



教育部职业教育与成人教育司推荐教材
(五年制) 高等职业教育电子信息类教学用书

21世纪高职高专系列规划教材

C++语言教程

主 编 翟乃强 廖少俊



北京师范大学出版社



清华大学出版社
Tsinghua University Press

清华大学出版社 清华大学出版社

C++语言教程

清华大学出版社 清华大学出版社



教育部职业教育与成人教育司推荐教材
21世纪高职高专系列规划教材

C++语言教程

主 编 翟乃强 廖少俊



北京师范大学出版社

内 容 简 介

C语言是高校计算机相关专业的基础课程,本书并不对相关的C语言知识进行阐述,因此本书适用于有一定C语言编程基础的读者。本书共分为9章。第1章为绪论,介绍了C++的发展史、C++的词法约定以及面向对象程序设计的基本思想。第2章为类与对象,介绍了类的定义方法、成员函数、构造函数和析构函数、类静态成员与友元以及this指针的内容。第3章为继承,介绍了基类与派生类、字类型化和类型适应以及多继承和虚基类。第4章为运算符重载,介绍了运算符重载函数的两种形式以及常用运算符的重载和类型转换。第5章为虚拟函数与多态性,介绍了虚拟函数,纯虚函数以及抽象类。第6章为C++的输入和输出 C++的流类库,预定义输入输出以及文件的输入输出。第7章为模板,介绍了模板的需求与基础、模板的特化、基类模板、多态性以及仿真动态绑定和函数模板。第8章为Windows编程初步,介绍了Windows程序与DOS程序的区别、线程、消息、窗口构件以及一般Windows程序的组成和图形基础对象。第9章为异常处理和名字空间,介绍了异常处理的机制和名字空间的使用法。

本书适合用做高职教材,也可供相关人员参考。

图书在版编目(CIP)数据

C++语言教程 / 翟乃强, 廖少俊编. —北京: 北京师范大学出版社, 2005.8
(21世纪高职高专系列规划教材)
ISBN 7-303-07659-X

I. C... II. ①翟... ②廖... 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086361号

北京师范大学出版社出版发行
(北京新街口外大街19号 邮政编码: 100875)

<http://www.bnup.com.cn>

出版人: 赖德胜

北京新丰印刷厂印刷 全国新华书店经销

开本: 185 mm × 260 mm 印张: 15.75 字数: 330千字

2005年9月第1版 2005年9月第1次印刷

印数: 1~3 000 定价: 21.00元

出版说明

随着我国经济建设的发展,社会对技术型应用人才的需求日趋紧迫,这也促进了我国职业教育的迅猛发展,我国职业教育已经进入了平稳、持续、有序地发展阶段。为了适应社会对技术型应用人才的需求和职业教育的发展,教育部对职业教育进行了卓有成效的改革,职业教育与成人教育司、高等教育司分别颁布了调整后的中等职业教育、高等职业教育专业设置目录,为职业学校专业设置提供了依据。教育部连同其他五部委共同确定数控技术应用、计算机应用与软件技术、汽车运用与维修、护理等四个专业领域为紧缺人才培养专业,选择了上千家高职、中职学校和企业作为示范培养单位,拨出专款进行扶持,力争培养一批具有较高实践能力的紧缺人才。

职业教育的快速发展,也为职业教材的出版发行迎来了新的春天和新的挑战。教材出版发行为职业教育的发展服务,必须体现新的理念、新的要求,进行必要的改革。为此,在教育部高等教育司、职业教育与成人教育司、北京师范大学等的大力支持下,北京师范大学出版社在全国范围内筹建了“全国职业教育教材改革与出版领导小组”,集全国各地上百位专家、教授于一体,对中等职业、高等职业文化基础课、专业基础课、专业课教材的改革与出版工作进行深入地研究与指导。2004年8月,“全国职业教育教材改革与出版领导小组”召开了“全国有特色高职教材改革研讨会”,来自全国20多个省、市、区的近百位高职院校的院校长、系主任、教研室主任和一线骨干教师参加了此次会议。围绕如何编写出版好适应新形势发展的高等职业教育教材,与会代表进行了热烈的研讨,为新一轮教材的出版献计献策。这次会议共组织高职教材50余种,包括文化基础课、电工电子、数控、计算机教材。其特点如下:

