免费提供电子教案

高等院校规划教材
计算机科学与技术系列

C++程序设计教程——基于Visual Studio 2008

刘 冰 张 林 蒋贵全 编 著

机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

C++程序设计教程

——基于 Visual Studio 2008

刘 冰　张 林　蒋贵全　编著

机械工业出版社

本书从实用的角度出发，详细介绍了 C++语言基础、面向对象的 C++语言程序设计、Windows 编程基础、Visual C++ 2008 开始平台、MFC 基本应用程序、用户界面设计、对话框和常用控件、文档和视图、图形与文本等知识，并介绍了 Visual C++ 2008 的高级应用以及开发实例。每章均配有习题，以指导读者深入地进行学习。本书内容丰富、通俗易懂，概念清晰、深入浅出，实例丰富、实用性强，对于 Visual C++ 2008 开发平台初学者，通过对本教材的学习可以熟练掌握操作并能够解决实际工程问题。

本书可作为高等学校计算机专业 C++语言程序设计课程的教材或教学参考用书，也可作为通信、电子信息、自动化等相关专业教材。

图书在版编目（CIP）数据

C++程序设计教程——基于 Visual Studio 2008 / 刘冰，张林，蒋贵全编著.
—北京：机械工业出版社，2009.8
（高等院校规划教材·计算机科学与技术系列）
ISBN 978-7-111-27700-2

Ⅰ.C… Ⅱ.①刘…②张…③蒋… Ⅲ.C 语言—程序设计—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2009）第 118462 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：唐德凯 谷玉春
责任印制：乔 宇

北京四季青印刷厂印刷（三河市杨庄镇环伟装订厂装订）

2009 年 8 月·第 1 版第 1 次印刷
184mm×260mm·20.5 印张·507 千字
0001－3000 册
标准书号：ISBN 978-7-111-27700-2
定价：34.00 元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294 68993821
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革。计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21 世纪高等院校计算机教材系列”、“21 世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21 世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术，以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时，本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

作为新一代开发工具，Visual Studio 2008 对编程语言、设计器、编辑器和数据访问功能进行了全面的提升，确保开发人员克服软件开发难题，快速创建互连应用程序。Visual Studio 2008 提供了一些新的工具，可使开发人员在最新的平台上快速地构建杰出的、高度人性化用户体验的和互联的应用。这些最新平台包括 Web、Windows Vista、Office 2007、SQL Server 2008、Windows Mobile 和 Windows Server 2008。Microsoft Visual Studio Team System 2008 提供完整的工具套件和统一的开发过程，适用于任何规模的开发团队，帮助所有团队成员提高自身技能，使开发人员、设计人员、测试人员、架构师和项目经理更好地协同工作，缩短软件或解决方案的交付时间。

本书不仅重视理论知识的介绍，并且将实践与理论相结合，在教学内容上也做了较大调整，着重对基于 Visual Studio 2008 的 Visual C++ 2008 开发平台的新增功能和特点进行理论和实例的讲解，使学生能更深刻地理解这种先进的程序设计思想。力求通过实例让学生全面掌握面向对象与可视化程序设计的思路和开发技巧。同时，为了让学生更好地掌握“Visual C++面向对象与可视化程序设计”的思想和方法，还增加了部分紧扣相关知识点的典型实例。本书涉及的知识点包括 C++语言基础、面向对象的程序设计、Windows 编程基础、Visual C++ 2008 开发平台概述、MFC 基本应用程序、用户界面设计、对话框和常用控件、文档和视图、图形与文本、Visual C++ 2008 的高级应用综合实例分析等。

本书的编者都是长期在高校从事软件教学的教师，有丰富的教学经验和科研开发能力。其中第 1、2、3、4 由刘冰编写；第 6、7、8、11 章由张林编写；第 5、9、10 章由蒋贵全编写；龙昭华教授负责统稿及内容审定，林远华、王少锋、郑幸福，王松勇、施佳、王波等参与了文字录入，并对书中的实例及图表做了大量的工作。本书的顺利完成要感谢重庆邮电大学计算机科学与技术学院的领导和相关老师给予的大力支持和帮助。

