



高职高专课程与教学改革系列教材

GAOZHI GAOZHUAN KECHENG YU JIAOXUE GAIGE XILIE JIAOCAI

"十一五"规划教材

C++程序设计基础导学

赵海廷 严运国 主 编



科学出版社

www.sciencep.com

· 21 世纪高职高专课程与教学改革系列教材 ·

“十一五”规划教材

C++程序设计基础导学

赵海廷 严运国 主编

科学出版社

北 京

内 容 简 介

本书为《C++程序设计基础》的配套教材,是实践训练环节必不可少的辅助教材。本书按照《C++程序设计基础》的顺序组织编撰,由基本要求、内容小结、典型例题分析、习题、实验及习题参考答案组成。本书的目的是通过典型例题剖析、习题和实验环节来巩固所学的理论知识,提高学生的实践操作水平。

本书适用高职高专的理工科学生,也适合于C++语言的初学者,还可供普通高校理工科学生、成人高校理工科学生、计算机等级考试考生、计算机培训班学员和C++语言自学者参考和使用。

图书在版编目(CIP)数据

C++程序设计基础导学/赵海廷,严运国主编. - 北京:科学出版社,2006
(21世纪高职高专课程与教学改革系列教材)
ISBN 7-03-016739-2

I . C… II . ①赵…②严… III . C语言 - 程序设计 - 高等学校:技术学校 - 教材 IV . TP312

中国版本图书馆CIP数据核字(2005)第160065号

责任编辑:王雨舸/责任校对:董艳辉

责任印制:高 嵘/封面设计:李梦佳

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

武汉大学出版社印刷总厂印刷

科学出版社发行 各地新华书店经销

*

2006年1月第 一 版 开本:787×1092 1/16

2006年1月第一次印刷 印张:12 1/4

印数:1~5 000 字数:300 000

定价:18.50 元

(如有印装质量问题,我社负责调换)

前 言

目前,相当一部分高职高专院校仍然把C语言作为工科学生程序设计语言学习的必修课和选修课,然而,C语言的运行环境是DOS环境,这使得C语言的教学和上机实验显得极不协调。显然,将教学C语言改为教学C++语言是一种必然。可是,由于C++功能强大,使用灵活,再加上教学时数的限制,对于初学者来说,C++语言的学习就显得比较困难。

现在关于C++语言的书籍不少,但适合高职高专理工科学生和程序设计语言初学者的书籍则不多。我们根据多年的教学经验及目前高职高专程序设计语言教学的实际情况,对C++语言的内容进行了精选,确定了本套教材的结构。

本套教材在体系结构的确定和内容的选择上,坚持了由浅入深、深入浅出、循序渐进的方针,力求简单明了、通俗易懂,所以本套教材的可读性、实用性和先进性俱佳,既适合教学,又适合自学。

本书是《C++程序设计基础》的配套教材,主要内容结构为:基本要求,内容小结,典型例题分析,习题,实验及习题参考答案。

为了加强学生动手能力的训练,本书安排了15个实验,它们是:熟悉Visual C++ 6.0集成开发环境,数据类型、运算符及表达式,输入/输出与顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,指针与数组,多级指针与指针数组,函数及其调用,指针和函数,内联函数与带默认参数值的函数,编译预处理,结构体、联合体,类与对象,继承与虚函数。在每个实验环节中都安排了实验目的、实验内容和步骤等环节,并给出了调试通过的参考程序。本书还提供典型例题44个,实验题目及参考程序70个,其他各类习题345个并给出了参考答案,所有程序都在Visual C++ 6.0环境下调试通过。

为了配合学生参加全国计算机等级C++二级考试,本书在附录中列出:C++二级计算机等级考试大纲,C++二级计算机等级考试样题及参考答案,C++二级计算机等级考试笔试模拟试卷。

在本书的编写过程中,作者参考了参考文献中列举的书籍及其他资料,在此向这些书籍、资料的作者表示诚挚的谢意!

