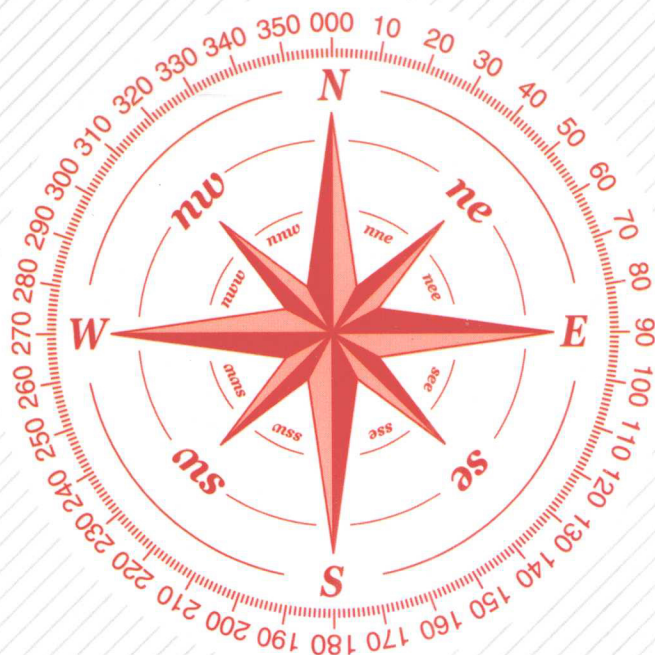


从零开始学系列

近14小时详细的教学视频
本书源代码+精美教学PPT



理论知识 + 核心技术 + 精彩示例 + 实战案例 + 完整视频 + 教学PPT = 编程高手

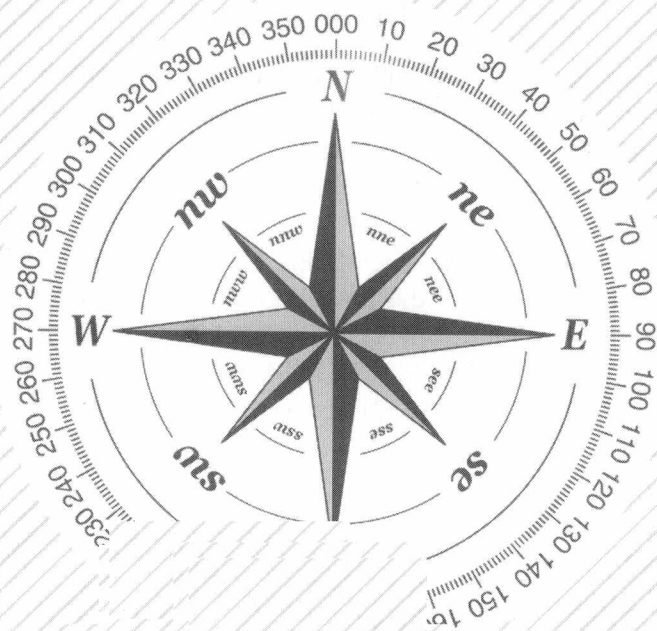
从零开始 学C++

- 以实际应用为出发点，全面细致地讲解了C++程序语言的整个知识体系
- 理论联系实际，视频和PPT相辅，快速提高读者开发水平
- 语言生动，通俗易懂，示例技巧丰富，针对性强

梁伟 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

从零开始学系列



从零开始 学C++

梁伟 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书分为3篇,第1篇介绍C++基础知识,包括Visual C++ 6.0集成开发环境、C++程序设计基础知识,以及数组、函数、指针和预处理机制等方面的知识。第2篇介绍C++语言面向对象编程的相关知识,包括封装、继承、多态、虚函数、函数重载、文件输入/输出流及C++数据库方面的基础知识等。第3篇从应用的角度出发,介绍教学管理系统实例程序,通过实例程序的开发,使用户能够巩固本书中所介绍的C++基础知识。