本书在编写过程中参阅了大量国内外有关 C++语言程序设计的教材和资料，在此向文献的作者表示感谢！

与本书配套的电子教案和习题答案将于本书正式出版后，向使用本教材的单位与个人提供，如有需要可在机械工业出版社网站（www.cmpedu.com）下载。

目前，国内外有关 Visual C++方面的资料很多，新理论、新技术层出不穷。由于时间仓促，加上 C++语言程序设计方法、设计思想发展迅速和编者水平有限，书中难免存在不妥和错误之处，恳请读者批评指正。

感谢您阅读本书。请将您的宝贵建议和意见发送至：jsjfw@mail.machineinfo.gov.cn。

编　者

目　录

第1章　C++语言基础

本章内容主要包括标识符和关键字、运算符和表达式、基本数据类型、常量和变量、数组、指针和引用、字符串处理函数、标准输入输出、函数等 C++语言中所涉及的相关基础理论知识。

1.1　标识符和关键字

1.1.1　标识符

标识符是一个字符序列，用来标识变量、函数、数据类型等。任何程序都离不开标识符。也就是说，不可能存在没有标识符的 C++语言程序。include、void、main、int、i、cin、cout 等都是标识符。

标识符可以由大写字母、小写字母、下画线（_）和数字（0～9）组成，但必须是以大写字母、小写字母或下画线（_）开头。在 C++语言程序中，大写字母和小写字母不能混用，比如 Name 和 name 就代表两个不同的标识符。表 1-1 给出了一些正确和不正确的标识符实例。

表 1-1　正确和不正确的标识符实例

正　确	不 正 确
samrt	5smart（不能以数字开头）
_decision	bomb?（有非法字符）
key_board	&Keyboard（有非法字符）

标识符的命名规则：

1）所有标识符必须由一个字母(a~z 或 A~Z)或下画线(_)开头。

2）标识符的其他部分可以用字母、下画线或数字(0~9)组成。

3）大小写字母表示不同意义，即代表不同的标识符，如 cout 和 Cout 表示不同的标识符。

在定义标识符时，虽然从语法上讲允许用下画线开头，但是，最好避免定义用下画线开头的标识符，因为编译器常常定义一些下画线开头的标识符。

C++语言的标识符经常用在以下情况中：

1）标识对象或变量的名字。

2）类、结构和联合的成员。

3）函数或类的成员函数。

4）自定义类型名。

5）标识宏的名字。

6）宏的参数。

1.1.2 关键字

在C++语言中，有一些预定义的标识符称为关键字，也可称为保留字，如int、void等都是关键字。关键字是一种特殊的标识符。关键字具有特定的含义，不能对它们再定义。例如，int、void在C++语言中被预定义为特定的数据类型，不能把它们再定义为变量的标识符。C++语言中关键字很多，除int和void两个外，标准C++语言中预定义了63个关键字。另外，还定义了11个运算符关键字，它们分别是and、and_eq、bitand、bitor、compl、not、not_eq、or、or_eq、xor 、xor_eq。

C++语言中的关键字：

asm	do	if	return	typedef
auto	double	inline	short	typeid
bool	dynamic_cast	int	signed	typename
break	else	long	sizeof	union
case	enum	mutable	static	unsigned
catch	explicit	namespace	static_cast	using
char	export	new	struct	virtual
class	extern	operator	switch	void
const	false	private	template	volatile
const_cast	float	protected	this	wchar_t
Continue	for	public	throw	while
default	friend	register	true	
delete	goto	reinterpret_cast	try	

另外，有些标识符虽然不是关键字，但C++语言中将其以固定的形式用于专门的地方，也不能当做一般标识符使用，以免造成混乱。这样的标识符有include、define等。

