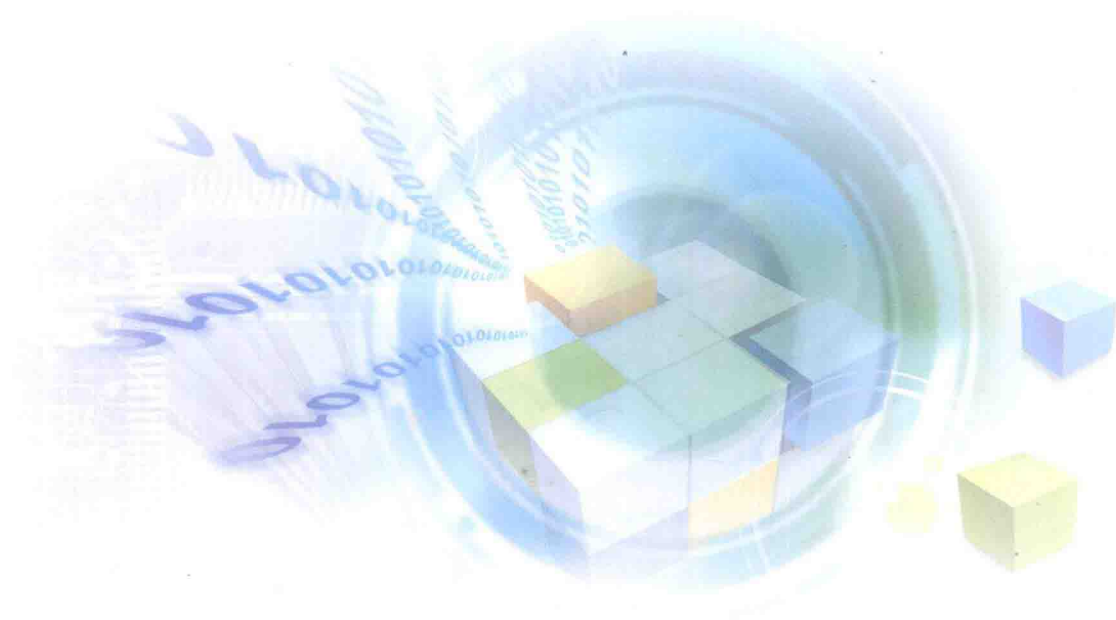


高等教育“十二五”规划教材

VC++ 程序设计 深度解析与指导

主 编 杨平乐 孙 娜

副主编 黄 霞 周 塔 李 佳 王 勇



中国矿业大学出版社

VC++ 程序设计深度解析与指导

主 编	杨平乐	孙 娜
副主编	黄 霞	周 塔
	李 佳	王 勇

中国矿业大学出版社

内 容 提 要

本书主要包括 C++ 程序设计基本概述、数据类型与表达式、流程控制语句、数组、函数、结构体与链表、类和对象、继承与多态性、友元与运算符重载、上机考试辅导等。

本书是基于最新的计算机等级考试大纲和历年考试真题,结合本课题组多年对等级考试的研究编写而成。全书内容紧扣大纲,涵盖了考试大纲规定的所有知识点,并有重点地进行了细化和深化。

本书不仅可以作为计算机等级考试培训教材和自学用书,也可以作为高等院校计算机程序设计课程的指定教材或自学参考书。

图书在版编目(CIP)数据

VC++ 程序设计深度解析与指导/杨平乐,孙娜主编.

徐州:中国矿业大学出版社,2014.1

ISBN 978 - 7 - 5646 - 2142 - 1

I. ①V… II. ①杨…②孙… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 286192 号

书 名 VC++ 程序设计深度解析与指导
主 编 杨平乐 孙 娜
责任编辑 王美柱
出版发行 中国矿业大学出版社有限责任公司
(江苏省徐州市解放南路 邮编 221008)
营销热线 (0516)83885307 83884995
出版服务 (0516)83885767 83884920
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 徐州中矿大印发科技有限公司
开 本 787×1092 1/16 印张 17 字数 414 千字
版次印次 2014 年 1 月第 1 版 2014 年 1 月第 1 次印刷
定 价 32.80 元

(图书出现印装质量问题,本社负责调换)

