21世纪 高等学校计算机语言课 进阶辅导

权威推荐

C++程序设计

↘知识点全面覆盖

↘国内外精典实例剖析

↘典型习题拓展训练

《进阶辅导》编委会 组编

超值享受

- 全国计算机等级考试上机考试练习系统
  - 模拟考试，完全贴近真实上机考试环境
  - 仿真练习，全面熟悉考试真题
- 全国计算机等级考试笔试练习系统
  - 模拟考试，充分体验考试氛围
  - 仿真练习，通过练习历年真题掌握考试内容
- 书中涉及的源代码

大连理工大学出版社
大连理工大学电子音像出版社

21世纪高等学校计算机语言课**进阶辅导**

C++程序设计

组编 《进阶辅导》编委会

编著 黄 明 梁 旭 王民生

梁 霞 李正光 李 昕

大连理工大学出版社

大连理工大学电子音像出版社

《进阶辅导》编委会

（按笔画顺序排列）

王民生　任建娅　刘玉霞　刘晓红　李正光
李　昕　汪　静　谷晓琳　郎大维　金　花
闻丽华　梁　旭　梁　霞　黄　明　程智勇

C++程序设计

《进阶辅导》编委会　组编

大连理工大学出版社
大连理工大学电子音像出版社　出版

地址：大连市凌水河　邮政编码：116024

电话：0411-84708842　传真：0411-84701466　邮购：0411-84707961

E-mail：dzcb@dutp.cn　URL：http://www.dutp.cn

大连业发印刷有限公司印制　　大连理工大学出版社发行

幅面尺寸：185mm×260mm　　印张：13　　字数：260千字

2005年7月第1版　　2005年7月第1次印刷

责任编辑：王影琢　高智银　　责任校对：达　理

封面设计：宋　蕾

ISBN 7-900670-46-7　　定价：21.80元

前　言

在多年的教学实践中，我们总是听到学生反映学完“计算机语言课”后，实际编程还有很大的困难。我们认为，主要的问题就在于课内教材偏重于语句、语法的教学，实际应用方法的介绍、应用技能的培养非常有限，而练习题较简单，无法满足学生提高技能并达到实际应用的需求。为此，我们编写了这套《21世纪高等学校计算机语言课进阶辅导》丛书。

1.内容结构

共分为两个部分：

第一部分是程序设计学习指导。按内容分章节讲述，每章中都包含知识点归纳、例题精讲、拓展训练及其参考答案。概念清晰、讲解透彻、重点突出、适用性强，知识点全面覆盖。

第二部分是程序设计上机指导。包括编译环境介绍、编程技巧、应用实例、经典游戏等内容。

2.特点

(1)以内容难度适中、讲解深入浅出为基础，以切实帮助学生提高能力为出发点和根本目的。

(2)参加编写的人员都是多年从事计算机语言课教学、毕业设计指导、计算机语言培训的教师，他们既有丰富的实际开发经验，又真切地了解大多数学生在日常学习中的不足，所以本套丛书可以帮助学生解决学习中遇到的普遍性问题。

(3)在精选国内外实例的同时，对每个实例的解题思路均进行详细分析，对程序中的重点语句也通过注释的形式加以特殊强调。本套丛书力争做到指导性与可操作性相结合。

(4)书中的程序通过调试，可以直接在实际项目中使用。

3.适用范围

适用范围广泛,既可以作为计算机语言课的课程设计、毕业设计辅导用书,又可以作为学习软件开发的辅助用书,还可以作为参加计算机等级考试、软件水平考试的参考用书,同时还可以作为学员培训、程序员入门用书。

4.配套光盘

与其配套的光盘中附有书中涉及的源代码和资源文件,可供读者调试、运行。另外,还为读者提供了最新的“全国计算机等级考试练习系统”,全真的模拟环境、历年的真题练习可以满足广大全国计算机等级考试考生的要求。

本套丛书由《进阶辅导》编委会组织编写。由于水平有限,书中错误和不妥之处在所难免,请读者和专家批评指正。

读者在使用过程中如有问题,可与编者联系:dlhm@263.net。

编　者

2005年7月

目　录

Note:

第一部分 C++语言程序设计学习指导

第1章 C++基础知识及其基本控制结构程序设计

1.1 知识点归纳

一、C++程序的结构和书写格式

1.一个C++程序可以由一个程序单位或多个程序单位构成。每一个程序单位作为一个文件。在程序编译时,编译系统分别对各个文件进行编译,因此,一个文件是一个编译单位。

2.在一个程序单位中,可以包括以下几个部分:

- 预处理命令
- 全局声明部分
- 函数

但是并不要求每一个程序文件都必须具有以上3个部分,可以缺少某些部分(包括函数),完全根据需要而定。

3.一个函数由两部分组成:

- 函数首部
- 函数体

其中函数体又包括局部声明部分和执行部分。

4.语句包括两类:一类是声明语句,它是非执行语句;另一类是执行语句,用于实现用户指定的操作。

5.一个C++程序总是从main函数开始执行的,而不论main函数在整个程序中的位置如何(main函数可以放在程序文件的最前头,也可以放在程序文件的最后,或在一些函数之前,在另一些函数之后)。

Note:

6.类(class)是C++新增加的重要的数据类型,是C++对C最重要的发展。类实现了面向对象程序设计方法中的封装、信息隐蔽、继承、派生、多态等功能。

7.C++程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。

8.C++提供了一种新的注释方式:从"//"开始,直到行尾,都将被计算机当做注释。用"//"作注释时,有效范围只有一行,即本行有效,不能跨行。若需要多行注释,可以在每行用一个"//"开头,或用"/*……*/"作注释,它的有效范围为多行。

二、C++的词法单位

1.C++的字符集

字符集是构成C++语言的基本元素。用C++语言编写程序时,除字符型数据外,其他所有成分都只能由字符集中的字符构成。C++语言的字符集由下列字符构成:

- 英文字母:A~Z,a~z
- 数字字符:0~9
- 特殊字符:空格 ! # % ^ & * _ (下划线)
  + = : - ~ < > / \ ?
  ' " ; . , ( ) [] {} |

2.关键字

关键字(Keyword)又称保留字,是由系统定义的具有特殊含义的英文单词,关键字不能另作他用。C++区分字母的大小写,关键字全由小写字母组成。标准C++(ISO14882标准)定义了74个关键字,表1-1为常用关键字及分类。

表1-1 C++常用关键字及分类

数据类型说明与修饰符	bool char class const double enum float int long short signed struct union unsigned void volatile wchar_t
存储类型说明符	auto extern inline register static
访问说明符	friend private protected public
语句	break case catch continue default do else for goto if return switch throw try while
运算符及逻辑值	delete false new sizeof true
其他说明符	asm explicit namespace operator template this typedef typename using virtual

3.标识符

标识符是用来标识变量、符号常量、函数、数组、类型等实体名字的有效字符序列,简单地说,标识符就是一个名字。合法标识符由字母或下划线开头,由字母、数字和下划线组成,其有效长度为1~31个字符,长度超过31个字符的只识别前31个字符,C++标识符长度为1~247个字符。C++区分大小写字母,例如name和Name是不同的标识符。用户自定义标识符时不能使

用关键字,也不可以与 C++ 编译器提供的资源如函数库名、类名等同名。

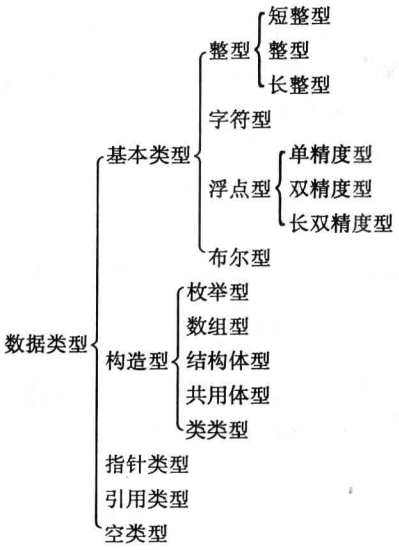
Note:

三、数据类型和表达式

1.数据类型

C++ 将数据分为若干类型,程序中使用到的所有数据都必须指明其类型。定义数据类型实际上给出了两方面信息:一是该类型数据在内存中如何存储;二是该类型数据能进行哪些合法运算。C++ 中的数据类型分为两大类:基本数据类型和非基本数据类型。基本数据类型有 bool(布尔型)、char(字符型)、int(整型)、float(浮点型)和 double(双精度型)。非基本数据类型是用户根据程序需要,按 C++ 语法规则参与构造出来的数据类型。

C++ 可以使用的数据类型如下:



2.常量

所谓常量是指在程序运行的整个过程中其值始终不可改变的量,常量包括两大类,即数值型常量和字符型常量。

(1)数值常量

数值常量就是通常所说的常数。在 C++ 中,数值常量是区分类型的,从字面形式即可识别其类型。

①整型常量

整型常量就是整数,包括正整数、负整数和零。整型常量的表示形式有十进制、八进制和十六进制。

◆ 十进制整型常量的一般形式与数学中我们所熟悉的表示形式是一样的:

[±]若干个 0~9 的数字

即符号加若干个 0~9 的数字,但数字部分不能以 0 开头,整数前边的正号可以省略。

Note:

◆ 八进制整型常量的数字部分要以数字 0 开头，一般形式为：

[±]0 若干个 0～7 的数字

◆ 十六进制整型常量的数字部分要以数字 0x 开头，一般形式为：

[±]0x 若干个 0～9 的数字及 A～F 的字母(大小写均可)

整型常量可以用后缀字母 L(或 l)表示长整型，后缀字母 U(或 u)表示无符号型，也可以同时后缀 L 和 U(大小写无关)。

②实型常量

C++ 中包含小数点或 10 的幂的数为实型常量。实型常量有一般形式和指数形式。

一般形式：例如，5.6，-45.1 等。

指数形式：例如，1.23E+3 表示 1.23×10^3，-6.18E-2 表示 -6.18×10^{-2}，其中，字母 E 可以大写或小写。当以指数形式表示一个实数时，整数部分和小数部分可以省略其一，但不能都省略。例如，.35E-2，45.E3 都是正确的，但不能写成 E-2 这种形式。

实型常量默认为 double 型，如果后缀 F(或 f)，则为 float 型。

(2)字符常量

①普通字符常量

用单撇号括起来的一个字符就是字符常量。如'a'，'%'都是合法的字符常量，在内存中占一个字节。注意：

◆ 字符常量只能包括一个字符，如'ab'是不合法的。

◆ 字符常量区分大小写字母，如'A'和'a'是两个不同的字符常量。

◆ 撇号(')是定界符，而不属于字符常量的一部分。

②转义字符常量

除了以上形式的字符常量外，C++ 还允许用一种特殊形式的字符常量，就是以“\”开头的字符序列。例如，'\n'代表一个“换行”符。常用的以“\”开头的特殊字符见表 1-2。

表 1-2　　转义字符及其含义

字符形式	含　义	ASCII 代码
\a	响铃	7
\n	换行，将当前位置移到下一行开头	10
\t	水平制表(跳到下一个 tab 位置)	9
\b	退格，将当前位置移到前一列	8
\r	回车，将当前位置移到本行开头	13
\f	换页，将当前位置移到下页开头	12
\v	竖向跳格	8
\\	反斜杠字符	92
\'	单引号(撇号)字符	39
\"	双引号字符	34
\0	空字符	0
\ddd	1～3 位八进制数所代表的字符	
\xhh	1～2 位十六进制数所代表的字符	

Note:

转义字符虽然包含两个或多个字符，但它只代表一个字符。编译系统在见到字符“\”时，会接着找它后面的字符，把它处理成一个字符，在内存中只占一个字节。将一个字符常量存放到内存单元时，实际上并不是把该字符本身放到内存单元中去，而是将该字符相应的 ASCII 代码放到存储单元中。

③字符串常量

用双撇号括起来的部分就是字符串常量，如"abcd","Hai!","mm + nn"都是字符串常量。编译系统会在字符串最后自动加一个'\0'作为字符串结束标志。但'\0'并不是字符串的一部分，它只是作为字符串的结束标志。因此，字符串常量"abcd"在内存中占 5 个字节。