1.2 运算符和表达式

1.2.1 运算符

运算符是指运算的符号。例如，加法运算符（+）、乘法运算符（*）、取地址运算符（&）等。表达式与运算符密不可分，它由运算符与操作数组合而成，并由运算符指定对操作数要进行的运算，一个表达式的运算结果是一个值。

C++语言中的运算符是可以让C++语言编译器能够识别的具有运算意义的符号。编译器把这些符号及其组成的表达式翻译成相应的机器代码，就可以由计算机运行得出正确的结果。

下面给出由运算符组成的表达式的例子：

```
100+200-300*200+1000/20+100%10
a || b && c || d
```

```
a = b + c + d*e
a += b++
c -= d--
```

只要是按 C++语法写出的表达式，编译器就能够解释其中的运算符和由运算符、操作数组成的表达式的意义。

C++语言提供的运算符有以下几种：算术运算符、关系运算符、逻辑运算符、位运算符、条件运算符、赋值运算符、逗号运算符、sizeof 运算符及其他运算符(这是按功能分的)。不同的运算符，需要指定的操作数的个数并不相同。根据运算符需要的操作数个数，可将其分为 3 种：单目运算符（一个操作数）、双目运算符（两个操作数）和三目运算符（三个操作数）。下面介绍几种基本的运算符。

1．算术运算符

C++语言提供 5 种基本的算术运算符，见表 1-2。

表 1-2　5 种基本的算术运算符

运 算 符	名　字	实　例
+	加	12 + 4.9 // 得出 16.9
–	减	3.98 – 4 // 得出-0.02
*	乘	2 * 3.4 // 得出 6.8
/	除	9 / 2.0 // 得出 4.5
%	取余	13 % 3 // 得出 1

这 5 个算术运算符都是双目运算符。除“%”运算符外，其他算术运算符的两个操作数可以是整型（short int、int、long int、unsigned short int、unsigned int 或 unsigned long int）和实型（float 或 double）的混合类型，运算结果的数据类型是：两个操作数的数据类型中，具有较高级别的数据类型。例如，一个 int 型操作数和一个 float 型操作数的运算结果是 float 型；一个 float 型操作数和一个 double 型操作数的运算结果是 double 型。

在程序设计中，有时可以利用整数除法获得所需要的结果，但容易产生错误。如果两个操作数是整数，要获得实数除法，应当将两个或任一个整型操作数强制转换为实型数，例如：

```
int cost = 100;
int volume = 80;
double unitPrice = cost / (double) volume; // 得 1.25
```

执行除法运算时，如果除数为零，程序运行时，会产生一个被零除的错误。

取余运算符（%）的两个操作数都必须是整型数，运算结果是整除后的余数。例如，13%3 的结果是 1。

算术运算的结果可能太大，而不能存储在一个指定的变量中，这种情形称之为溢出。例如：

```
unsigned char k = 10 * 92; // 溢出: 920 > 255
```

在进行除法运算时，需要注意右操作数，即除数应为非零值，否则就会产生浮点运算错

误。编程时，通常用 if 语句判别除数是否为 0，例如：

```
if (a != 0) // 只有 a 不等于 0 时才执行下面的操作
c = b / a;
```

上面所有的运算符都是双目运算符，左右操作数可以是数值，也可以是表达式。根据左右操作数类型的不同会产生不同的结果。例如，9/4 = 2，而 9/4.0 = 2.5、9.0/4 =2.5、9.0/4.0 = 2.5。

另外，整型的数值中有不同的类型，如字符型、短整型、长整型。它们之间的类型转换需要特别的注意。例如，一个无符号字符型的变量。

```
unsigned char a, b;
a = 90*10; // 结果 a 不等于 900，而等于 132
```

其实就是把 900 这个整数转化为无符号字符型数的结果，即(unsigned char)900 的结果为 132。

```
b = 200+100; // 结果 b 不等于 300，而等于 44
```

它们之间的优先级关系是：运算符“*”、“%”优先级相同，且优先于运算符“+”和“–”。低级运算符先于高级运算符计算时，用括号“(”、“)”表示其优先关系。例如：

```
(100+20) – (20 – 30) *40
```

