定义和初始化	80	第6章 结构和链表	152
4.2.2 二维数组的基本操作	81	6.1 结构类型	152
4.3 字符数组	84	6.1.1 结构类型的说明	152
4.3.1 字符数组的使用	84	6.1.2 结构变量的定义及初始化	153
4.3.2 字符串处理函数	86	6.1.3 结构变量的访问	155
4.4 指针和数组	87	6.2 结构的应用——链表	158
4.4.1 指针和一维数组	87	6.2.1 链表的基本操作方法	159
4.4.2 指针和二维数组	89	6.2.2 链表的建立	160
4.4.3 指针和字符串	92	6.2.3 单链表的基本操作	162
4.5 应用举例	94	6.2.4 带头结点的单链表	167
习题	100	6.3 应用举例	170
第5章 函数	105	习题	175
5.1 函数的定义、调用和说明	105	第7章 文件	180
5.1.1 引例	105	7.1 文件概述	180
5.1.2 函数定义	106	7.1.1 C/C++ 文件	180
5.1.3 函数调用	109	7.1.2 文件指针	181
5.1.4 函数说明	111	7.2 文件的打开与关闭	182
5.2 函数间的参数传递	112	7.2.1 文件的打开	182
5.2.1 传值参数	112	7.2.2 文件的关闭	184
5.2.2 引用参数	114	7.3 文件的读/写	184
5.2.3 指针参数	116	7.3.1 文件的字符读/写	184
5.2.4 数组名作参数	122	7.3.2 文件的字符串读/写	186
5.3 递归函数	125	7.3.3 文件的数据块读/写	187
5.3.1 递归的概念	125	7.3.4 文件的格式化读/写	188
5.3.2 递归函数和递归调用	125	7.3.5 文件的随机读/写	189
5.3.3 递归调用的执行过程	126	7.4 应用举例	190
5.4 函数参数缺省	128	习题	194
5.5 函数重载	129	第8章 面向对象程序设计	198
5.6 函数模板	131	基础	198
5.7 作用域与存储类别	133	8.1 面向对象程序设计概述	198
5.7.1 自动变量	134	8.1.1 面向对象程序设计的产生	198
		8.1.2 面向对象的基本概念和特点	199

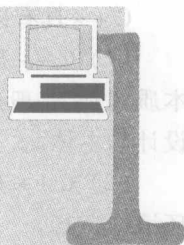
8.2 类和对象	200	8.3.2 单继承	221
8.2.1 类的定义	200	8.4 多态性	229
8.2.2 对象的定义	203	8.4.1 函数重载	230
8.2.3 对象的初始化	205	8.4.2 运算符重载	230
8.2.4 常对象和常成员	213	8.4.3 虚函数	233
8.2.5 静态成员	216	8.4.4 抽象类	237
8.2.6 友元函数	219	8.5 应用举例	240
8.3 继承和派生	220	习题	247
8.3.1 基类和派生类	220		

第二篇 实 验 篇

实验1 C/C++ 概述	255	实验6 函数	266
实验2 顺序结构程序	256	实验7 结构和链表	268
实验3 选择控制结构	258	实验8 文件	269
实验4 循环控制结构	260	实验9 对象和类	270
实验5 数组和指针	263	实验10 继承和多态性	272
参考文献	274		

第一篇

教 学 篇



第 1 章

C/C++ 概述

1.1 C/C++ 发展简史

1972 年,为了编写 UNIX 操作系统,美国贝尔实验室的 D. M. Ritchie 设计并实现了 C 语言。经过 3 年的改进和完善,C 语言走出了实验室,成为目前应用最广泛的程序设计语言之一。C 语言在其 30 多年的发展史上涌现了众多的版本,但普遍遵守两个重要的标准:一是 B. W. Kernighan 和 D. M. Ritchie 于 1978 年合著的 The C Programming Language,称为老标准;二是美国国家标准化协会(American National Standards Institute, ANSI)于 1983 年制定的新标准,称为 ANSI C。1988 年,B. W. Kernighan 和 D. M. Ritchie 根据 ANSI C 重新修订了 The C Programming Language,它成为了 C 语言的经典名著。两人由于在 C 语言和 UNIX 操作系统上的突出贡献,于 1983 年共同获得了计算机科学领域的最高奖——图灵奖。