3.变量

在程序的执行过程中其值可以变化的量称为变量。变量必须用标识符进行标识，称为变量名。变量有类型之分，如整型变量、浮点型变量、字符变量等，还有各种导出类型变量，如数组变量。任何类型的变量都必须先定义后使用，定义一个变量即确定了它的 4 个属性：名字、数据类型、允许的取值及合法的操作。C++ 要求对变量做强制定义的目的是：

◆ 凡未被事先定义的，不作为变量名，这就保证了程序中变量名使用的正确性；

◆ 便于编译程序为变量预先分配存储空间；

◆ 便于在编译期间进行语法检查。

(1)变量定义

在 C++ 中，变量定义的一般格式为：

存储类型 数据类型 变量名 1,变量名 2,……,变量名 n;

变量的存储类型包括：

◆ auto：采用堆栈方式分配内存空间，属于暂时性存储，其存储空间可以被若干变量多次覆盖使用。

◆ register：存放在通用寄存器中。

◆ extern：在所有函数和程序段中都可以使用。

◆ static：在内存中是以固定地址存放的，在整个程序运行期间都有效。

数据类型可以是 C++ 定义的基本数据类型，还可以是非基本数据类型。

变量定义可以出现在程序的任何位置，但必须在使用变量之前。同一变量只能做一次说明性定义，与此相对的是变量声明，同一变量可以多次声明。

经过说明的变量可以使用，包括赋值和引用。引用变量时，必须保证变量有确定的值。变量经过说明而未被赋值时，其取值为不确定的随机值，不能被引用；否则，会产生编译错误。

(2)变量赋初值

首次引用变量时，变量必须有惟一确定的值，这个取值称为变量的初值。给变量赋初值称为初始化，初始化有两种方法：

①变量说明时直接赋初值。例如：

Note:

```
int a=8,b=9,c=19;
char m='c',n='1';
```

②用赋值语句赋初值。例如：

```
int x,y;
x=3;
y=3839;
```

(3)常变量

在定义变量时，如果加上关键字 const，则变量的值在程序运行期间不能改变，这种变量成为常变量。例如：

```
const int x=5;
```

在定义常变量时必须同时对它初始化(即指定其值)，此后它的值不能再改变。常变量不能出现在赋值号的左边。

4.运算符、表达式和语句

(1)算术运算符与算术表达式

①基本的算术运算符、优先级与结合性(见表 1-3)

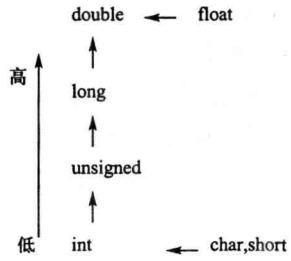
表 1-3　　算术运算符及优先级

优先级	运算符	名　称
优先级自上向下的顺序为从高到低	+	正，单目
	-	负，单目
	*	乘，双目
	/	除，双目
	%	求余，双目
	+	加，双目
	-	减，双目

C++规定算术运算符的结合方向为“自左至右”，即先左后右。“自左至右的结合方向”又称“左结合性”，即运算对象先与左面的运算符结合。同理结合方向为“自右至左”称为右结合性。

②表达式中的各类数值型数据间的混合运算

在表达式中常遇到不同类型数据之间进行运算，在进行运算时，不同类型的数据要先转换成同一类型，然后进行运算。转换的规则如下所示。



③自增、自减运算符

在 C++中，常在表达式中使用自增(++)和自减(--)运算符，它们的作用是使变量的值增 1 或减 1，例如：

Note:

++i（在使用 i 之前，先使 i 的值加 1）

--i（在使用 i 之前，先使 i 的值减 1）

i++（在使用 i 之后，使 i 的值加 1）

i--（在使用 i 之后，使 i 的值减 1）

正确地使用 ++ 和 --，可以使程序简洁、清晰、高效。但应注意：

◆ 自增运算符和自减运算符只能用于变量，而不能用于常量或表达式。

◆ ++ 和 -- 的结合方向是“自右至左”。

◆ 自增运算符在 C++ 程序中是经常见到的，常用于循环语句中，使循环变量自动加 1；也用于指针变量，使指针指向下一个地址。

◆ 自增运算符和自减运算符使用十分灵活，但在很多情况下可能出现歧义性，应该尽量避免出现歧义性。

(2)赋值运算符与赋值表达式

①赋值运算符

赋值符号“=”就是赋值运算符，它的作用是将一个数据赋给一个变量。

②赋值过程中的类型转换

若赋值运算符两侧的类型不一致，但都是数值型或字符型时，在赋值时会自动进行类型转换。

◆ 将浮点型数据(包括单、双精度)赋给整型变量时，舍弃其小数部分；

◆ 将整型数据赋给浮点型变量时，数值不变，但以指数形式存储到变量中；

◆ 将一个 double 型数据赋给 float 型变量时，要注意数值范围不能溢出；

◆ 字符型数据赋给整型变量，将字符的 ASCII 代码赋给整型变量；

◆ 将一个 int、short 或 long 型数据赋给一个 char 型变量，只将其低 8 位原封不动地送到 char 型变量(发生截断)；

◆ 将 signed(有符号)型数据赋给长度相同的 unsigned(无符号)型变量，将存储单元内容原样照搬(连原有的符号位也作为数值一起传送)。

③复合的赋值运算符

在赋值符是“=”之前加上其他运算符，可以构成复合的运算符。C++ 可以使用以下几种复合赋值运算符：

+=，-=，*=，/=，%=，<<=，>>=，&=，^=，|=

④赋值表达式

由赋值运算符将一个变量和一个表达式连接起来的式子称为“赋值表达式”。它的一般形式为：

<变量> <赋值运算符> <表达式>

对赋值表达式求解的过程是：先求“赋值运算符”右侧的“表达式”的值，然后赋给“赋值运算符”左侧的“变量”。

Note:

(3)逗号运算符与逗号表达式

C++提供一种特殊的运算符——逗号运算符。用它将两个表达式连接起来。逗号表达式的一般形式为:

表达式1,表达式2,表达式3,……,表达式n

逗号运算符是所有运算符中级别最低的。

(4)关系运算符、逻辑运算符及其表达式

在解决许多问题时都需要进行情况判断,对复杂的条件进行逻辑分析。C++中提供了用于比较、判断的关系运算符和用于逻辑分析的逻辑运算符。

关系运算是比较简单的一种逻辑运算,关系运算符及优先级次序为:

优先级相同(较高)<、<=、>、>=;优先级相同(较低)==、!=

用关系运算符将两个表达式连接起来,就是关系表达式。关系表达式是一种最简单的逻辑表达式,其结果类型为bool,值只能是true或false。

用逻辑运算符将简单的关系表达式连接起来,构成较复杂的逻辑表达式。逻辑表达式的结果类型为bool,值只能是true或false。C++中的逻辑运算符及其优先次序为:

!(非) &&(与) ||(或)

优先级次序: 高 → 低

"!"是一元运算符,使用形式是:!操作数。非运算的作用是对操作数取反。"&&"和"||"都是二元运算符。"&&"运算的作用是求两个操作数的逻辑与,只有当两个操作数的值都为true时,"与"运算结果才为true,其他情况下与运算结果均为false。"||"运算的作用是求两个操作数的逻辑或,只有当两个操作数的值都为false时,"或"运算结果才为false,其他情况下"或"运算结果均为true。

(5)条件运算符与条件表达式

C++中惟一的一个三元运算符是条件运算符"?:",它能够实现简单的选择功能。条件表达式的形式是:

表达式1? 表达式2:表达式3

其中"表达式1"必须是bool型,"表达式2"、"表达式3"可以是任何类型,且类型可以不同。条件表达式的执行顺序是:先求解"表达式1",若"表达式1"的值为true,则求解"表达式2","表达式2"的值为最终结果;若"表达式1"的值为false,则求解"表达式3","表达式3"的值为最终结果。条件运算符的优先级高于赋值运算符,低于逻辑运算符;结合方向为自右至左。

(6)sizeof运算符

sizeof运算符用于计算某种类型的对象在内存中所占的字节数。该运算符使用的语法形式为:

sizeof(类型名)或sizeof(表达式)

(7)强制类型转换运算符

强制类型转换的作用是将表达式的结果类型转换为类型说明符所指定的

Note:

类型。强制类型转换是通过类型说明符和括号来实现的，其语法形式为：

类型说明符(表达式)或(类型说明符)表达式

使用强制类型转换时，应该注意：

◆ 强制类型转换不改变表达式本身的值类型，而是产生一个临时变量，用来暂存转换后的值，该临时变量被引用后即自动释放；

◆ 强制类型转换运算符优先级较高，只对紧随其后的表达式起作用，而对其他部分不起作用；

◆ 强制类型转换应当用在不做转换将影响表达式结果的正确性或精度，或不能完成相应运算的场合。

(8)语句

语句是程序的基本单位。C++ 中的语句分为以下几种：

①表达式语句

表达式语句是最简单的语句形式，一个表达式后面加上一个分号就构成了表达式语句，一般格式为：

表达式;

②空语句

只由一个分号构成的语句称为空语句。空语句不执行任何操作，但具有语法作用。

③复合语句

由一对“{}”括起来的一组语句构成一个复合语句。复合语句描述一个块，在语法上起一个语句的作用。

对单个语句，必须以“;”结束，对复合语句，其中的每个语句仍以“;”结束，而整个复合语句的结束符为“}”。

④流程控制语句

流程控制语句用来控制或改变程序的执行方向。

四、简单的 C++ 输入/输出

1.输入/输出

程序执行期间，从外设接收信息的操作称为输入，向外设发送信息的操作称为输出。C++ 没有专门的输入/输出语句，而是通过系统提供的输入/输出流类来实现。应用 cin 和 cout 能够实现最基本的数据输入/输出，包括整数、实数、字符和字符串。

cin 用来在程序执行期间给变量输入数据，一般格式为：

cin>>变量1>>变量2>>……>>变量n;

其中“>>”称为提取运算符，它将暂停程序执行，等待从键盘上输入相应数据，直到所列出的所有变量均获得值后，程序才继续执行。用 cin 可以只给一个变量输入，也可以一次给多个变量输入。

cout 实现将数据输出到显示器的操作，一般格式为：

cout<<表达式1<<表达式2<<……<<表达式n;

Note:

其中“<<”称为插入运算符，它将紧跟其后的表达式的值输出到显示器当前光标位置。

由于流对象 cin、cout 和流运算符的定义等信息是存放在 C++ 的输入/输出流库中的，因此如果在程序中使用 cin、cout 和流运算符，就必须使用预处理命令把头文件 iostream 包含到本文件中：

```
#include<iostream>
```

2.输入/输出的格式控制

(1)不同进制数据的输入/输出

缺省状态下，输入/输出的数据是十进制的，如果要求按八进制或十六进制输入/输出，在 cin 或 cout 中必须指明相应的数据形式，oct 为八进制，hex 为十六进制，dec 为十进制。关于输入/输出非十进制数，还需要说明以下几点：

◆ 由于已经在 cin 中指明数制，因此从键盘输入时，八进制和十六进制数可以省略其开头的 0 和 0x 标志；

◆ 只适用于整型变量，不适用于实型和字符型变量；

◆ 在 cin 或 cout 中指明数制后，该数制将一直有效，直到重新指明用其他数制。

(2)设置数据间隔

缺省状态下数据是无间隔输出的。为了使数据间隔开，除了用输出空格和回车换行的方法外，还可以用 C++ 提供的函数 setw()。setw()可以指定输出数据项的宽度。setw()括号中可以是表达式，但取值必须是正整数。该设置仅对紧跟其后的一个数据项起作用，指明该数据项的输出宽度为其括号中的数值。

(3)使用新的换行符

在语句 cout<<"Hello,"<<name<<"!"<<endl;中，C++ 使用 endl 换行。这个换行符号可以用在其他位置。例如：

```
cout<<endl<<"How are you?"<<endl<<endl;
```

上述语句在“How are you?”的前后各产生一个空行。

五、算法的基本控制结构

算法的基本控制结构有三种：顺序结构、选择结构和循环结构。

1.选择结构和 if 语句

if 语句称为分支语句或条件语句，其功能是根据给定的条件是否满足来选择程序的执行方向。

(1)if 语句的 3 种形式

①if(表达式) 语句

②if(表达式) 语句 1 else 语句 2

③if(表达式 1) 语句 1
else if(表达式 2) 语句 2