前 言

本书是《C++程序设计语言》的配套教材,其编写目的是为进一步巩固和加强学生对C++程序设计基本理论和基本知识的学习、理解与掌握,提高学生对C++程序设计的认识水平以及在应用过程中的实践动手能力。

作为计算机程序设计语言,其学习过程大致可分为以下几个阶段:

- (1) 了解并熟悉程序设计语言的语法规则及相关语义;
- (2) 解读教材中的经典例题,包括其中的算法和程序;
- (3) 模仿例题编写程序;
- (4) 在模仿的基础上不断提高分析问题、解决问题和编程实现的能力。

其中,前两个阶段主要由教师带领学生完成,而后两个阶段更多地需要学生主动自觉地完成。针对这一过程的特征,同时结合C++程序设计的特点,为了让学生易于学、乐于学,教师易于教、善于教,我们对传统教材的体系结构进行了调整,在突出重点的同时,将课程知识有机地加以分模块,既体现了循序渐进的原则,又方便学生在每次课程教学后的巩固复习。

本教材每章包括主要知识点、典型例题讲解、过关练习、章节测试题、上机实践、算法分析等。每章的内容模块化组织,方便学生理解和复习。本书以条目化的形式列出了每个章节学生应掌握的知识点。同时有针对性地选择了具有代表性的典型例题,帮助学生理解和掌握C++程序设计的基本方法。过关练习和章节测试题主要是由经验丰富的一线教师自主设计的习题和以往等级考试中经常出现的题目组成。上机实践部分主要从解决问题的角度出发,引导学生如何应用所学知识进行计算机编程。算法分析部分对每章所学的重要算法进行归纳和总结,便于学生学习和提高。

本书由江苏科技大学张家港校区计算机基础教学课题组集体编写,具体分工如下:杨平乐编写第1章、第5章、第8章;孙娜编写第2章、第7章;周塔编写第3章、第6章;王勇编写第4章;黄霞编写第9章、第10章。

本书在编写过程中,参考了业已出版的书籍和网络资料,在此,我们对这些书籍的作者以及提供网络资料的同仁表示由衷的感谢。此外,编写工作还得到了江苏科技大学张家港校区和苏州理工学院领导的关心和大力支持,还要特别感谢计算机科学与工程学院计算机基础教研室诸位同仁和学校教材中心给予的帮助,特别是祁云嵩、刘永良、吴慧英、刘广峰、

王红梅、汪燕、李佳等老师的积极参与并提出了宝贵意见,张家港校区张伟、荆天枢、周赞赞等十几位学生也参与了书稿程序的调试过程,还有很多给予帮助,但受限于篇幅不能一一提名的同仁和朋友,在此一并表示感谢,衷心祝愿大家明天更美好!

由于编者水平所限,且编写时间仓促,错漏之处在所难免,恳请读者及同行不吝赐教。

编 者

2013 年 12 月

目 录

第一章 概述	1
第一节 程序设计语言及 C++ 程序开发	1
第二节 章节测试题	3
第三节 上机实践	4
第二章 数据类型与表达式	5
第一节 标识符及基本数据类型	5
第二节 运算符和表达式	14
第三节 章节测试题	21
第四节 上机实践	22
第五节 本章算法分析	23
第三章 流程控制语句	24
第一节 操作运算语句及选择语句	24
第二节 流程控制语句	25
第三节 循环语句及其他控制语句	33
第四节 章节测试题	40
第五节 上机实践	46
第六节 本章算法分析	47
第四章 数组	49
第一节 数组的定义、存储及访问	49
第二节 指针及访问数组	62
第三节 章节测试题	75
第四节 上机实践	84
第五节 本章算法分析	87
第五章 函数	90
第一节 函数定义使用及其参数传递	90
第二节 函数与指针、函数的特性	99
第三节 作用域、存储类型和编译预处理	105
第四节 章节测试题	111