本书面向高职高专理工科学生,也兼顾了C++语言的初学者。本书还可供普通高校理工科学生、成人高校理工科学生、计算机等级考试考生、计算机培训班学员和C++语言自学者参考和使用。

本书由赵海廷、严运国主编,并负责全书的总体策划与统稿工作,闻朝中、周大勇任副主编。其中,第1、2章由闻朝中编写,第3、9章由赵海廷编写,第4、8章由严运国编写,第5章由周大勇编写,第6章由严运国和周大勇编写,第7章由杨晓燕编写。最后由赵海廷统稿、定稿。

由于编者的水平有限,书中难免有错漏之处,敬请各位专家和读者批评指正。

编 者

2005年10月

目 录

第一章 认识 C++	1
1.1 基本要求	1
1.2 内容小结	1
习题 1	2
实验 1.1 熟悉 Visual C++ 6.0 的集成开发环境	3
习题 1 参考答案	6
第二章 数据类型、运算符及表达式	8
2.1 基本要求	8
2.2 内容小结	8
2.3 典型例题分析	9
习题 2	12
实验 2.1 数据类型、运算符及表达式	14
习题 2 参考答案	16
第三章 结构化程序设计	19
3.1 基本要求	19
3.2 内容小结	19
3.3 典型例题分析	24
习题 3	29
实验 3.1 输入/输出与顺序结构程序设计	34
实验 3.2 选择结构程序设计	36
实验 3.3 循环结构程序设计	39
习题 3 参考答案	42
第四章 数组	46
4.1 基本要求	46
4.2 内容小结	46
4.3 典型例题分析	49
习题 4	54
实验 4.1 数组	63
习题 4 参考答案	66
第五章 指针	69
5.1 基本要求	69
5.2 内容小结	69
5.3 典型例题分析	71
习题 5	77
实验 5.1 指针与数组	83

*实验 5.2 多级指针与指针数组	85
习题 5 参考答案	86
第六章 函数	89
6.1 基本要求	89
6.2 内容小结	89
6.3 典型例题分析	90
习题 6	93
实验 6.1 函数及其调用	99
实验 6.2 指针和函数	102
*实验 6.3 内联函数与带默认参数值的函数	103
习题 6 参考答案	104
第七章 编译预处理	108
7.1 基本要求	108
7.2 内容小结	108
7.3 典型例题分析	110
习题 7	111
*实验 7.1 编译预处理	111
习题 7 参考答案	112
第八章 复杂数据类型	114
8.1 基本要求	114
8.2 内容小结	114
8.3 典型例题分析	116
习题 8	120
实验 8.1 结构体和联合体	127
习题 8 参考答案	130
第九章 类与对象	133
9.1 基本要求	133
9.2 内容小结	133
9.3 典型例题分析	139
习题 9	146
实验 9.1 类与对象	148
实验 9.2 继承与虚函数	153
习题 9 参考答案	159
附录 I C++二级计算机等级考试大纲	166
附录 II C++二级计算机等级考试样题及参考答案	169
附录 III C++二级计算机等级考试笔试模拟试卷	178
参考文献	188

第一章 认 识 C++

1.1 基 本 要 求

1. 了解程序设计语言的分类和发展过程。
2. 掌握面向过程和面向对象程序设计语言的一些基本概念。
3. 掌握 C++ 程序的基本结构。
4. 熟悉 Visual C++ 6.0 开发环境的使用。

1.2 内 容 小 结

计算机系统的硬件是由计算机的各部件和外部设备组成。计算机部件主要包括微处理器芯片、存储器芯片和输入、输出接口芯片。

计算机指令是计算机可以识别的命令。一台计算机硬件能够识别的所有指令集合称为该计算机的指令系统。为计算机运行所编制的各种程序及其相关文档资料的总和称为软件。

机器语言是用二进制指令表示的、计算机能够直接识别和执行的语言。执行速度快是机器语言的主要优点。但机器语言不易记忆，不便于学习，软件开发难度大、周期长、易出错，软件维护工作量大等是其突出缺点。

汇编语言是用一些易读、易记的助记符表示机器指令操作码，使用汇编语言编写程序提高了编程速度，也比较容易编写出质量较高的程序，检查、维护也比较容易。但汇编语言抽象层次低，程序员编写程序工作量大。