1. 紧紧围绕教育改革,适应新的教学要求。教育部等六部委联合发文确定紧缺型人才培养战略,并明确提出了高等职业教育将从3年制逐渐向2年制过渡。过渡时期具有新的教学要求,这批教材是在教育部的指导下,针对过渡时期教学的特点,以2年制为基础,兼顾3年制,以“实用、够用”为度,淡化理论,注重实践,消减过时、用不上的知识,内容体系更趋合理。

2. 教材配套齐全。将逐步完善各类专业课、专业基础课、文化基础课教

材,所出版的教材都配有电子教案,部分教材配有电子课件和实验、习题指导。

3.教材编写力求语言通俗简练,讲解深入浅出,使学生在理解的基础上学习,不囫圇吞枣,死记硬背。

4.教材配有大量的例题、习题、实训,通过例题讲解、习题练习、实验实训,加强学生对理论的理解以及动手能力的培养。

5.反映行业新的发展,教材编写注重吸收新知识、新技术、新工艺。

北京师范大学出版社是教育部职业教育教材出版基地之一,有着近20年的职业教育教材出版历史,具有丰富的编辑出版经验。这批高职教材是针对2/3年制编写的,同时也向教育部申报了“2004—2007年职业教育教材开发编写规划”,部分教材通过教育部审核,被列入职业教育与成人教育司5年制高职推荐教材。我们还将开发电子信息类的通信、机电、电气、计算机等其他专业,以及工商管理、财会等方面教材,希望广大师生积极选用。

教材建设是一项任重道远的工作,需要教师、专家、学校、出版社、教育行政部门的共同努力才能逐步获得发展。我们衷心希望更多的学校、更多的专家加入到我们的教材改革出版工作中来,北京师范大学出版社职业与成人教育事业部全体人员也将备加努力,为职业教育的改革与发展服务。

全国职业教育教材改革与出版领导小组
北京师范大学出版社

参加教材编写的单位名单

(排名不分先后)

沈阳工程学院

山东劳动职业技术学院

济宁职业技术学院

辽宁省交通高等专科学校

浙江机电职业技术学院

杭州职业技术学院

西安科技大学电子信息学院

西安科技大学机械学院

天津渤海职业技术学院

天津渤海集团公司教育中心

连云港职业技术学院

景德镇高等专科学校

徐州工业职业技术学院

广州大学科技贸易技术学院

江西信息应用职业技术学院

浙江商业职业技术学院

内蒙古电子信息职业技术学院

济源职业技术学院

河南科技学院

苏州经贸职业技术学院

浙江工商职业技术学院

温州大学

四川工商职业技术学院

常州轻工职业技术学院

河北工业职业技术学院

太原理工大学轻纺学院

浙江交通职业技术学院

保定职业技术学院

绵阳职业技术学院

北岳职业技术学院

天津职业大学

北京轻工职工职业技术学院

石家庄信息工程职业学院

襄樊职业技术学院

九江职业技术学院

青岛远洋船员学院

无锡科技职业学院

广东白云职业技术学院

三峡大学职业技术学院

西安欧亚学院实验中心

天津机电职业技术学院

漯河职业技术学院

济南市高级技工学校

沈阳职业技术学院

江西新余高等专科学校

赣南师范学院

前 言

C++程序设计是面向对象编程的基石,它不仅保留了C语言功能强、目标代码效率高、可移植性好的优点,而且还提供了面向对象的程序设计能力。由于C++强大的功能,目前它已经成为全世界专业编程人员的首选语言。由于C语言是高校中计算机相关专业的基础课程,本书并不对相关的C语言知识进行阐述,因此本书适用于有一定C语言编程基础的读者。