第五节 上机实践·····	118
第六节 本章算法分析·····	119
第六章 结构体与链表·····	122
第一节 结构体·····	122
第二节 链表·····	129
第三节 章节测试题·····	132
第四节 上机实践·····	139
第七章 类和对象·····	140
第一节 类和对象的关系·····	140
第二节 类的特殊成员函数·····	149
第三节 章节测试题·····	171
第四节 上机实践·····	180
第五节 本章算法分析·····	181
第八章 继承与多态性·····	183
第一节 类的继承、赋值兼容性·····	183
第二节 冲突和支配规则、虚函数与多态性·····	192
第三节 章节测试题·····	201
第四节 上机实践·····	211
第九章 友元与运算符重载·····	214
第一节 友元与运算符重载·····	214
第二节 章节测试题·····	228
第三节 上机实践·····	237
第十章 上机考试辅导·····	242
第一节 改错题·····	242
第二节 类的编程·····	254
参考文献·····	266

第一章 概 述

第一节 程序设计语言及 C++ 程序开发

【知识归纳】

知识点	主要知识要素	知识水平	学习要求
程序设计语言	机器语言		理解
	汇编语言		理解
	高级语言及其编译过程		掌握
程序设计思想	面向过程程序设计		理解
	面向对象程序设计		理解
C++ 程序开发	程序开发步骤	重点	熟练掌握
	程序框架	重点	熟练掌握
	VC++ 开发环境	重点	熟练掌握

【主要知识点】

一、机器语言

指令代码采用二进制(0、1)表示,称为机器语言,或称为低级语言。二进制序列指令十分难记。因此,其程序具有不可移植性,即对某一种体系结构(如 Intel CPU)的计算机编写的计算机程序,在另一种体系结构(如 AMD CPU)的计算机上不能运行。

二、汇编语言

为解决机器语言难掌握的问题,采用助记符代替二进制序列,即机器语言的符号法,称为汇编语言。汇编程序必须翻译为机器语言后,计算机才能识别运行。汇编程序仍与体系结构有关,具有不可移植性。但与机器语言一样具有运算效率高的特点。

三、高级语言

高级语言是一种更接近于社会生活中的自然语言,如两个数的加法可写为 $z=x+y$ 。高级语言程序与机器类型无关,具有可移植性、易学易记等特点,采用高级语言编写的程序称为源程序。但计算机不能直接识别源程序,必须将其译为机器语言后才能识别运算。

其过程分为两种:

一种是边翻译,边运行,翻译一句,执行一句,这种过程称为解释过程。程序不能离开其

解释环境。

另一种是编译语言,它是将整个源程序全部翻译成机器语言后,计算机才能运行,这个过程称为编译过程。源程序经编译后生成的机器语言程序称为目标程序,计算机不能直接运行目标程序,还必须经过链接过程,才能变为可执行文件(.exe 文件),以后可以脱离编译环境,直接在计算机上运行。C++语言采用编译执行。

四、面向过程程序设计

“面向过程”是一种以事件为中心的编程思想。

就是分析出解决问题所需要的步骤,然后用函数把这些步骤一步一步实现,使用的时候一个一个依次调用就可以了。但是当程序规模较大时,其有明显缺点。

比如拿学生早上起来的事情来说明面向过程,粗略的可以将过程拟为:(1)起床;(2)穿衣;(3)洗脸刷牙;(4)去学校。而这4步就是一步一步地完成,它的顺序很重要,只需一个一个的实现就可以。

五、面向对象程序设计

面向对象的程序设计方法用更加接近现实问题的对象模型对实际问题进行建模。对象模型将数据和函数封装在一起,数据用来描述实际问题的静态属性,函数用来描述实际问题的动态属性。程序由不同类型的多个对象组成,对象之间通过互发消息进行通信,它们相互配合协作,共同实现程序的功能。

与面向过程相比有如下优点:(1)更好的模块化;(2)更高级的抽象;(3)更好的信息隐藏性;(4)更好的可重用性。

六、程序的开发步骤

(1) 编辑程序。在开发环境 Visual C++6.0 提供的编辑器中编写程序(源程序缺省扩展名为 cpp)。

(2) 编译。通过开发环境 Visual C++6.0 把源程序编译成能够在本地计算机运行的目标程序。扩展名为 cpp 的源程序经编译后生成扩展名为 obj 的目标文件。

(3) 链接。用链接器将编译成功的目标文件与对应的系统模块链接成扩展名为 exe 的可执行文件。

上述3步过程可用图 1.1 来描述。

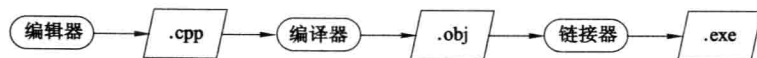


图 1.1 C++程序可执行文件的生成过程

七、程序框架

一个完整的 C++ 程序由注释、编译预处理和程序主体构成。

其程序结构如下:

[预处理语句]

[用户自定义函数]

主函数定义

其中,[]中的内容为可省略部分。

(1) 注释

单行注释符: // 该行后面为注释内容

多行注释符: /*
多行注释内容
*/

(2) 编译预处理

以符号“#”开始的行称为编译预处理行。编译预处理是 C++ 处理程序的工具。

如: #include <iostream.h> 的作用是在编译之前将文件 iostream.h 的内容插入程序中。

(3) 程序主体

主要包含主函数和用户自定义函数。主函数有且仅有一个,自定义函数可以有 0 个或多个。程序的运行从主函数开始,至主函数结束(只有当主函数调用某自定义函数时,该自定义函数才会运行)。

八、Visual C++ 6.0 开发环境

Visual C++ 6.0 开发环境是一个基于 Windows 操作系统,并包含 C++ 语言子集的可视化集成开发环境,它集编辑、编译、链接、运行和调试等操作于一体。

在 Visual C++ 6.0 开发环境下,C++ 程序按工程(project)进行组织,每个工程可包括一个或多个 CPP 源文件,但只能有一个 main 函数。

第二节 章节测试题

一、选择题

- (1) 一个完整的可运行的 C++ 源程序中()。
- A. 可以有一个或多个主函数 B. 必须有且仅有一个主函数
- C. 可以没有主函数 D. 必须有主函数和其他函数
- (2) 构成 C++ 语言源程序的基本单位是()。
- A. 子程序 B. 过程 C. 文本 D. 函数
- (3) 某 C++ 程序由一个主函数 main() 和一个自定义函数 max() 组成,则该程序()。
- A. 总是从 max() 函数开始执行 B. 写在前面的函数先开始执行
- C. 写在后面的函数先开始执行 D. 总是从 main() 函数开始执行
- (4) Visual C++ 语言规定,一个 C++ 源程序的主函数名必须为()。
- A. program B. include C. main D. function
- (5) C++ 语言源程序文件的后缀是(),经过 Compile 后,生成文件的后缀是(),经过 Build 后,生成文件的后缀是()。
- A. .obj B. .exe C. .cpp D. .doc
- (6) Visual C++ 6.0 IDE 的编辑窗口的主要功能是(),输出窗口的主要功能是(),调试器(Debug)的主要功能是()。

- A. 建立并修改程序
B. 将 C++ 源程序编译成目标程序
C. 跟踪分析程序的执行
D. 显示编译结果信息(如语法错误等)

(7) 在 Visual C++6.0 开发环境下,C++程序按工程(project)进行组织,每个工程可包括()CPP 源文件,但只能有()main 函数。

- A. 1 个
B. 2 个
C. 3 个
D. 1 个以上(含 1 个)

二、简答题

- (1) 如何使用注释语句? 使用注释有何好处?
(2) 简述 C++程序从 .cpp 源文件到 .exe 可执行文件的生成过程?

第三节 上机实践

【上机实践要求】

- (1) 熟练使用 VC++6.0 编程环境。
(2) 掌握程序结构。
(3) 了解程序的编辑、编译、链接及运行的过程。

【上机实践内容】

以下程序实现了在屏幕上输出“江苏科技大学欢迎您!”,请用 VC++6.0 完成编译调试。

```
#include<iostream.h>
void main()
{
    cout<<"江苏科技大学欢迎您!"<<endl;
}
```

第二章 数据类型与表达式

第一节 标识符及基本数据类型

【知识归纳】

知识点	主要知识要素	知识水平	学习要求
标识符	字符集		理解
	关键字	重点	熟练掌握
	自定义标识符	重点	熟练掌握
基本数据类型	基本数据类型简介		理解
	常量	难点	熟练掌握
	变量	难点	熟练掌握
	数据的输入/输出	重点	熟练掌握

【主要知识点】

一、字符集

C++中的字符集由以下字符组成。

- (1) 大小写英文字母: a~z 和 A~Z。注意: C++中严格区分英文的大小写。
- (2) 数字: 0~9。
- (3) 下划线_。
- (4) 其他符号: ~、!、#、%、*、(、)、-、+、=、|、\、{、}、[、]、:、;、"、'、<、>、.、?、/、空格、制表符、换行符等。