高级语言屏蔽了机器语言的细节，提高了语言抽象层次。程序中采用了具有一定含义的数据命令和容易理解的执行语句。高级语言不受计算机种类的限制，通用性强，易于理解、学习和掌握。

面向过程是程序设计的一种传统的方法，它把待解决的问题按功能划分为若干个相对独立的小问题，针对每个小问题编写相对独立的程序模块，并由主程序按解决问题的步骤组合程序模块，以形成完整的程序。

结构化程序设计采用了模块分解与功能抽象、自顶向下分而治之的方法，从而有效地将一个较复杂的程序设计任务分解成许多易于控制和处理的子任务，以便于开发和维护。

面向对象的程序设计方法将求解问题中所有的独立个体都看作各自不同的对象，与对象相关的数据和对数据操作的处理方法都放在一起，成为一个相互依存、不可分离的整体。面向对象的程序设计方法将具有共同特性的对象抽象出其共同的属性而形成类。类中的多数数据只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系。对象与对象之间通过消息进行通信。这样，程序模块间的关系更为简单，程序模块的独立性、数据的安全性有了良好的保障。

C++ 既支持面向过程的程序设计，又支持面向对象的程序设计。

C++ 程序由一个主函数、若干个子函数以及头文件组成的。主函数和子函数中使用的变

量要用定义语句确定其数据类型。头文件由一行或若干行预处理命令行组成。程序文件的扩展名默认为 cpp。

习 题 1

一、填空题

1. 计算机系统由_____和_____两部分组成。计算机部件主要包括_____、存储器芯片和_____、_____接口芯片。
2. 为了使计算机运行所编制的_____及其_____的总和称为软件。
3. 计算机程序设计语言分为_____、_____和_____。
4. 汇编语言和各种高级语言都要转换为_____表示的程序, 计算机才能识别和执行。将高级语言转换成机器语言要通过_____或_____。
5. _____把高级语言指令序列一次性的翻译成机器语言, 并生成相应的可执行的代码文件。
6. 面向对象的程序设计方法将与对象相关的_____和对数据的_____都放在一起, 成为一个相互依存、不可分离的整体。
7. C++既支持_____的程序设计, 又支持_____的程序设计。
8. C++程序由一个_____、若干个_____以及_____组成。
9. C++的源程序的文件扩展名默认为_____。可执行文件的主名与源程序的主名相同, 而扩展名为_____。
10. C++有两种加注释的方法: 其一使用_____; 其二是用_____和_____将注释内容包含在内。
11. 编译时检查出的错误包含_____类: 一类是严重错误, 系统用_____提示; 另一类是警告性错误, 系统用_____提示。

二、简答题

12. 简述机器语言程序、汇编语言程序和高级语言程序的主要特点。
13. 解释程序和编译程序的作用以及它们的不同特点。
14. 什么是对象? 什么是面向对象的程序设计方法?
15. 简述 C++程序的基本结构。
16. 简述 Visual C++ 6.0 的程序开发流程。
17. 注释有什么作用? C++中有哪几种注释的方法?
18. 编写一段 C++程序, 程序的功能实现在计算机屏幕上显示: "Hello! Welcome to C++!" 并在程序中用两种方式加注释。

三、写出执行下列程序的结果, 并上机实验

```
19. #include <iostream.h>
    void main( )
    {
        int x,y;
        x=5; y=6;
        cout <<"x+y="<<x+y<<',';
```

```

        cout <<"x*y="<<x*y<<endl;
    }
20. #include <iostream.h>
    int cube (int);
    void main(void)
    {
        cout <<"cube(3)="<<cube(3)<<endl;
        cout <<"cube(5)="<<cube(5)<<endl;
        cout <<"cube(8)="<<cube(8)<<endl;
    }
    int cube(int x)
    {
        return x*x*x;
    }
21. #include <iostream.h>
    #include "abc.cpp"
    void main( )
    {
        double a,b,c;
        double averageValue;
        a=2;b=3;c=4;
        averageValue =AVE(a,b,c)
        cout <<"averageValue:"<<averageValue<<endl;
        averageValue=AVE(a,b+1,c+2)
        cout <<"averageValue:"<<averageValue<<endl;
    }
    //其中 abc.cpp 文件的内容如下:
    double AVE(double x, double y, double z)
    {
        return (x+y+z)/3;
    }