全书注重对读者进行C++语言基础知识及良好编程风格的培养,不仅可以方便读者深入学习MFC的相关内容,还对学习其他程序设计语言有一定的帮助。虽然各种计算机语言不同,但是程序设计的基本思想是相同的。同时,本书也着重于培养读者的程序设计思想,以及良好的编程习惯。读者可以直接使用本书中的部分实例程序进行实际开发。

本书适合C++的初学者和有一定C++程序设计经验的读者,同时也是一本不可多得的项目开发参考书籍。

图书在版编目(CIP)数据

从零开始学C++ / 梁伟编著. --

出版社, 2011.1

ISBN 978-7-113-11915-7

I. ①从… II. ①梁… III. ①C语言—程序设计

IV. ①TP312

中国版本图书馆CIP数据核字(2010)第179252号

书 名: 从零开始学C++

作 者: 梁 伟 编著

责任编辑: 韩中领

编辑助理: 李维芳

封面设计: 王晓武

责任印制: 李 佳

读者热线电话: 400-668-0820

特邀编辑: 李新承

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 北京市昌平开拓印刷厂

版 次: 2011年1月第1版

2011年1月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 26 字数: 603千

印 数: 3 000册

书 号: ISBN 978-7-113-11915-7

定 价: 56.00元(附赠光盘)

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换。

前言

C++语言是当前非常流行的一门编程语言。该语言不仅支持面向过程的编程方法，还支持面向对象的编程方法。本书将全面细致地介绍 Visual C++ 6.0 开发环境下的 C++编程技术。

笔者精心编写本书，目的是使初学者不只停留在了解理论知识的基础上，还能够真正地动手进行 C++程序的编写。为了帮助初学者尽快掌握编程的技术，扩展自己的知识面，笔者在书中使用了大量的实例程序，进行简单明了的讲述，尽量使读者少走弯路，同时也能够在实际开发中作为参考实例来使用。

本书特点

本书知识丰富，内容全面。介绍了从 C++入门基础知识到 C++的基本语法、面向过程编程方法到面向对象编程技术的过渡，从控制台程序设计到 VC++的可视化界面设计等知识。同时也介绍了 C++语言中的一些重要特性，如封装、继承、多态，以及虚函数等相关的知识点。

本书的特点主要体现在以下几个方面。

1. 内容丰富，结构合理

本书结构安排合理，内容丰富，并且针对初学者对语言本身和常用功能函数缺乏了解的特点，着重介绍了 C++的语法基础。同时也针对具有少量经验的 C++程序员，在书中使用了大量的实例程序，使其可以直接从部分实例程序中获得所需要的参考信息。效果图如下：

第 1 章 C++轻松入门

C++是在 C 语言基础上扩展而成的一门面向对象编程的计算机语言。其中面向对象编程的许多特性都是非常实用的。例如，用户可以对目标对象进行封装、继承等以满足面向对象编程的需要。在本章中，主要的学习内容有：

- 关于 C++的相关基础知识；
- C++的常用关键字；
- 如何使用微软公司提供的编译工具；
- 如何使用编译器对 C++程序进行编译和调试等。

1.1 C++基础知识

C++语言延续了 C 语言的基本结构，但是也进行了相关的优化。例如，在 C++语言中，增加了面向对象编程的相关特性，如封装、继承和多态等。这些特性增强了 C++程序代码的可读性，并且提高了程序的运行效率。

1.1.1 面向对象编程

面向对象编程的基本思想是程序员将所需要的客观存在于现实世界中的特定实体（即对象）从现实世界中抽象出来进行软件系统的构造。在整个软件系统的构造中，将抽象出来的对象使用逻辑思维进行操作。在面向对象编程中，有几个非常重要的特性，分别如下：

1. 封装

将一组或多组相关的数据组合起来，形成一个抽象的类，叫做封装。而这个类的一个实例其实就是对象。组成该对象的数据则为该对象的相关属性。例如，一个完整的人是由鼻子、耳朵等部位组成的。那么，用户便可以将这些组成部位看做是一组相关联的数据组合成的类，而人则是这个类的实例对象。