当 C 语言程序达到一定的规模(25 000 ~ 100 000 行)后,维护和修改就会变得相当困难。为了满足管理程序复杂性的需要,贝尔实验室的 Bjarne Stroustrup 博士于 1979 年开始对 C 语言进行改进和扩充,并从 Simula 67 引入了面向对象程序设计的内容,最初取名为“带类的 C”,1983 年改名为 C++。在经历了 3 次重大修订后,于 1994 年制定了标准 C++ 草案,后又经不断完善,成为目前的 C++,并逐渐取代 C,如图 1.1.1 所示。

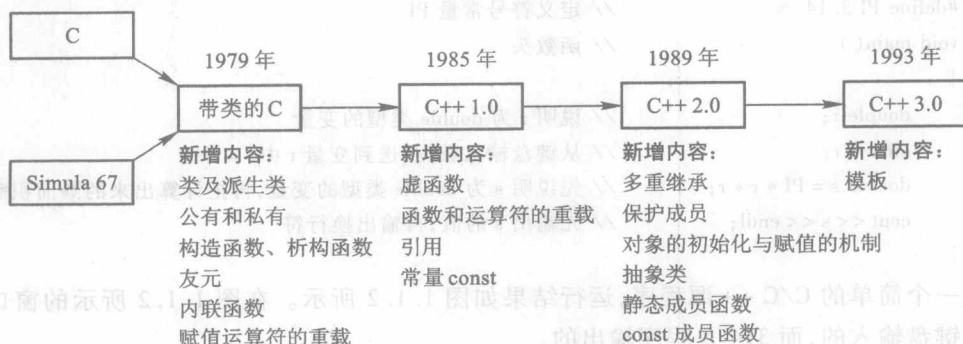


图 1.1.1 C/C++ 的发展

C++ 具有以下突出的优点:

(1) C++ 是 C 语言的超集。C++ 由两部分组成:一是过程性语言部分,这部分与 C 语言无本质区别,一般遵守 ANSI C 标准;二是类和对象部分,这是 C 语言所没有的,它是面向对象程序设计的主体。

(2) C++ 充分保持了与 C 语言的兼容性,绝大多数 C 语言程序可以不经修改直接在 C++ 环境中运行。

(3) C++ 仍然支持面向过程的程序设计,它是一种理想的结构化程序设计语言,又几乎包含了面向对象程序设计的全部特征。

(4) C++ 继承了 C 语言的高效、灵活等优点。用 Bjarne Stroustrup 博士的话来说,C++ 使程序“结构清晰、易于扩展、易于维护而不失效率”。

(5) C++ 是一种标准化的、与硬件基本无关的、广泛使用的程序设计语言,它具有很好的通用性和可移植性。C++ 程序通常无需修改,或稍作修改,即可在其他计算机系统上运行。

(6) C++ 具有丰富的数据类型和运算符,并提供了功能强大的库函数。

目前主要的 C++ 开发工具包括 Inprise 公司的 C++ Builder、Borland 公司的 Borland C++、Microsoft 公司的 Visual C++ 等。

本书的编程环境是 Visual C++。由于它既支持 C 语言程序,又能运行 C++ 程序,同时考虑到 C++ 对 C 语言的兼容性,因此教材前 5 章不再具体说明一个程序是 C 语言程序还是 C++ 程序,统一称为 C/C++ 程序。但是,需要读者注意的是,由于程序中使用了 C 语言中没有的标准输入/输出设备——cin 和 cout,所以,严格来说,教材中所有程序都是 C++ 程序。