本书共分为9章。第1章为绪论,介绍了C++的发展史、C++的词法约定以及面向对象程序设计的基本思想;第2章为类与对象,介绍了类的定义方法、成员函数、构造函数和析构函数、类静态成员与友元以及this指针的内容;第3章为继承,介绍了基类与派生类、字类型化和类型适应以及多继承和虚基类;第4章为运算符重载,介绍了运算符重载函数的两种形式以及常用运算符的重载和类型转换;第5章为虚拟函数与多态性,介绍了虚拟函数,纯虚函数以及抽象类;第6章为C++的输入和输出C++的流类库,预定义输入输出以及文件的输入输出;第7章为模板,介绍了模板的需求与基础、模板的特化、基类模板、多态性以及仿真动态绑定和函数模板;第8章为Windows编程初步,介绍了Windows程序与DOS程序的区别、线程、消息、窗口构件以及一般Windows程序的组成和图形基础对象;第9章为异常处理和名字空间,介绍了异常处理的机制和名字空间的使用法。

本书的第1、2、3章由廖少俊编写,第4、6章由闫思江编写,第5章由费建刚编写,第7章由冯文仙编写,第8、9章由翟乃强编写,最后由翟乃强定稿,杜海平、王亮、邓居英、王晓等参与了本书的编排工作。

本书在写作过程中,得到了许多专家的大力支持,参考了大量的文献资料,在此表示诚挚的感谢。

限于作者的水平有限,书中难免有错误或不妥之处,恳切希望读者予以指正。我们也会在适当的时间进行修改和补充。

编 者

2005年6月

目 录

第 1 章 绪论	(1)
1.1 C++ 的发展史	(1)
1.2 C++ 的词法与规则	(2)
1.3 结构化程序设计	(17)
1.4 面向对象的程序设计	(17)
1.5 简单的 C++ 程序	(23)
本章小结	(27)
第 2 章 类与对象	(28)
2.1 类的定义	(28)
2.1.1 类的概念	(28)
2.1.2 类的定义格式	(29)
2.1.3 对象的定义	(31)
2.2 类的封装	(31)
2.3 成员函数	(33)
2.3.1 成员函数的说明	(33)
2.3.2 不同类型的成员函数	(34)
2.3.3 成员函数的重载	(38)
2.4 构造函数与析构函数	(39)
2.4.1 构造函数	(39)
2.4.2 析构函数	(42)
2.4.3 调用构造函数与析构函数	(44)
2.5 类静态成员与友元	(46)
2.5.1 静态数据成员	(46)
2.5.2 静态成员函数	(47)

2.5.3 友元	(49)
2.6 this 指针与引用	(57)
2.6.1 this 指针	(57)
2.6.2 引用	(61)
本章小结	(64)
习题与思考题	(65)
第 3 章 继承	(69)
3.1 基类与派生类	(70)
3.2 单继承	(76)
3.3 子类型化和类型适应	(82)
3.3.1 子类型化	(82)
3.3.2 类型适应	(82)
3.4 多继承	(83)
3.4.1 多继承的定义	(83)
3.4.2 多继承的构造函数和析构函数	(89)
3.4.3 二义性问题	(91)
3.5 虚基类	(94)
本章小结	(97)
习题与思考题	(98)
第 4 章 运算符重载	(103)
4.1 运算符重载概述	(103)
4.1.1 运算符重载的好处	(103)
4.1.2 运算符重载的规则	(106)
4.2 运算符重载函数的两种	