2. 继承

继承是指在 C++语言中，两个单独的类之间所发生的联系，即一个类具有另一个类的全部或部分属性。例如，两个独立的人 A 和 B，A 的基因中包含了 B 基因中的全部或部分基因，则用户可以认为 A 是继承于 B 的。此时，A 称为子类，B 称为父类或基类。用户在程序中使用继承，可以避免重复定义相关类的属性，使类定义达到最简化。

3. 多态

当父类中以特殊形式进行声明的函数或属性被子类继承后，这些函数根据子类对象调用的不同方式而表现出不同的反应形式，称为多态性。例如，人类和家畜同属于动物类，但是人类的手可以用来拿东西、打字等，而家畜却不能实现这些功能。

4. 虚函数

虚函数是多态性编程的基础，其具体是指在基类中使用 C++关键字 virtual 进行声明的函数。

从入门开始学 C++

2. 输入代码

当用户创建控制台项目成功后，便可以将相应的代码输入到编译器中了。

【实例 1.4】由于控制台程序可以接收用户的输入，所以本小节将以加法计算程序为实例。用户在控制台程序添加一个 C++源文件，名称为 Cpp1。具体代码如下所示：

【本例代码位置：源文件\01\实例 1.4\1.4.cpp】

```
01 #include<iostream.h> //包含标准输入/输出头文件
02 main()
03 {
04     int a,b,c; //定义3个整型变量
05     cout<<"请输入两个加数"<<endl; //接收用户的输入
06     cin>>a;
07     cin>>b;
08     c=a+b;
09     cout<<"计算结果为："<<c<<endl; //输出计算结果
10     return 0; //程序正常结束
11 }
```

【运行程序】用户将上面的代码输入到 VC++编译器中，保存并编译和执行后，可以看到控制台程序接收用户的输入并进行计算，如图 1.41 所示。

【深入学习】在第 06 和 07 行代码中，用户可以将代码 `cin>>a;cin>>b;` 替换为 `cin>>a>>b;`，也可以实现相同的功能。在本小节中，以加法计算程序为例向用户介绍了如何在 VC++编译器中创建控制台程序，以及编译控制台程序等相关操作。

图 1.41 控制台程序接收用户输入

提示：用户在空的控制台程序中必须使用“文件”->“新建”命令，为基添加一个 C++源文件，否则程序编译通过后不能成功执行。

1.4 小结

本章主要向用户介绍了 C++语言的基础知识和基本结构，并着重讲解了 VC++编译器的使用方法及 C++程序的调试方法。最后，本章结合编译器与 C++语言编写了两个实例程序向用户分别讲解了如何在 VC++编译器中创建项目、编写程序、添加 C++源文件，以及调试程序等相关操作。用户可以将随书光盘中的相关程序根据书中的操作步骤进行修改和调试，以便达到学习和掌握的效果。

1.5 习题

一、选择题

1. 比较经典的 C++语言编译器是：_____。
- A. Turbo C B. Visual C++ 6.0 C. Visual Studio 2005 D. Visual Studio 2008

2. 图例及代码丰富，实用性强

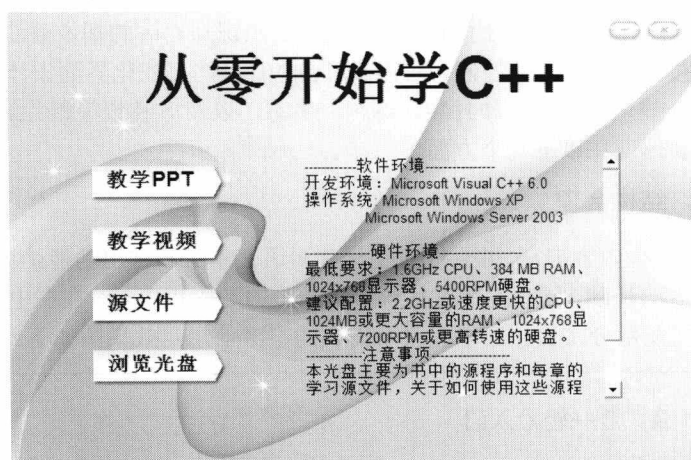
全书提供了大量的图例及代码说明，以图文并茂的形式帮助读者快速掌握 C++ 的基础知识。同时，读者还可以参考本书的实例实现更多的功能。