1.2 简单的 C/C++ 程序实例

下面通过两个简单的实例,说明 C/C++ 源程序的基本组成。

【例 1.1】 输入圆的半径,求面积。

程序:

```
#include "iostream.h" // 文件包含命令
#define PI 3.14 // 定义符号常量 PI
void main() // 函数头
{
    double r; // 说明 r 为 double 类型的变量
    cin >> r; // 从键盘输入数据,送到变量 r 中
    double s = PI * r * r; // 先说明 s 为 double 类型的变量,再把计算出来的圆面积赋值给 s
    cout << s << endl; // 先输出 s 的值,再输出换行符
}
```

这是一个简单的 C/C++ 源程序,运行结果如图 1.1.2 所示。在图 1.1.2 所示的窗口中,10 是用户从键盘输入的,而 314 是程序输出的。

说明:

① 一行中以“//”开始的内容称为注释。它的作用是对程序进行说明,提高程序的可读性。在编译时,注释将被忽略。

② 一般来说,源程序开始几行是文件包含命令,其作用是指示编译预处理程序将指定的文件嵌入到该源程序文件中。有关文件包含命令的使用可参阅第 1.3 节。

③ 语句“#define PI 3.14”定义 PI 为符号常量,代表 3.14。在程序设计时,凡是需要书写 3.14 的地方都可以用 PI 代替;在编译预处理时,对源程序中所有的 PI 用 3.14 替换。

④ main() 函数称为主函数。C/C++ 程序必须包括一个 main() 函数,这是 C/C++ 程序的标志。main() 的一般形式为:

```
void main() //函数头,void 表示 main() 函数没有返回值
```

```
{
    语句组
}
```

} 函数体

⑤ 语句“double r;”是说明语句,说明 r 为 double 类型的变量,用于存放浮点数。

⑥ cin 和 cout 是标准输入输出设备,用来实现数据的输入输出。cin 一般代表键盘,语句“cin >> r;”的作用是将键盘输入的数据送到变量 r 中;cout 一般代表显示器,语句“cout << s << endl;”的作用是将 s 的值和换行符“endl”依次输出到屏幕上。

⑦ 语句“double s = PI * r * r;”有两个作用:一是说明变量 s 为 double 类型;二是计算圆的面积,然后赋值给 s。

【例 1.2】 求两个数之和。

程序:

```
#include "iostream.h"
int add(int x,int y) //定义 add() 函数,求两个数之和
{
    int z; //说明 z 为 int 类型的变量
    z = x + y; //将 x + y 的值赋给 z
    return z; //结束 add() 函数的执行,并将 z 的值返回给主函数
}
void main() //定义主函数
{
    int a,b,c; //说明 a,b,c 为 int 类型的变量
    cin >> a >> b; //从键盘输入两个数据送到 a 和 b 中
    c = add(a,b); //调用 add() 函数,计算 a + b,并把结果赋给 c
    cout << c << endl; //先输出 c 的值,再输出换行符
}
```

程序运行结果如图 1.1.3 所示。其中 10 和 33 是用户输入的数据,中间用空格分隔。

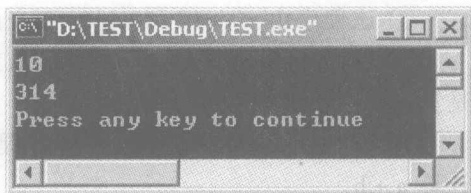


图 1.1.2 例 1.1 程序运行结果

说明:

- ① 该程序由 `main()` 和 `add()` 两个函数组成。
- ② 尽管 `add()` 函数在 `main()` 函数前面,但程序仍然是从 `main()` 函数开始执行的。
- ③ 语句“`c = add(a,b);`”的作用是调用 `add()` 函数计算 `a + b`。