需要注意的是，当这些符号出现在字符串中时，就被计算机的编译器处理成字符串中的字符，而不是运算符。例如，字符串"ab+cd-def"中的“+”和“0”就只是代表两个字符而已。而且，“+” 、“–”运算符也可以作为单目运算符使用，如“+100”、“–100”分别表示正的 100 和负的 100。当它们作为单目运算符时，运算优先级较高。

2．自增、自减运算符

“＋＋”是自增运算符，“—”是自减运算符，这两个运算符都是单目运算符，且功能相近，都是将数值变量的值加 1 或减 1，用户只能将这类运算符应用于变量而不能应用于常量。要替代下列代码：

```
valuel=valuel+1;
```

可使用“++valuel;”或“valuel++;”语句。

这里前一种方式称为前缀方式，后一种称为后缀方式，其目的都是使 valuel 加 1。两者的区别是：前缀式先将操作数增 1（或减 1），然后取操作数的新值参与表达式的运算；后缀是先将操作数增 1（或减 1）之前的值参与表达式的运算，到表达式的值被引用之后再做加 1（或减 1）运算。

自增、自减运算符见表 1-3，假定变量 valuel 已预定义：int valuel = 5;

表 1-3　自增和自减运算符

运　算　符	名　　字	实　　例
++	自增(前缀)	++valuel + 10 // 得出 16，valuel 变为 6
++	自增(后缀)	valuel++ + 10 // 得出 15，valuel 变为 6
--	自减(前缀)	—valuel + 10 // 得出 14，valuel 变为 4
--	自减(后缀)	valuel— + 10 // 得出 15，valuel 变为 4

从表中可以看出，自增和自减运算符可在变量名前，也可在变量名后，即都可以用于前缀和后缀的形式，但含义并不相同。对于前缀的形式，变量先做自增或自减运算，然后将运算结果用于表达式中；而对于后缀的形式，变量的值先在表达式中参与运算，然后再做自增或自减运算。

可以做自增或自减运算的变量类型可以是实型，但实型变量自增或自减运算的用处不大，很少使用。通常，对整型变量做自增或自减运算。同时指针变量也可以做自增或自减运算。

需要注意的是，自增运算符在操作数的前面和后面是不一样的，例如：

```
a = 6;
b = a++;
```

执行结果为：b=6, a=7。

而

```
a = 6;
b = ++a;
```

执行结果为：b=7, a=7。

自减运算符也存在同样情况，例如：

```
a=6;
b = a--;
```

执行结果为：b=6, a=5。

而

```
a=6;
b=--a;
```

执行结果为：b=5, a=5。

3. 关系运算符

C++语言提供 6 种关系运算符，用于数值之间的比较，表达式的值或为 1（表示 true），或为 0（表示 false），见表 1-4。

表 1-4　关系运算符

运　算　符	名　　字	实　　例
==	等于	5 == 5 // 得出 1
!=	不等于	5 != 5 // 得出 0
<	小于	5 < 5.5 // 得出 1
<=	小于或等于	5 <= 5 // 得出 1
>	大于	5 > 5.5 // 得出 0
>=	大于或等于	6.3 >= 5 // 得出 1

“<=”和“>=”运算符不能写成“=<”和“=>”，“=<”和“=>”是无效的运算符。关系运算符的操作数应当是一个数值，字符是有效的操作数，因为它们是用数值来表示的。例如（假定采用 ASCII 编码）：

```
'A' < 'F'    // 得出 1 (它等价于 65 < 70)
```

字符串不应当用关系运算符比较，因为被比较的不是字符串的内容本身，而是字符串的地址。例如：

```
"HELLO"< "BYE"
```

引起“HELLO”的地址与“BYE”的地址进行比较。由于字符串的地址是由编译器决定的，所以表达式的结果或为0，或为1，并不确定。以后会看到，可以用C++语言的库函数strcmp，比较两个字符串。