```

实验 1.1 熟悉 Visual C++ 6.0 的集成开发环境

一、实验目的

1. 熟悉 Visual C++ 6.0 的集成开发环境。
2. 掌握 C++语言程序在 Visual C++环境下的编辑、运行方法。
3. 掌握在 Visual C++环境下 C++程序的简单调试。
4. 了解 Visual C++环境下多文件 C++编辑运行方法。

二、实验内容和步骤

1. 编辑一个能够输出“C++语言程序设计 沙市大学 武汉工程职业技术学院联合编著”等字符串的 C++ 程序并调试运行。

(1) 参考源代码如下:

```
#include <iostream.h>
void main( )
{
    cout<<"\n\n      C++程序设计基础\n";
    cout<<"\n\n      沙市大学\n";
    cout<<"武汉工程职业技术学院联合编著"<<endl;
}
```

(2) 调试运行上述程序。

(3) 总结程序调试过程中的经验和教训。

2. 编辑一个求和函数，主函数调用求和函数求出两组数据的和并将其输出。

(1) 参考源代码如下:

```
#include <iostream.h>
int sum(int x,int y)    //求和函数
{
    return x+y;
}
void main( )
{
    int a,b,c;
    cout<<"输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout<<"a="<<a<<" ,b="<<b<<" , a+b="<<c<<endl;
    cout<<"再输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout<<"a="<<a<<" ,b="<<b<<" , a+b="<<c<<endl;
}
```

(2) 调试运行上述程序。

(3) 总结程序调试过程中的经验和教训。

3. 基本内容同第 2 题，编辑一个 sum 函数，其文件名为 sum.h。

(1) 其他步骤同第 2 题，然后选择“工程”菜单的“添加工程”子菜单中“新建”选项。

(2) 选择“文件”选项卡中的“C/C++ Header File”选项，编辑 sum 函数。

(3) 参考源代码如下:

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

```

#include "sum.h"          //必须有此头文件
void main( )
{
    int a,b,c;
    cout <<"输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout <<"a="<<a <<" ,b="<<b <<" , a+b="<<c <<endl;
    cout <<"再输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout <<"a="<<a <<" ,b="<<b <<" , a+b="<<c <<endl;
}
int sum(int x,int y)      //sum.h 为文件名
{
    return x+y;
}

```

(4) 调试运行上述程序。

(5) 总结程序调试过程中的经验和教训。

4. 基本内容同第 2 题，编辑一个 sum 函数，其文件名为 sum.cpp。

(1) 其他步骤同第 2 题，然后选择“工程”菜单的“添加工程”子菜单中“新建”选项。

(2) 选择“文件”选项卡中的“C++ Source File”选项，编辑 sum 函数。

(3) 参考源代码如下：

```

#include <iostream.h>
int sum(int,int);
void main( )
{
    int a,b,c;
    cout <<"输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout <<"a="<<a <<" ,b="<<b <<" , a+b="<<c <<endl;
    cout <<"再输入 a 和 b 的值:";
    cin >>a >>b;
    c=sum(a,b);
    cout <<"a="<<a <<" ,b="<<b <<" , a+b="<<c <<endl;
}
int sum(int x,int y)      //sum.cpp 为文件名
{
    return x+y;
}

```

}

(4) 调试运行上述程序。

(5) 总结程序调试过程中的经验和教训。

习题 1 参考答案

1. 硬件, 软件, 微处理器芯片, 输入, 输出
2. 各种程序, 相关的文档资料
3. 机器语言, 汇编语言, 高级语言
4. 机器码, 解释程序, 编译程序
5. 编译程序
6. 数据, 操作处理方法
7. 面向过程, 面向对象
8. 主函数, 子函数, 头文件
9. cpp, exe
10. 双斜线 “//”, “/*”, “*/”
11. 两, error, warning