④ `add()` 函数是一个用户自定义的函数,其功能是求两个数的和。调用时,先将实际参数 `a`、`b` 的值分别传递给对应的形式参数 `x`、`y`,然后依顺序执行 `add()` 函数中的各条语句,遇到语句“`return z`”后将 `z` 的值作为函数的返回值返回给主函数并结束 `add()` 函数的执行,继续执行 `main()` 函数中的语句,变量 `c` 将获得调用 `add()` 函数所返回的 `a + b` 的值。

从上面两个实例,可以看到 C/C++ 程序的组成及书写规则为:

- ① C/C++ 程序是由一个或多个函数组成的,其中必须要有一个 `main()` 函数。无论这个函数的位置在哪里,程序总是从它开始执行。
- ② 在一个函数内,语句的执行顺序是从上到下的。
- ③ 注释是从“`//`”开始到本行结束的内容。除了增加程序的可读性之外,它没有其他的任何作用。
- ④ C/C++ 程序书写形式自由。一行可以写多条语句,一条语句也可以分写在不同行中。C 语言中的大、小写字母是有区别的,这一点初学者要特别注意。

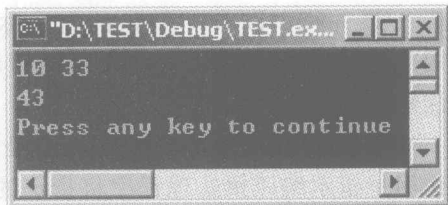


图 1.1.3 例 1.2 程序运行结果

1.3 基本语法成分

C/C++ 作为一种程序设计语言也有一个严格的字符集和一套严密的语法规则。程序中各种成分(如常量、变量、表达式、语句等)是根据语法规则由字符集中的字符构成的。程序中不能使用这个字符集以外的字符,也不能违反语法规则。

1. 字符集

- ① 大、小写英文字母各 26 个: `a ~ z` 和 `A ~ Z`。
- ② `0 ~ 9` 共 10 个数字。
- ③ 下划线: `_`。
- ④ 特殊符号:包括运算符在内的其他字符,这里暂不一一列举,详见第 2 章。

要特别注意的是,在 C/C++ 中大、小写字母是敏感的,即大、小写字母是不等价的、有区别的。

2. 关键字

关键字是指有特定含义的、专用的单词,如 `int`、`char`、`break`、`for`、`define` 等。用户不能用关键字来命名变量名、符号常量名、函数名、类名等。

3. 标识符

标识符是以字母或下划线开始的,由字母、数字字符和下划线组成的字符串,用来标识变量

名、符号常量名、函数名、类名等。

例如,下面的标识符都是合法的:

A2 student area_of_circle num _dd Int

而下面的标识符则不合法:

2A A-B area of circle(一个标识符) M. D int

使用标识符时,应注意以下几点:

- ① 由于 C/C++ 中大、小写字母不等价,因而像 Sum 和 sum 这两个标识符是不同的。习惯上,变量名小写,符号常量大写。
- ② 关键字不能作标识符用。例如, int、define 等关键字不能作为标识符使用。
- ③ 定义标识符时最好能简洁且“见名知义”,以提高程序的可读性。如 min 表示最小值, average 表示平均值, day 表示日期。
- ④ 在 Visual C++ 中,标识符中不能有汉字,但是字符串和注释中可以有汉字。

1.4 数据的输入和输出

数据的输入和输出是应用程序不可缺少的功能。没有输出,程序也就没有意义。在 C 语言中,数据的输入和输出是通过调用函数来实现的,主要有 scanf()、printf()、getchar()、putchar() 等。C++ 对数据的输入和输出进行了扩充,引入了标准设备 cin(代表键盘)和 cout(代表显示器),把数据的输入和输出处理为从一个对象到另一个对象的流动。