由关系运算符组成的关系表达式的值是逻辑型的，即bool 型。在C++语言中常常将逻辑真用非0表示（一般为1），逻辑假用0表示。

其中：

“==”用于判断其前后数值或表达式的结果a和b是否相等。例如，a为2001，b为667*3，则表示a等于b，结果为真。

“!=”用于判断其前后数值或表达式的结果a和b是否不相等。例如，a为2002，b为667*3，则表示a不等于b，结果为真。

“<”用于判断其前后数值或表达式的结果a是否小于b。例如，a为2002，b为667*3，则表示a小于b，结果为假。

“<=”用于判断其前后数值或表达式的结果a是否小于或等于b。例如，a为2002，b为667*3，则表示a小于或等于b，结果为假。

“>”用于判断其前后数值或表达式的结果a是否大于b。例如，a为2002，b为667*3，则表示a大于b，结果为真。

“>=”用于判断其前后数值或表达式的结果a是否大于或等于b。例如，a为2002，b为667*3，则表示a大于或等于b，结果为真。

关系运算符组成的表达式，一般是和逻辑运算符组合用在条件表达式中的。条件表达式用于程序的分支处理。

1.2.2 优先级和结合性

当不同的运算符混合运算时，运算顺序是根据运算符的优先级而定的，优先级高的运算符先运算，优先级低的运算符后运算。在一个表达式中，如果各运算符有相同的优先级，运算顺序是从左向右，还是从右向左，是由运算符的结合性确定的。所谓结合性是指运算符可以和左边的表达式结合，也可以与右边的表达式结合。C++语言运算符的优先级和结合性见表1-5。

表1-5 C++语言运算符的优先级和结合性

优先级	运算符						种类	结合性
最高	::（全局）						单	从右到左
最高	::（类域）						双	从左到右
	()(括号及函数调用)			->	.[]		双	从左到右
	+ -	++ --	! ~	* &	new delete	Sizeof (类型)	单	从右到左
	->*		.*				双	从左到右

（续）

优 先 级	运 算 符	种 类	结 合 性
	+ / %	双	从左到右
	+ —	双	从左到右
	<< >>	双	从左到右
	< <= > >=	双	从左到右
	== !=	双	从左到右
	&	双	从左到右
	^	双	从左到右
	|	双	从左到右
	&&	双	从左到右
	||	双	从左到右
	?:	三	从左到右
	=	双	从右到左
最低	,	双	从左到右

注：“单”表示是单目运算符；“双”表示是双目运算符；“三”表示是三目运算符。

每种运算符都有一个优先级，优先级是用来标志运算符在表达式中的运算顺序的。优先级高的先做运算，优先级低的后做运算，优先级相同的由结合性决定计算顺序。

大多数运算符都是按从左到右计算，只有 3 类运算符的结合性是从右到左。它们是：单目、三目和赋值。

不同类型的运算符混合使用时，即使实现优先级与实际需要相符，最好也使用括号隔离，以便代码更加清晰。当然，在实现优先级与实际需要不相符时，更需要使用括号来改变。

例如，本来是正确的代码：

```
if (year%4==0 || year%100!=0 &&year%400==0)
```

如果加上括号，则更加清晰，提高了程序的可读性和可维护性。

```
if（(year%4)==0 ||（year%100）!=0 &&（year%400）==0）
```

1.2.3 表达式

表达式是由运算符和操作数组成的式子。运算符可以是前面介绍过的运算符，操作数包含了常量、变量、函数和其他一些命名的标识符。最常见的表达式是常量和变量。

在表达式中，连续出现两个运算符时，最好用空格符分隔。例如：

```
a+++b;  //在 C++ 2008 语言中这种写法是错误的，编译将不能通过
```

系统将默认为是“a++ +b”，因系统将按尽量取大的原则来分割多个运算符。如果想执行 a 加++b，则应写成“a+ ++b;”。

任何表达式经过计算都应有一个确定的值和类型。在计算一个表达式的值时，应注意下述两点：