二、关键字

关键字指由系统事先定义好的,有特定含义与用途的词汇,也称为保留字,在程序中不能另作他用。如 char, void, main 等。

三、用户自定义标识符

用户自定义标识符指程序员根据具体编程需要,自行定义的变量名、函数名、数组名等字符序列。用户定义标识符的命名必须符合以下三个规则:

- (1) 标识符只能由字母、数字和下划线 3 种字符组成;
- (2) 标识符必须由字母或下划线开头,不能以数字开头;

(3) 标识符不能是 C++ 语言中的关键字。

四、基本数据类型简介

C++ 语言数据类型可以分为基本数据类型和构造数据类型两种。

(1) 基本数据类型指 C++ 语言中已经定义好的,不可再进一步分割的数据类型。主要有 void, bool, char, int, float, double 等,为了更精确表示各种情况下的数据,基本类型数据还可以使用一些修饰符,如 long, short, signed 和 unsigned。另外,signed 和 unsigned 也可以修饰字符型数据。不同数据类型的数据所占内存空间的大小不同,所能表示的数值的取值范围也不相同。表 2.1 中长度一栏为 32 位机上所有基本类型数据所占的内存空间的大小。

表 2.1 C++ 基本类型说明

类型	名称	长度(字节)	取值范围
char	字符型	1	-128~127
signed char	有符号字符型	1	-128~127
unsigned char	无符号字符型	1	0~255
short int	短整型	2	-32768~32767
int	整型	4	$-2^{31} \sim (2^{31}-1)$
long int	长整型	4	$-2^{31} \sim (2^{31}-1)$
unsigned short int	无符号短整型	4	0~65535
unsigned long int	无符号长整型	4	$0 \sim (2^{32}-1)$
float	单精度	4	$-10^{38} \sim 10^{38}$
double	双精度	8	$-10^{308} \sim 10^{308}$
void	空类型	0	无值

(2) 构造数据类型指由一种或几种数据类型组合而成的数据类型,以后章节中分别介绍。

五、常量

常量是指在程序运行过程中其值不会发生改变的量。常量可以分为整型常量、实型常量、字符型常量、字符串常量和符号常量等。

(1) 整型常量

① 根据数值的表示范围将整型常量定义为基本整型、短整型和长整型(在常量后加上 L 或 l),以及无符号整型(在常量后加 U 或 u)和有符号整型(signed)。

② 有三种表示方法:十进制(有正负之分)、八进制(以数字 0 开头,后面接 0~7 的数字)、十六进制(以 0x 或 0X 开头,后面接 0~9 的数字和 A~F 的字符)。如:50L、50U、50LU、50UL、0x2ED、0XAC、050 等都是正确的整型常量,而 068、xED、XA 都是不正确的整型常量。

(2) 实型常量

有单精度(float)和双精度(double)之分。有两种表示方法:

① 基本方法(十进制小数):如 2.5、-3.0、0.8、.9 等。

② 指数表示法:1.25 $\times 10^{-9}$ 表示为 1.25E-9 或 1.25e-9,在 E 或 e 的前面必须有数字,且其后面必须为整数,如.9e2、12.34E-5 等,而 e2、2e2.0 是错误的。

注意:表示实数时,整数和实数的运算规则有区别,如 10.0/4 为 2.5,而 10/4 为 2。

(3) 字符常量

① 基本表示方法:用一对单引号括起来单个西文字符,格式为‘单个西文字符’,如‘a’、‘+’、‘9’。

② 转义字符表示法:有一些起控制作用又不能直接显示的特殊字符,如换行、退格等字符就用转义字符表示,即将一个反斜杠“\”加一个控制字符或加一个 ASCII 码值用一对单引号括起来表示一个特殊字符。其中,ASCII 码值必须是八进制或十六进制整数,其取值范围(转换成十进制)在 0~255 之间。