3. 实例程序功能具体、全面、深入

本书中所介绍的实例程序都是经过反复编译后确定的。这些实例程序将与书中对应的知识点进行紧密的结合，力求全面、具体地向读者介绍这些重要的知识点，并试图通过实例程序的运行，使读者能够深入理解并学习 C++ 语言。

4. 完善的多媒体教学视频及 PPT 课件

良好的人机互动界面，我们尽量简化操作，力求使读者轻松地达到所要达到的目的。读者将本书随书光盘放入光驱后，即可进入本张光盘运行主界面，如下图所示。



我们聘请了专业老师，精心地录制了长达 14 小时的教学视频，对书中的重点知识和实例，亲自演示说明，以达到使读者轻松掌握理解的目的。

5. 精心组织的源代码

源代码按照图书章节依次排列，单击即可进入对应的文件夹。



本书内容安排

本书分为3篇,共15章。首先介绍了C++语言基础知识,再进一步介绍开发工具VC++ 6.0的操作方法及所提供的控件基础知识,并通过实例程序介绍相应的调试技术,然后再对C++语言的一些高级应用进行讲解。最后,通过完整的实例项目,向读者讲解开发VC++实例项目的一般步骤,使编程人员的编程水平得到不断地提高。

第一篇(第1章~第6章)入门篇。

在第一篇中,讲解了C++语言的基础知识,包括C++语言的基本语法、C++语言中的变量与常量、表达式与运算符、字符、流程控件语句、数组、指针、函数,以及预处理机制等。C++语言的初学者能够通过这6章的基础知识学习,可以了解C++语言,并为继续深入学习C++语言打下坚实的基础。

第二篇(第7章~第14章)提高篇。

在这一篇中,讲解了C++语言中面向对象编程的相关特性。包括C++语言中的类封装方法、C++类之间的继承方式、如何利用虚函数实现C++多态性、运算符重载的方法,以及C++输入/输出流、C++数据库编程基础等相关的知识。通过这一篇的学习,读者能够学习并掌握C++语言的核心技术,并快速提高编程能力及数据库操作能力。

第三篇(第15章)综合实例篇。

通过编写C++实例程序——教学管理系统,向读者介绍如何使用C++语言实现教学管理系统的基本功能及C++自定义类的编写方法等。在本篇中,读者将学习到VC++ 6.0开发工具的使用方法,以及MFC类库函数的作用及其调用方法。通过这一篇的学习,并结合前面所学习的C++基础知识,更深入理解C++面向对象编程的思想。

本书内容由浅入深,由理论到实践,尤其适合初级读者逐步学习和完善自己的知识结构。

适合阅读本书的读者

- 希望学习C++编程的初学者。
- 迫切希望提高个人编程技能和水平的初级C++程序员。
- 学习Visual C++ 6.0开发工具的技术人员。
- 具备一定C++语言基础知识,但是缺乏实战经验的程序员。

本书作者

本书主要由梁伟编写,其他参与编写的人员有陈强、陈燕、丁凤霞、丁礼友、范忠诚、黄俊灿、贾伟、李喜彤、林垚、尚文谊、孙亮亮、唐崇敏、陶则熙、王龙、王拥东、吴善才、徐砚颖、尹健慧、詹涵林、张薇、张小强、张运端、赵玉荣、郑慧、朱博、朱朝坤、邹小红。

由于时间仓促,加之水平有限,书中难免存在缺点和不足之处,敬请读者批评指正。如果读者在阅读本书的时候出现任何疑问,还可以发送电子邮件及时与我们取得联系,我们会尽快给予答复。我们的邮箱是:HappySants@163.com 或 zhangshuqiangbooks@163.com。