形式	(107)	6.2 C++的流库及其基本	
4.2.1 友元运算符函数	(108)	结构	(156)
4.2.2 成员运算符函数	(115)	6.2.1 C++的流	(156)
4.2.3 成员运算符函数与友元运		6.2.2 流类库	(156)
算符函数的比较	(120)	6.3 预定义类型的输入/输出	
4.3 几个常用运算符的重载			(159)
.....	(122)	6.3.1 无格式输入/输出	(159)
4.3.1 单目运算符“++”和“--”		6.3.2 格式化输入/输出	(161)
的重载	(122)	6.4 用户自定义类型的输入/	
4.3.2 赋值运算符“=”的重载		输出	(172)
.....	(127)	6.4.1 重载输出运算符“<<” ..	(172)
4.3.3 下标运算符“=”的重载		6.4.2 重载输入运算符“>>” ..	(173)
.....	(130)	6.5 文件的输入/输出	(175)
4.3.4 运算符 new 和 delete 的重载		6.5.1 文件的打开与关闭	(175)
.....	(132)	6.5.2 文件的读/写	(178)
4.4 类型转换	(135)	6.6 程序举例	(185)
4.4.1 系统预定义类型间的转换		本章小结	(188)
.....	(135)	习题与思考题	(189)
4.4.2 类类型与系统预定义类型		第7章 模板	(191)
间的转换	(135)	7.1 模板简介	(191)
本章小结	(141)	7.2 模板需求与基础	(192)
习题与思考题	(142)	7.3 作为基类的模板	(193)
第5章 虚拟函数与多态性	(145)	7.4 多态性	(194)
5.1 静态连接与动态连接 ..	(145)	7.5 仿真动态绑定	(195)
5.2 虚拟函数	(147)	7.6 函数模板	(196)
5.3 纯虚函数与抽象类	(151)	本章小结	(198)
本章小结	(154)	习题与思考题	(199)
习题与思考题	(154)	第8章 Windows 编程初步	(201)
第6章 C++的输入和输出		8.1 Windows 程序和DOS程	
.....	(155)	序的不同点	(201)
6.1 C++为何建立自己的输		8.2 线程介绍	(201)
入/输出系统	(155)	8.3 理解消息	(202)

8.4 窗口的构件	(203)	9.1 异常处理	(226)
8.5 父窗口和子窗口	(203)	9.1.1 基本的异常处理形式 ...	(226)
8.6 创建一般的 Windows		9.1.2 异常具有特定类型	(227)
程序	(204)	9.1.3 try 块内的函数抛出异常	
8.6.1 标识符、结构与句柄	(208)		(228)
8.6.2 匈牙利表示法	(210)	9.1.4 函数内的局部 try 块	(229)
8.6.3 注册窗口类型	(210)	9.1.5 程序何时执行 catch	(230)
8.6.4 建立窗口	(213)	9.1.6 对单个 try 块使用多个	
8.6.5 显示窗口	(214)	catch 语句	(230)
8.6.6 消息循环	(214)	9.1.7 对异常使用省略符	(232)
8.6.7 窗口消息处理函数	(216)	9.1.8 捕获单个 try 块的显式和通	
8.6.8 处理消息	(216)	用异常	(233)
8.6.9 Windows 程序设计的难点		9.1.9 异常的限制	(234)
.....	(218)	9.1.10 重新抛出异常	(236)
8.7 图形基础	(219)	9.1.11 异常处理的应用	(237)
8.7.1 画笔对象	(220)	9.2 名字空间	(238)
8.7.2 画刷对象	(221)	9.2.1 名字空间的定义	(238)
8.8 鼠标消息基础	(221)	9.2.2 using namespace 语句 ...	(240)
8.9 综合示例	(222)	9.2.3 嵌套名字空间	(241)
本章小结	(225)	9.2.4 名字空间的别名	(242)
习题与思考题	(225)	本章小结	(243)
第 9 章 异常处理与名字空间		习题与思考题	(243)
.....	(226)		

第1章 绪论

本章导读

- C++语言的发展史
- C++语言的编程特点
- 如何编写简单的C++程序

1.1 C++的发展史

C语言是贝尔实验室(Bell Laboratories)的 Denms Ritchie 在B语言的基础上开发出来的,1972年在一台 DEC PDP-11 计算机上实现了最初的C语言。它为人们所认识,是由于运用它作为 UNIX 操作系统的开发语言获得了巨大的成功。C语言具有如下特点:

(1)语言简洁、紧凑,使用方便、灵活,书写形式自由。C语言只有32个关键字。

(2)丰富的运算符和数据类型。

(3)C语言可以直接访问内存地址,能进行位操作,因而承担能够开发操作系统的工作。

(4)生成的目标代码质量高,程序运行效率高。

(5)可移植性好。

在C语言盛行的同时,也暴露出它的局限性:

(1)C类型的检查机制相对比较弱,有些错误在编译的时候不能发现。

(2)C语言本身几乎没有支持代码重用的语言结构,往往一个精心设计好的程序,很难为其他的程序所用。

(3)程序的规模达到一定的程度时,控制程序的复杂性非常困难。

为了克服这些缺陷,1980年贝尔实验室的 Bjame Stroustrup 开始对C语言进行改进和扩充。他发展和完善适用于系统程序设计的C语言设施,后来又充分吸收 Simula 语言在组织与设计方面的特性;兼具这两方面的特点,从而成功地开发出C++语言。

C++的许多特性是从C语言中派生的,它在引进面向对象的同时保留了C语言的简洁性和高效性,并且具有强有力处理硬、软件接口和低层系统程序设计的能力。C++最初的版本被称做“带类的C(C with classes)”[Stroustrup,1980],在1980年被第一次投入使用,当时它只支持系统程序设计和数据抽象技术。1983年,支持面



面向对象程序设计的语言设施被加入,正式取名为 C++;之后,面向对象设计方法和面向对象程序设计技术就逐渐进入了 C++ 领域。

随着若干独立开发的 C++ 实现产品的出现和广泛应用,正式的 C++ 标准化工作在 1990 年启动。标准化工作由 ANSI(American National Standard Institute)以及后来加入的 ISO(International Standards Organization)负责。1998 年正式发布了 C++ 语言的国际标准[C++,1998]。在标准化工作进展期间,标准委员会充当了一个重要的角色,其发布的 C++ 标准之草案在正式标准发布之前,一直被作为过渡标准而存在。

C++ 程序设计语言提供了一种用于计算和存储的模型,这个模型与大多数计算机所使用的计算和存储模型极为接近。C++ 的语言结构还提供了具有可适应性的、强大的抽象机制,用于对问题进行抽象。这种语言结构允许程序员创建和使用新的型别(type),而这些新的型别则可以与实际应用中所包含的概念相适应。由此可知,C++ 既支持面向低层次的程序设计风格,又支持面向高层次的程序设计风格。其中,面向低层次的程序设计风格是基于对硬件资源的直接操纵,以此来获得相当高的效率;而面向高层次的程序设计风格则是基于用户自定义的新的型别,以此来提供这样一种数据和计算模型:当完成一项任务时,计算机使用的这种模型与人类自身观察并完成该项任务时所使用的模型极为相似。这些面向高层次的程序设计风格通常被描述为数据抽象、面向对象程序设计以及范型程序设计。

C++ 的设计目标,就是要让 C++ 既具有适合于系统程序设计的 C 语言所具有的可适应性和高效性,又能在其程序组织结构方面具有像 Simula 那样的语言设施(Simula 所支持的这种程序组织结构通常被称为面向对象程序设计风格)。在设计的时候,还做了很大的努力,使得引借自 Simula 的高层次的程序设计技术能够应用于系统程序设计之中。这就是说,C++ 所提供的抽象机制能够被应用于那些对效率和可适应性具有极高要求的程序设计任务之中。

1.2 C++ 的词法与规则

C++ 程序是由语句、定义、说明等构造的,而它们都是由被称做“词法元素”或“语言符号”的元素所构造的。

C++ 和 C 的字符集相同,它的字符集由大小写英文字母(a~z、A~Z),数字字符(0~9)和标点符号(! % ~ & * () - + = { } | [] / \ ; ' " < > ? , . : _ ^ #)组成。