12. 机器语言是用二进制指令表示的、计算机能够直接识别和执行的语言。由于计算机可以直接执行机器语言, 执行速度快是机器语言的主要优点。在计算机发展初期, 软件工程师使用机器语言来编写程序。在计算机指令集不断增大的情况下, 机器语言不易记忆, 不利于学习, 软件开发难度大、周期长、易出错, 软件维护工作量大等缺点更加突出, 机器语言已不能胜任编程工作。

汇编语言是用一些易读的助记符表示机器指令操作码, 如 ADD、CALL 等。使用汇编语言编写程序提高了编程速度, 也比较容易编写出质量较高的程序, 检查、维护也比较容易。但汇编语言抽象层次低, 程序员编写程序工作量大。

高级语言屏蔽了机器语言的细节, 提高了语言抽象层次。程序中采用了具有一定含义的数据命令和容易理解的执行语句。高级语言不受计算机种类的限制, 通用性强, 易于理解、学习和掌握。使用高级语言减少了大量的编程工作量。目前使用比较广泛的高级语言有 Basic、Fortran、Pascal 和 C 语言。C++ 是一种功能强大、应用更广泛的高级语言。

13. 解释程序和编译程序的作用都是将高级语言转换成机器语言。但它们的工作方式不同。解释程序把高级语言代码翻译成机器代码, 翻译的过程是边翻译边执行。翻译一句, 执行一句, 直至将整个源程序翻译并执行完毕。解释程序不产生目标代码, 对源程序中的重复语句(即循环体中语句)需要重复地解释和执行。因此它执行程序的时间长, 效率较低。编译程序把高级语言指令序列一次性翻译成机器语言, 并生成相应的可执行代码文件。可执行文件可作为单独的执行程序而不需要编译程序的辅助。

14. 在面向对象的程序设计方法中将求解问题中所有独立个体都看作各自不同的对象。

面向对象的程序设计方法将与对象相关的数据和对数据的操作处理方法都放在一起, 成为一个相互依存、不可分离的整体。将具有共同特性的对象抽象出其共性形成类。类中的大多数数据只能用本类的方法进行处理。类通过一个简单的外部接口与外界发生关系。对象与对象之间通过消息进行通信。这样, 程序模块间的关系更为简单, 程序模块的独立性、数据

的安全性有了良好的保障。

15. C++程序由一个主函数、若干个子函数以及头文件组成。

头文件由一条或若干条预处理命令行组成,预处理命令行以#include 开头,在其后的尖括号内引用 C++系统函数。C++系统函数又称为标准函数、预处理函数或库函数。系统函数是由 C++语言系统本身为用户提供的一系列具有各种用途的函数的总称。

主函数是程序执行的开始点,由主函数头和主函数体组成。主函数头以 main 为标识;主函数体包含在一对大括号内。main()前的 void 是类型标识符,表示无返回值的函数。

子函数由子函数头和子函数体组成。子函数头主要有子函数名。子函数名由编程人员根据函数的功能选择。主函数调用子函数时,使用子函数名。子函数名前的类型说明符表示子函数返回值的属性,子函数名后面括号内是带入子函数中的形式参数。子函数体的内容安排在子函数头行下面的一对大括号内。

如果子函数在主函数之后,在主函数前要有子函数的原型语句。如果子函数在主函数之前,则可不写子函数的原型语句。

主函数体内的各行语句、子函数体内的各行语句以及子函数的原型语句都要以“;”结尾。

16. 用 C++语言编写好一个完整的程序后,第一步要上机运行编辑程序,输入源程序;第二步是对源程序进行编译生成目标代码文件,目标代码文件名与源程序文件名相同,而扩展名改为 obj;第三步运行连接程序,将目标代码文件与 C++的库文件相连接,生成可执行文件,可执行文件名与源程序文件名相同,而扩展名为 exe;第四步运行可执行文件,实现程序所具有的功能。