编者
2010年8月

目 录

第 1 篇 入 门 篇

第 1 章 C++轻松入门 ( 教学视频: 49 分钟)	2
1.1 C++基础知识	2
1.1.1 面向对象编程	2
1.1.2 C++程序的基本结构	3
1.2 C++开发工具介绍	4
1.2.1 安装 Visual C++集成开发环境	4
1.2.2 启动 Visual C++编译器	8
1.2.3 Visual C++菜单详解	9
1.2.4 Visual C++的常用工具栏	11
1.2.5 Visual C++应用程序向导	12
1.2.6 程序调试方法	13
1.3 C++程序实例	14
1.3.1 Hello World 程序实例	15
1.3.2 创建简单的控制台程序	16
1.4 小结	17
1.5 习题	17
第 2 章 C++数据类型介绍 ( 教学视频: 50 分钟)	19
2.1 C++基本数据类型	19
2.1.1 常用的基本数据类型	19
2.1.2 结构体	27
2.1.3 联合体	29
2.2 C++数据的仓库——数组	30
2.2.1 什么是数组	30
2.2.2 初始化数组	32
2.2.3 使用下标表示数据顺序	33
2.2.4 使用二维数组装载数据	36
2.2.5 使用字符数组保存字符串	40
2.3 C++常用运算符	42
2.3.1 算术运算符	42
2.3.2 逻辑运算符	45
2.3.3 C++程序实例	48
2.4 小结	49
2.5 习题	50
第 3 章 C++程序控制语句 ( 教学视频: 75 分钟)	51
3.1 C++程序基本结构	51

3.1.1	预处理	51
3.1.2	主函数 main	54
3.1.3	输入/输出	56
3.2	循环语句	57
3.2.1	while 循环	57
3.2.2	do...while 循环	59
3.2.3	for 循环	61
3.2.4	嵌套使用循环结构	62
3.3	选择语句	64
3.3.1	if 选择结构	64
3.3.2	if...else 选择结构	65
3.3.3	switch 选择结构	67
3.4	输出控制语句	69
3.4.1	常用输出控制符	69
3.4.2	格式化数据输出实例	73
3.5	小结	75
3.6	习题	75
第 4 章	指针 (教学视频: 75 分钟)	77
4.1	指针与地址	77
4.1.1	指针与地址的基本概念	77
4.1.2	指针与地址的区别	78
4.2	指针变量	78
4.2.1	定义指针变量	78
4.2.2	初始化指针变量	79
4.2.3	指针与地址之间的转换	80
4.2.4	引用指针变量	80
4.3	数组与指针	81
4.3.1	操作空指针	82
4.3.2	操作一维数组指针	83
4.3.3	操作字符数组指针	86
4.3.4	操作字符串数组指针	90
4.3.5	操作结构体指针	93
4.4	函数与指针	98
4.4.1	数组指针作为函数参数	99
4.4.2	字符串指针作为函数参数	108
4.4.3	函数返回值作为指针类型	111
4.5	指针实例程序	115
4.6	小结	117
4.7	习题	117
第 5 章	C++功能的实现——函数 (教学视频: 47 分钟)	118
5.1	函数的基本概念	118
5.2	如何自定义函数	118
5.2.1	函数声明	119
5.2.2	函数实现	120

5.3	函数之间的参数传递及返回值.....	123
5.3.1	函数形参	123
5.3.2	函数实参	124
5.3.3	函数返回值	124
5.3.4	在函数之间传递参数实例	125
5.4	函数调用	126
5.4.1	函数的调用方式	126
5.4.2	函数的嵌套调用	127
5.4.3	递归函数	129
5.4.4	函数的作用域	130
5.5	内联函数	131
5.5.1	定义内联函数	131
5.5.2	调用内联函数	132
5.6	小结	134
5.7	习题	135
第 6 章	预处理机制 (教学视频: 47 分钟)	136
6.1	常用宏介绍	136
6.1.1	定义宏 define	136
6.1.2	标识符定义宏 typedef	138
6.2	宏定义	140
6.2.1	具有参数的宏定义	140
6.2.2	无参数的宏定义	142
6.3	包含相关文件	144
6.3.1	包含头文件	144
6.3.2	链接库文件	145
6.4	条件预编译格式	153
6.4.1	#ifdef...#else...#endif 格式	154
6.4.2	#ifndef...#else...#endif 格式	156
6.4.3	#if...#else...#endif 格式	157
6.5	实例程序分析	159
6.6	小结	163
6.7	习题	163