1. 语言符号

一个语言符号是 C++ 程序中对编译器有意义的最小单元。C++ 语法分析程序识别以下这些种类的语言符号:标识符、关键字、文字、运算符、标点符号和其他分

隔符号。用一串语言符号构成一个转换单元。

语言符号通常由空格、换行符或注释等分开。试看下面这条语句：

```
int ratio = 0.5; // the convert ratio
```

包含在这个语句中的语言符号(词法元素)依次是:关键字、空格、标识符、运算符、常量、分隔符和注释。

词法分析程序通过对输入字符从左到右进行扫描,尽可能地输入流中创建最长的语言符号集来分割语言符号。如下面代码:

```
a=i+++j;
```

写此代码的程序员可能有以下两种目的:

```
a=i+(++j);与 a=(i++)+j;
```

由于语法分析程序从输入流尽可能创建最长的符号集,所以选择第二种解释,得到符号 `i++`、`+` 和 `j`。

2. 注释

注释是编译器忽略的文本,但对程序设计者而言却非常有用。注释通常对代码进行标注以便以后参考,编译它们作为空白对待。

程序设计者在调试程序中,往往应用注释使得一些特定代码不运行,从而达到控制的目的。当然, `#if` 和 `#endif` 预处理命令在这方面效果更为方便,因此可以括起包含注释在内的代码,而嵌套注释是不允许的。

一个 C++ 注释可以按以下任一种方式书写:

- “`/*`”(斜杠,星号)字符,后面跟随任意字符序列(包括新行),最后加上“`*/`”。这种语法与 ANSI C 的一样。

- “`//`”(两个斜杠)字符,后面跟随任意字符序列。一个不是紧随后面反斜杠的新行将结束此种形式的注释。因此,它通常被称为“单行注释”。

注释字符(“`/*`”、“`*/`”和“`//`”)在字符常量、字符串文字或注释内部没有任何特殊意义。因此,使用第一种方法的注释不能被嵌套使用,考虑以下例子:

```
/* Intent: Comment out this block of code.
Problem: Nested comments on each line of code are illegal.
FileName=String("hello.dat");/* Initialize file string */
Cout <<"File:"<<FileName <<"\n";/* Print status message */
*/
```

执行代码不能编译,因为编译器扫描输入流,从第一个“`/*`”到第一个“`*/`”,则认为它是一个注释。在这种情况下,一个“`*/`”出现在注释 `Initialize file string` 的末尾。那么对最后一个“`*/`”,则缺少一个“`/*`”与其配对了。

3. 标识符

一个标识符是一个用于对以下内容之一进行编码的字符串序列:

(1)对象或变量名称;



- (2)类、结构或联合名称；
- (3)枚举的类型名称；
- (4)类、结构、联合或枚举的成员；
- (5)函数或类成员函数；
- (6)typedef 名称；
- (7)标号名称；
- (8)宏名称；
- (9)宏参数；
- (10)对标识符有以下限制：

- 标识符中的字符可以是下划线、字母(包括大写字母、小写字母)和数字，即

```
_ a b c d e f g h I j k l m  
n o p q r s t u v w x y z  
A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z  
0 1 2 3 4 5 6 7 8 9
```

- 标识符必须以非数字字符开头，即标识符的第一个字符必须是字母或下划线。
- 标识符不能与关键字使用一样的拼写和大小写方式。但标识符中含有关键字是合法的，例如，pint 是一个合法的标识符，尽管它包含了关键字 int。
- 标识符对大小写敏感，也就是说，标识符中的大写字母和小写字母意义是不同的。例如，fileName、FileName 和 FILENAME 是不同的标识符。

在一个标识符的开头使用连续的两个下划线(_ _)或一个下划线打头后跟一个大写字母，则在所有范围中为 C++ 的实现而保留，应当避免使用一个打头的下划线带一个小写字母作为有文件范围的名称，因为它可能在现在或将来与保存的标识符相冲突。