1.4.1 cout 和 cin

使用 cin 和 cout 进行输入和输出,用户不必考虑数据类型,也可以不考虑输入输出格式,一切都由系统自动完成,简单方便,形象直观。

要使用 cin 和 cout,需要在程序前面加语句:

```
#include "iostream. h"
```

或

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

1. cout

cout 称为标准输出设备,通常是指显示器。其使用形式为:

```
cout << 表达式 1 << 表达式 2 << ... << 表达式 n;
```

说明:

- ① 将各表达式的值按顺序输出到显示器上,数据的输出格式由系统自动决定。
- ② “<<”称为输出运算符,也称为插入运算符。
- ③ 各表达式可以是任意类型的。
- ④ 可以使用格式控制符控制数据的输出格式,如设置整数基数、数值表示形式、填充字符、精度、数据宽度等,参见表 1.1.1。若要使用格式控制符,应使用语句“#include "iomanip. h"”。

表 1.1.1 常用格式控制符

格式控制符	说 明	示 例	
		语 句	结 果
endl	输出换行符	cout << 123 << endl << 123	123 123
dec	十进制表示	cout << dec << 123;	123
hex	十六进制表示	cout << hex << 123;	7b
oct	八进制表示	cout << oct << 123;	173
setw(int n)	设置数据输出的宽度	cout << 'a' << setw(4) << 'b';	a b (中间有 3 个空格)
setfill(int n)	设置填充字符	cout << setfill('*') << setw(6) << 123;	*** 123
setprecision(int n)	设置浮点数输出的有效数字位数	cout << setprecision(5) << 123.456	123.46

【例 1.3】 cout 应用示例。

程序:

```
#include "iostream.h"
void main()
{
    int a=2;
    double b=3.32;
    char c='A';
    cout<<a<<' '<<b<<endl;
    cout<<c<<' '<<"abcd"<<endl;
}
```

程序运行结果如图 1.1.4 所示。

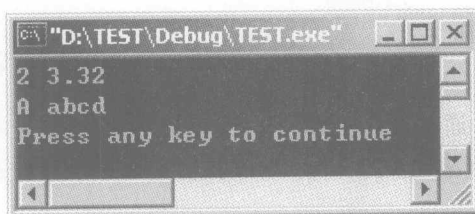


图 1.1.4 例 1.3 程序运行结果

【例 1.4】 使用格式控制符输出数据。

程序:

```
#include "iostream.h"
```

```
#include "iomanip.h"
void main()
{
    int x = 65;
    double f = 123.456;
    cout << "123456789012345" << endl;
    cout << dec << x << " " << hex << x << " " << oct << x << endl;
    cout << x << ends << x << endl;
    cout << f << endl;
    cout << setprecision(4) << f << endl;
    cout << setw(12) << f << endl;
    cout << setw(12) << setfill('#') << f << endl;
}
```

程序运行结果如图 1.1.5 所示。

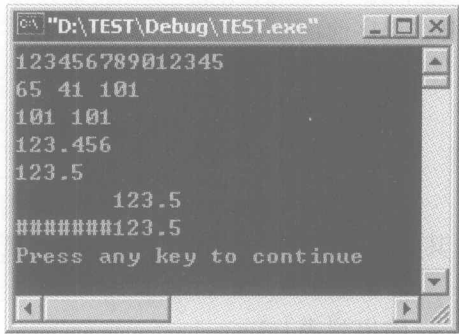


图 1.1.5 例 1.4 程序运行结果

2. cin

cin 称为标准输入设备,通常是指键盘。其使用形式为:

```
cin >> 变量 1 >> 变量 2 >> ... >> 变量 n;
```

说明:

- ① 从键盘输入数据,依次送入各变量中。
- ② “>>”称为输入运算符,也称为提取运算符。
- ③ 各变量可以是任意数据类型,输入时各个数据之间用空格、Tab 键或回车分隔。

【例 1.5】 cin 应用示例。

```
#include "iostream.h"
void main()
{
    int a;
    float b;
    cout << "input a,b: ";
    cin >> a >> b;
    cout << "a + b = " << a + b << endl;
}
```