a. 格式 1:‘\特殊控制字符’,如‘\n’、‘\t’等。

b. 格式 2:‘\ddd’,为用八进制所表示的 ASCII 码值对应的字符,d 的取值范围为 0~7。如‘\46’、‘\064’等,而‘\048’不正确。

c. 格式 3:‘\xdd’,为用十六进制所表示的 ASCII 码值对应的字符,d 的取值范围为 0~9,A~F。如‘\x4C’等。

例如:‘A’、‘\101’都是合法的字符,都表示字母‘A’。但用双引号括起来的“A”不是字符,而是字符串。常用转义字符详见表 2.2。

表 2.2 常用转义字符表

转义字符	所表示的含义
‘\n’	表示换行,鼠标光标移动到下一行的行首
‘\t’	表示水平制表符(Tab 键)鼠标光标向后移动 8 个空格
‘\a’	表示响铃
‘\b’	表示退格,鼠标光标向左移动一个空格
‘\’	表示输出一个单引号
‘\”	表示输出一个双引号
‘\0’	表示字符串的结束标志,其 ASCII 码值为 0
‘\’	表示输出一个反斜杠
‘\ddd’	用 1~3 为八进制表示的字符,d 为 0~7 之间的数字
‘\xdd’	用 1~2 为十六进制表示的字符,d 为 0~9 之间的数字或 A~F 之间的字母

(4) 字符串常量

① 表示字符串常量的方法:将一个或多个字符用双引号括起来表示字符串常量,其格式:“一串字符”,如“a”、“中”、“123”、“xyz\nabc”。注意:转义字符在字符串常量中同样有效。

② 表示字符串常量时注意:

a. 字符串总隐含一个结束标记‘\0’,如“a”占 2 个字节,“abc\0”占 5 个字节。

b. 字符串是若干个字符的组合,其中大多数字符既可以用基本方法表示,也可以用

转义字符表示,如“abc”和“a\142c”是两个一样的字符串;而有些字符只能用转义字符表示,如“123‘abc’456”,“123\xyz”等是错误的。注意,要输出的双引号必须用转义字符表示。

(5) 符号常量

① 用 const 定义符号常量,其格式为:const 数据类型 符号常量名=值常量;

其中,const 是关键字,数据类型用来表示要定义的符号常量的类型,符号常量名的定义要符合标识符命名规则,值常量后面分号不能少。

如:const int x=5;

const float y=5;

② 用预处理命令#define 定义符号常量,其格式为:#define 符号常量 值常量

其中,以#开头的命令是一个预处理命令,不允许带数据类型,值常量后面不带分号。

如:#define R 5

六、变量

变量是指在程序中其值可以改变的量,实质是在内存中分配一块存储空间,用以存储或读取数据。

(1) 变量定义,其格式:数据类型 变量 1,变量 2,...,变量 n;

其中,数据类型指明变量的数据类型,变量名的定义要符合标识符的命名规则。多个变量同时定义时用逗号分隔,最后以分号结束。

例如:int a,b;

double x,y,z;

(2) 变量初始化,其格式:数据类型 变量名=数据;

或 数据类型 变量名(数据);

其中,用来初始化的常量应与所定义的变量类型一致。

例如:int a=10,b(20);

七、指针变量

指针变量定义格式:

数据类型 * 变量名;

其中,* 表示定义的变量为指针类型。如:

int * p1; //表示定义一个整型的指针变量 p1

关于指针变量的使用需要注意以下几点:

(1) 取变量地址 &,用来求一个变量的存储的起始地址,其格式为:

& 变量名

例如:

int a=10,* p=&a;

(2) 取内容 *,用于求指针所指向的内存单元的值。

例如:

int a=10,* p=&a;

执行 cout<<* p<<endl; // 输出结果为 10

(3) 赋值运算,是指对指针变量初始化,或改变指针所指的位置,其格式为:

数据类型 * 变量名=变量的地址;

例如:

```
int a=4, * p3=&a;
```

```
int * p2=p3, * p4;p4=p3;
```

(4) 算术运算(+、-):

① 指针变量加(减)一个整数(n)表示指针从当前位置向后(前)移动 n 个存储单元。

② 指针变量相减表示两指针相差多少个存储单元。

③ 指针变量不允许相加,如 p1+p2 是错误的。

(5) 关系运算,用于比较指针所指位置的前后关系。有大于,大于等于,小于,小于等于,等于和不等于共 6 种关系。

(6) 指针变量的运算是改变指针所指的位置,而对指针变量所指值的运算是改变指针所指内存中的内容。

例如:

```
char * p="abcdefgh";
```

```
p+=2;           // p 原来指向字符'a',运算后 p 指向字符'c'
```

```
* p+=2;         // 把 p 所指的字符'c',修改为字符'e'
```