17. 注释在程序中的作用是对程序进行注解和说明。在源程序中加入注释,帮助使用者对程序功能的理解,增强了程序的可读性。编译程序在对源程序进行编译时会自动忽略注释部分。因此注释部分对于程序的执行功能没有任何作用。

在源程序的任何地方都可加入注释,但多数情况下将注释加在语句行的结尾或单独列一行(称为注释行)。C++有两种加注释的方法:① 使用双斜线“//”,从它开始到该语句行尾的所有内容都被看作为注释;② 在“/*”和“*/”之内的所有内容被视为注释。这是一种沿用 C 语言的注释方法。

18. #include <iostream.h> //进行标准 I/O 操作要引入 iostream.h 头文件

//以下是主函数

void main()

{ //以下是函数体*/

cout << "Hello! Welcome to C++! \n"; //屏幕显示

}

19. x+y=11,x*y=30

20. cube(3)=27

cube(5)=125

cube(8)=512

21. averageValue:3

averageValue:4

第二章 数据类型、运算符及表达式

2.1 基本要求

1. 掌握 C++ 语言的词法, 其中包括字符集、保留字和标识符。
2. 掌握 C++ 的数据类型, 尤其熟记基本数据类型及其表示范围。
3. 理解 C++ 的常量、变量及其差异。
4. 掌握 C++ 的算术运算符、关系运算符、逻辑运算符、位逻辑运算符及其对应的表达式的求值和有关规则。

2.2 内容小结

(1) C++ 语言的字符集由以下字符构成: 英文字母(52 个): A~Z, a~z。数字符号(10 个): 0~9; 特殊符号(30 个):

~	!	#	%	^	&	*	(	)	-
_	+	=	{	}	[	]		\	:
;	"	'	<	>	,	.	/	?	空格

保留字是 C++ 系统预定义的、用英文小写字母组成的特定单词。每个保留字都被 C++ 系统赋予了一定的含义, 并具有相应的功能。

C++ 标识符是由英文字母、下划线 “_” 和十进制数字符号组成的一串字符, 以英文字母或下划线 “_” 开始; 大写字母和小写字母代表不同的标识符; 不能使用 C++ 保留字。标识符的字符数可以有任意个, 但只有前 32 个字符有效。

(2) C++ 的基本数据类型有 bool(布尔型, 又称逻辑型)、char(字符型)、int(整型)、float(浮点型, 表示实数)和 double(双精度浮点型, 简称双精度型)。

(3) 常量是指在程序运行的整个过程中其值不能改变的量。常量有直接常量和符号常量(又称标识符常量)两种表示方法。

整型常量是以字符形式出现的整数, 它有十进制、八进制和十六进制 3 种表示方法。

实型常量是以十进制方式出现的实数, 它有一般形式和指数形式两种表示方式。

字符常量是用单引号括起来的一个字符。有一些字符是不可显示字符, 也无法通过键盘输入。如响铃、换行、回车等, C++ 提供了一种转义序列来从键盘输入。

字符串常量是用一对双引号括起来的字符序列。

布尔常量又称逻辑常量。布尔常量只含有两个值: 0 表示逻辑假, 用常量 false 表示; 1 表示逻辑真, 用常量 true 表示。

(4) 程序在执行过程中其值可以变化的量称为变量。变量声明语句的格式:

数据类型 变量名 1, 变量名 2, ..., 变量名 n;

符号常量声明语句的形式:

const 数据类型说明符 常量名=常数值;

或

数据类型说明符 `const` 常量名=常量值;

(5) 运算符又称为操作符,它是对数据进行运算的符号,参与运算的数据称为操作数。由操作数和操作符连接而成的式子称为表达式。

① 基本算术运算符:加(+)、减(-)、乘(*)、除(/)、取余(%)和取负值(-)。取余运算只用于整型操作数, $a \% b$ 的结果是 a 被 b 除的余数。除法运算用于两个整型数相除时,其结果是整数部分,小数部分被自动舍弃,如: $1/2$ 的结果是 0。