第 2 篇 提 高 篇

第 7 章	类 (教学视频: 56 分钟)	166
7.1	C++新特性	166
7.1.1	封装	166
7.1.2	抽象	168
7.1.3	继承	169
7.1.4	多态	173
7.2	C++类成员	176
7.2.1	成员变量	176
7.2.2	对象初始化	178

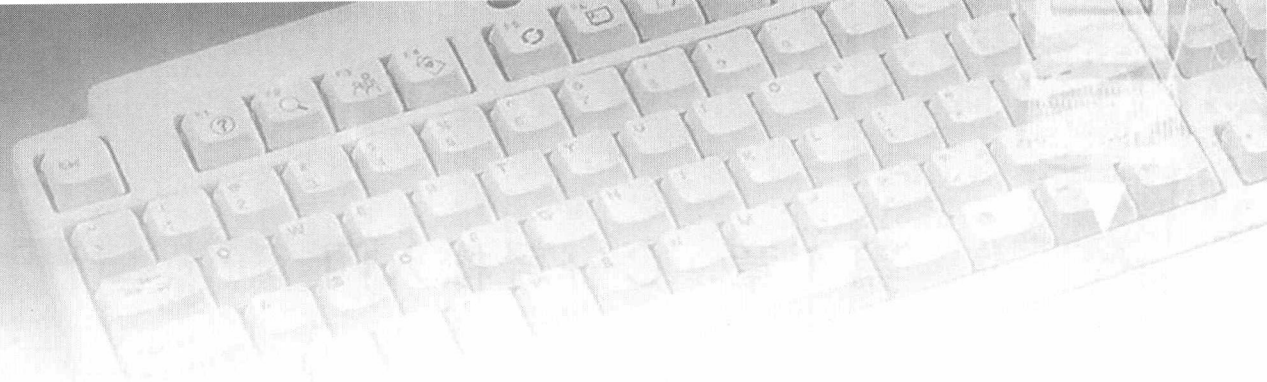
7.2.3	复制对象（拷贝构造函数）	185
7.2.4	this 指针	187
7.2.5	自定义类实例	188
7.3	创建类的指针实例对象	192
7.3.1	创建类指针对象	192
7.3.2	访问对象成员	192
7.3.3	删除类指针对象	193
7.3.4	对象的生存周期	193
7.3.5	指针对象实例程序	194
7.4	C++类的成员函数	195
7.4.1	定义成员函数原型	196
7.4.2	实现成员函数功能	196
7.5	小结	198
7.6	习题	199
第 8 章	封装 (教学视频: 42 分钟)	200
8.1	封装成员	200
8.1.1	设置成员访问控制	200
8.1.2	数据成员	204
8.1.3	函数声明	206
8.2	访问类中的数据	207
8.2.1	访问类中的私有成员	208
8.2.2	友元函数	210
8.2.3	引用实例对象	212
8.3	C++封装实例	215
8.4	小结	217
8.5	习题	217
第 9 章	C++功能的延续——继承 (教学视频: 58 分钟)	219
9.1	继承的相关知识点	219
9.1.1	继承的基本概念	219
9.1.2	继承的意义	221
9.1.3	继承的实现	221
9.2	基类和派生类	222
9.2.1	基类	222
9.2.2	派生类	223
9.3	继承方式	224
9.3.1	公共继承方式	224
9.3.2	私有继承方式	225
9.3.3	保护继承方式	226
9.4	初始化数据成员	227
9.4.1	基类和派生类的初始化顺序	227
9.4.2	多重继承中的初始化顺序	228
9.4.3	实例代码分析	230
9.5	创建子类实例对象	232
9.5.1	派生子类	232