标识符流行驼峰式的写法，例如 MyDocument、YourFile。

比较好的标识符应该让人望文生义，便于阅读和理解程序。

由于用户定义类型的名称由编译器“修饰”以保存类型信息，使得修饰的标识符的限制变得复杂，结果名称，包括类型信息，不能超过 247 个字符长度。所以，Microsoft C++ 标识符的前 247 个字符才是有意义的(详见联机“Microsoft Visual C++ 6.0 程序员指南”中的“修饰名称”)。

能够影响修饰的标识符长度的因素有：

- 根据标识符表示一个用户定义类型的对象还是表示用户定义类型的派生类型的对象。
- 根据标识符表示一个函数还是表示函数派生的类型。
- 根据一个函数的参量的个数。

4. 关键字

关键字是系统中已预定义的单词,它们在程序中表达特定的含义。下面列举出 C++ 语言中常用的关键字:

auto	break	case	char	class	const
continue	default	delete	do	double	else
enum	explicit	extern	float	for	friend
goto	if	inline	int	long	mutable
new	operator	private	protected	public	register
return	short	signed	sizeof	static	static_cast
struct	switch	this	typedef	union	unsigned
virtual	void	while			

5. 运算符

运算符是一些用来进行某种操作的单词,是连接数据容器的纽带。在程序设计语言中,运算符起着不可或缺的作用。它实际上是系统预定义的函数名,这些函数作用于被操作的对象上将获得一个结果值。

C++ 向用户提供了丰富的运算符。按照各自的功能,可分为以下几大类:

- 算术运算符
- 关系运算符
- 逻辑运算符
- 按位运算符
- 特殊运算符

(1) 算术运算符

C++ 中的基本算术运算符如表 1.1 所示。

表 1.1 算术运算符表

运 算 符	说 明	举 例
+	加运算	34+56 结果为 90
-	减运算	100-25 结果为 75
-	取负运算	-45 结果为 -45
*	乘运算	15*5 结果为 75
/	除运算	18/6 结果为 3
%	模运算	17%5 结果为 2

减运算和取负运算虽然运算符相同,但是意义是不同的,两种运算的结果也不同。

取负运算只有一个操作数,因此它是单目运算符;减运算有两个操作数,因此它



是双目运算符。在一个算术表达式中,优先顺序为取负运算、乘除运算、加减运算。
例如: $-78+6*5$ 的结果为 -48 。

但是,如果有括号,则括号优先。例如: $9*(-20-30)$ 结果为 -450 。

(2)关系运算符

C++中的基本关系运算符如表 1.2 所示。

表 1.2 关系运算符表

运 算 符	说 明	举 例
<code>==</code>	等于	$34==56$ 结果为 false
<code>!=</code>	不等于	$100!=25$ 结果为 true
<code>></code>	大于	$32>12$ 结果为 true
<code><</code>	小于	$32<12$ 结果为 false
<code>>=</code>	大于等于	$54>=65$ 结果为 false
<code><=</code>	小于等于	$54<=65$ 结果为 true

关系运算符是双目运算符,运算的结果是 true 或 false。

(3)逻辑运算符

C++中的基本逻辑运算符如表 1.3 所示。

表 1.3 逻辑运算符表

运 算 符	说 明	举 例
<code>!</code>	非运算	$!(34==56)$ 结果为 true
<code>&&</code>	与运算	$(34==56)\&\&(100!=25)$ 结果为 false
<code> </code>	或运算	$(34==56) (100!=25)$ 结果为 true

逻辑运算符是双目运算符,运算的结果是 true 或 false。在一个逻辑表达式中,优先顺序为非运算、与运算、或运算。例如: $!(34==56)\&\&(100!=25)$ 结果为 true。

(4)按位运算符

对二进制位直接进行操作的运算符叫做按位运算符。C++中的基本按位运算符如表 1.4 所示。