八、引用变量

实质是为一个已经定义好的变量起一个别名,系统并不为该引用类型变量重新分配内存空间。其定义格式为:

数据类型 & 引用变量名=已定义的变量名;

其中,对引用变量赋值时必须为已定义好的变量,且其数据类型必须一致。

例如:

```
double x=5.6;
```

```
double &y=x;
```

```
cout<<y<<endl;    // 输出 5.6
```

九、数据的输入和输出

(1) cin 输入流

cin 与输入运算符(提取运算符)>>连用,后面只能跟变量名、字符数组和指向确定位置的字符型指针,不能接任何格式符,以回车作为结束。当输入多个变量时,中间以空格或回车分开。其格式为:

```
cin>>变量 1>>变量 2>>.....>>变量 n;
```

注意:

① 在默认情况下输入的数据为十进制,要以十六进制或八进制方式输入数据(只对整型变量),必须在 cin 输入流中指定相应的数制,其格式为:

```
cin>>hex>>变量 1>>变量 2>>.....    // 输入十六进制数
```

```
cin>>oct>>变量 1>>变量 2>>.....    // 输入八进制数
```

```
cin>>dec>>变量 1>>变量 2>>.....    // 输入十进制数
```


注意:当指明使用某种数制后,将一直保持以这种进制进行输入,直到再次指明另一种数制输入为止。

例如:

```
int a, b, c, d;
cin>>a>>b;           // 缺省输入两个十进制整数
cin>>hex>>c>>d;       // 以十六进制方式输入直至重新指定输入方式
cin>>oct>>a>>b;       // 以八进制重新输入 a, b 的值
cin>>c;                // 仍然是八进制输入方式
cin>>dec>>d;          // 回到缺省的十进制输入方式
```

② 输入字符型变量时,各数据可以以空格分开,也可以不分开而连续输入,如:

```
char c1, c2, c3;
cin>>c1>>c2>>c3;
```

执行时输入 a b c<回车>和 abc<回车>以及 a<回车>b<回车>c<回车>效果是一样的。cin 输入流自动跳过输入的空格和回车符。如果字符型变量必须接受一个空格字符,则应使用函数 cin.get(字符型变量),该函数从输入行中取出一个字符赋给字符型变量,但一次只能接受一个字符,可以是空格或回车符。

例如:

```
char c1, c2, c3;
cin.get(c1);
cin.get(c2);
cin.get(c3);
```

执行时输入 a b c 和 abc 是不一样的。前者结果为 c1,c2,c3 的值分别为‘a’,空格,‘b’,后者 c1,c2,c3 的值分别为‘a’,‘b’,‘c’。

(2) cout 输出流

cout 与输出运算符(插入运算符)<<连用,后面可以跟变量、常量、表达式(含有返回值的函数调用表达式)等,也可以加控制输出格式‘\n’(换行)‘\t’(制表符)或 endl(换行)等,其格式为:

```
cout<<变量 1<<变量 2<<变量 3<<……<<变量 n<<endl;
```

例如:cout<<a<<'\t'<<b<<'\n'; 输出字符‘\t’是为了将输出数据分隔开来。

注意:

① 可以用 setw()函数来指定数值的输出宽度,它只对紧跟其后的输出有效。使用该函数时程序中必须包含头文件 iomanip.h,即#include <iomanip.h>。

例如:cout<<setw(8)<<a<<setw(8)<<b<<'\n'; 执行后无论 a 和 b 的值是几位数,其输出均占 8 个字符的宽度,其不足 8 位部分前面以空格补足 8 位。

② 输出字符或字符串的方法与输出数值基本相同。

```
int c1='a';cout<<c1<<'a'<<'\n'; // 输出字符变量 c1 的值和直接输出字符 a 并换行
cout<<"VC++Program!"<<'\n'; // 输出一个字符串并换行
```

③ 与输入数据类似,要以十六进制和八进制输出整数,则需指定其输出进制。系统默认的输出方式为十进制。