9.5.2 指定子类的继承方式	233
9.5.3 创建子类对象	233
9.5.4 删除子类对象	235
9.6 解决继承中的成员函数重名问题	236
9.6.1 成员函数重载	236
9.6.2 成员函数覆盖	237
9.6.3 成员函数隐藏	239
9.7 实例程序分析	241
9.8 小结	243
9.9 习题	243
第 10 章 虚函数与多态性 ( 教学视频: 45 分钟)	245
10.1 虚函数	245
10.1.1 静态联编	245
10.1.2 动态联编	247
10.1.3 虚基类	250
10.1.4 虚析构函数	253
10.1.5 虚函数实例程序	255
10.2 抽象类	257
10.2.1 纯虚函数	257
10.2.2 创建抽象类指针对象	258
10.2.3 抽象类派生新类	260
10.2.4 实例分析	262
10.3 多态性	266
10.3.1 继承多态性接口	266
10.3.2 实现多态性接口	267
10.4 小结	269
10.5 习题	269
第 11 章 运算符重载 ( 教学视频: 66 分钟)	271
11.1 C++运算符	271
11.1.1 C++常用运算符	271
11.1.2 运算符重载基础	272
11.1.3 运算符重载的优点	272
11.2 运算符重载为成员函数和友元函数	273
11.2.1 运算符重载为成员函数	273
11.2.2 运算符重载为友元函数	276
11.3 单目运算符重载	278
11.3.1 前置单目运算符	278
11.3.2 后置单目运算符	281
11.3.3 运算符小结	284
11.4 双目运算符重载	284
11.4.1 双目运算符重载函数	284
11.4.2 实例分析	285
11.5 赋值运算符重载	286
11.5.1 重载 new 运算符	286

11.5.2	重载 delete 运算符	288
11.6	重载运算符++和--实例	289
11.7	小结	290
11.8	习题	291
第 12 章 C++输入/输出流 (教学视频: 70 分钟)		291
12.1	输入/输出流的使用准备	291
12.1.1	相关头文件	291
12.1.2	输入/输出流类	293
12.2	输入流	296
12.2.1	输入流读取运算符	296
12.2.2	获取输入流	298
12.2.3	实例程序分析	299
12.3	输出流	300
12.3.1	输出流插入运算符	300
12.3.2	使用函数实现流输出	301
12.3.3	实例程序分析	303
12.4	格式化输入/输出流函数	305
12.4.1	简介	305
12.4.2	格式状态标志介绍	305
12.4.3	显示小数点	308
12.4.4	对齐方式	308
12.4.5	设置填充字符	309
12.4.6	设置输出宽度	310
12.4.7	设置和清除格式标志	311
12.5	文件输入/输出流	312
12.5.1	创建文件输入/输出流对象	312
12.5.2	打开文件	314
12.5.3	格式化输出文件	315
12.5.4	格式化输入文件	316
12.5.5	文件输入/输出流实例分析	317
12.6	实例程序分析	319
12.7	小结	320
12.8	习题	321
第 13 章 程序运行时的异常处理 (教学视频: 42 分钟)		322
13.1	异常处理机制基础	322
13.1.1	异常处理机制	322
13.1.2	使用 C++异常处理结构	323
13.1.3	简单实例程序	323
13.2	异常处理步骤	325
13.2.1	发现异常	325
13.2.2	抛出异常	326
13.2.3	捕获异常	326
13.2.4	多次处理异常	326
13.3	实例程序分析	327