② 赋值运算符“=”。赋值运算是将等号右边表达式的值赋给等号左边的变量。C++的复合赋值运算有+=, -=, *=, /=, %=, <<=, >>=, &=, ^=和|=等 10 种。

③ 关系运算符:小于(<)、小于等于(<=)、大于(>)、大于等于(>=)、等于(==)和不等于(!=)。6 个关系运算都是双目运算,用来比较两个操作数的大小,运算结果均为逻辑值 0(false)或 1(true)。

④ 逻辑运算符:逻辑非(!)、逻辑与(&&)和逻辑或(||)。逻辑非的优先级最高,逻辑与次之,逻辑或最低。逻辑非是单目运算,逻辑与和逻辑或都是双目运算。

⑤ 位运算符:按位与(&)、按位或(|)、按位异或(^)、按位取反(~)、左移(<<)和右移(>>)。

⑥ 增 1、减 1 运算符:++, --。由增 1、减 1 运算符和左值表达式构成的表达式是增 1、减 1 表达式。使用增 1、减 1 运算符需注意:增 1、减 1 运算的操作对象只能是左值表达式即只允许为变量,而不允许为常量或表达式、函数调用等。增 1、减 1 运算符的操作数通常是整型或字符型,因为上述类型的数据操作后其值是确定的;而实型数操作前后并不能确保其差的绝对值是 1。增 1、减 1 操作又分前置和后置操作两类,它们有质的区别。增 1、减 1 运算是除了第 1 级外优先级最高的。但是后置操作则要注意:先引用其值参加运算,再做后置操作。

注意:增 1、减 1 操作的副作用,所谓“副作用”是指某些变量依赖于表达式的计值方法而发生的改变。

⑦ 其他运算符:

- “?” 是 C++惟一的一个三元条件运算,可实现简单的条件选择功能。

条件运算的使用格式:表达式 1 ? 表达式 2: 表达式 3

其中:表达式 1 必须是 bool 类型,表达式 2 和表达式 3 可以是任意类型。条件运算的执行顺序是:先求解表达式 1,当表达式 1 的值为 true 时,将表达式 2 的值作为最终结果,当表达式 1 的值为 false 时,将表达式 3 的值作为最终结果。

- “,” 运算是一种顺序运算,它用逗号分开若干个表达式。计算时,按照每个子表达式从左到右的先后次序依次计算它们的值,最后一个表达式的值是整个逗号运算表达式的值。

- “sizeof” 是测试类型长度运算符,用于计算某种类型对象在内存中所占的字节数。

- 圆括号运算能够改变运算的优先级,使括号内的运算优先进行。

2.3 典型例题分析

由于 C++语言的运算符的优先级和结合性较之其他高级语言复杂,以下通过 4 个例题对其进行综合讨论,以使读者能深入掌握 C++语言运算符和表达式的应用。

例 2.1 算术表达式应用举例。

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

```

void main()
{   int x;
    x=-3+4*5-6; cout<<x;
    x=3+4%5-6; cout<<" "<<x;
    x=-3*4%-6/5; cout<<" "<<x;
    x=(7+6)%5/2; cout<<" "<<x<<endl;
}

```

分析：对于“ $x=-3+4*5-6$ ；”中的常量 3 前的“-”号是取负运算符，其优先级高于本表达式中的所有其他运算符，应先进行计算，即值为-3，所以 x 的值应该是 11。对于“ $x=3+4\%5-6$ ；”，应注意取余运算符“%”，“ $4\%5$ ”其余数是 4，所以 x 的值为 1。对于“ $x=-3*4\%-6/5$ ；”，有两处取负运算，“-3”和“-6”中的负号的优先级高于*、/、%运算，而*、%、/在同一表达式中出现，因为它们具有同一优先级，所以要从左到右进行计算： $-3*4$ 等于-12， $-12\%-6$ 其余数为 0， $0/5$ 其商为 0，所以 x 的值应该是 0。对于“ $x=(7+6)\%5/2$ ；”，由于圆括号的作用使 7+6 的优先级提高， $(7+6)\%5$ 余数为 3， $3/2$ 商为 1，所以 x 的值为 1。

因此，例 2.1 程序的运行结果：

11, 1, 0, 1

例 2.2 条件表达式应用举例。

```

#include <iostream.h>
void main()
{   int a,b,c,d; bool f;
    a=b=c=1;
    a+=b+=c;d=a<b?b:a;
    cout<<d;
    d=a<b?a++:b++;
    cout<<" "<<d;
    cout<<" "<<a<<" "<<b;
    c+=a<b?a++:b++;
    cout<<" "<<c<<" "<<b<<" "<<c;
    a=3;b=c=4;
    f=(c>=b>=a)?1:0;
    cout<<" "<<f;
    f=c>=b&& b>=a;
    cout<<" "<<f<<endl;
}

```

分析：由于自动整型变量 a, b, c 被赋值为 1，执行“ $a+=b+=c$ ；”的结果是： $a=3, b=2, c=1$ 。对于表达式“ $a<b?b:a$ ”，由于 $a<b$ 不成立(false)，所以条件表达式的 e3 作为条件表达式的值，即 d 输出为 3。对于表达式“ $a<b?a++:b++$ ”，只是 $b++$ 作为条件表达式的值，由于 $b++$ 是后置增 1 操作，故输出 b 的原值 2，输出后 b 的值由 2 变为 3；由于 $a++$ 在表达式中未被执行，故维持 a 的原值，所以输出为 3，输出 b 值为 3。由于“ $c+=a<b?a++:b++$ ”中 $a<b$ 不成立，故以 e3 作为条件表达式的值参加 $c+=b++$ 运算。由于 c 原值是 1，变量 b 的后置加 1 之前的值为 3，

所以变量 c 运算后值为 4 并输出；变量 b 经后置运算后也变为 4，输出变量 b，c 的值为 4 和 4。由于 a=3 和 b=c=4 的赋值操作，使 a 值为 3，b 和 c 的值为 4。条件表达式“(c>=b>=a)?1:0”中的 e1 为 c>=b>=a，这是一个关系表达式，由于其结合性是从左至右，则先判 c>=b 的关系式值为逻辑值 1，而 1>=a 则不成立，则其逻辑值为 0，故取 e3 作为条件表达式的值，即为 0。“c>=b&&b>=a”是一个逻辑表达式，由于 c>=b 成立，其值为 1，又由于 b>=a 成立，其值为 1，则 1&&1 结果为逻辑值 1。这是因为逻辑运算优先级低于关系运算的缘故。

因此，例 2.2 程序的运行结果：

3, 2, 3, 3, 4, 4, 4, 0, 1

例 2.3 移位表达式应用举例。

```
#include <iostream.h>
void main()
{ unsigned short b=248,bb;
  short a=-8,aa;
  aa=a>>2;
  cout<<"带符号数"<<a<<"右移两位的值是:"<<aa<<endl;
  bb=b>>2;
  cout<<"无符号数"<<b<<"右移两位的值是:"<<bb<<endl;
}
```

分析：变量 a 被定义为有符号短整型数，其值赋予-8，其二进制数表示形式为 11111000。其右移操作是将低位移弃，而高位是按原符号位复制，右移 2 位后为 11111110，即其值为-2。变量 b 是无符号短整型数，其初始值为 248，其二进制数表示形式为 11111000。其右移操作将低位移弃，由于是无符号数，故其高位填补零，即变为 00111110，也就是无符号数值为 62。它们都遵循 $k/2^n$ 的原则，即 $-8/4=-2$ ， $248/4=62$ 。

因此，例 2.3 程序的结果：

带符号数 a=-8 右移两位的值是-2

无符号数 b=248 右移两位的值是 62

例 2.4 逻辑表达式应用举例。

```
#include <iostream.h>
void main()
{ int x,y,z,w;
  x=2; y=1; z=0;
  x=x&&ylz;
  cout<<x;
  w=x||y&&z;
  cout<<" , "<<w;
  x=y=1; z=x++-1;
  cout<<" , "<<x<<" , "<<z;
  z+=-x+++y;
  cout<<" , "<<x<<" , "<<z<<endl;
}
```