13.4	小结	328
13.5	习题	329
第 14 章	C++数据库基础 (教学视频: 55 分钟)	330
14.1	数据库的基础知识	330
14.1.1	使用 ODBC 开发数据库	330
14.1.2	使用 ADO 开发数据库	331
14.1.3	数据库相关操作小结	331
14.2	使用 ODBC 开发数据库的操作方法	331
14.2.1	创建数据库	331
14.2.2	配置 ODBC 数据源	336
14.2.3	与 ODBC 相关的 C++类	338
14.2.4	ODBC 小结	340
14.3	使用 ADO 开发数据库的操作方法	341
14.3.1	添加 ADO 控件	341
14.3.2	导入 ADO 的动态链接库	343
14.3.3	初始化 ADO 控件环境	343
14.3.4	连接数据库	346
14.3.5	操作数据库	346
14.3.6	ADO 数据库开发实例	350
14.4	小结	355
14.5	习题	355
第 3 篇 综合实例篇		
第 15 章	教学管理系统的 C++实现 (教学视频: 67 分钟)	358
15.1	项目概述	358
15.1.1	概述	358
15.1.2	实例中的相关类	358
15.1.3	实例程序的相关功能	359
15.2	自定义类	360
15.2.1	学生类	360
15.2.2	教师类	360
15.2.3	课程类	361
15.2.4	文件管理类	361
15.2.5	自定义类的继承关系	362
15.3	实现自定义类	363
15.3.1	新建 C++头文件和实现文件	363
15.3.2	实现类功能	367
15.4	使用自定义类	376
15.4.1	复制自定义类文件到工程目录	376
15.4.2	包含类的头文件	377
15.4.3	创建类实例对象	378
15.4.4	调用对象成员完成相应功能	379
15.5	制作个性化系统界面	379

15.5.1	Visual C++控件简介	379
15.5.2	制作个性化的系统界面	379
15.6	在控件消息响应函数中使用自定义类功能	385
15.6.1	为控件添加消息响应函数	385
15.6.2	为界面中的控件关联变量名	386
15.6.3	在消息响应函数中调用类成员	387
15.7	使用文件保存数据	391
15.7.1	创建文件实例对象	391
15.7.2	打开文件	391
15.7.3	格式化读取文件	392
15.7.4	格式化写入文件	393
15.7.5	关闭文件	394
15.7.6	实例程序	395
15.8	小结	400
15.9	习题	401



第 1 篇 入 门 篇

本篇包括：

第 1 章 C++轻松入门

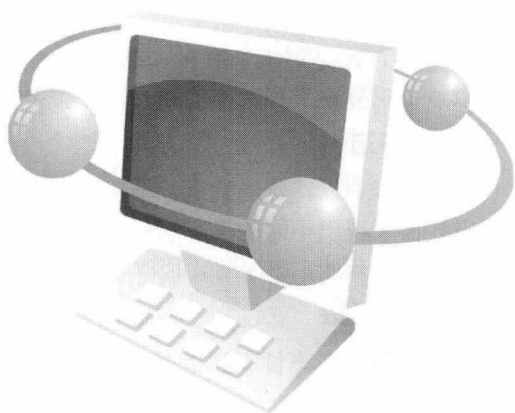
第 2 章 C++数据类型介绍

第 3 章 C++程序控制语句

第 4 章 指针

第 5 章 C++功能的实现——函数

第 6 章 预处理机制



第 1 章 C++轻松入门

C++是在 C 语言基础上扩展而成的一门面向对象编程的计算机语言。其中面向对象编程的许多特性都是非常实用的。例如，用户可以对目标对象进行封装、继承等以满足面向对象编程的需要。在本章中，主要的学习内容有：

- 关于 C++的相关基础知识；
- C++的常用关键字；
- 如何使用微软公司提供的编译工具；
- 如何使用编译器对 C++程序进行编译和调试等。

1.1 C++基础知识

C++语言延续了 C 语言的基本结构，但是也进行了相关的优化。例如，在 C++语言中，增加了面向对象编程的相关特性，如封装、继承和多态等。这些特性增强了 C++程序代码的可读性，并且提高了程序的运行效率。

1.1.1 面向对象编程

面向对象编程的基本思想是程序员将所需要的客观存在于现实世界中的特定实体（即对象）从现实世界中抽象出来进行软件系统的构造。在整个软件系统的构造中，将抽象出来的对象使用逻辑思维进行操作。在面向对象编程中，有几个非常重要的特性，分别如下：

1. 封装

将一组或多组相关的数据组合起来，形成一个抽象的类，叫做封装。而这个类的一个实例其实就是对象，组成该对象的数据则为该对象的相关属性。例如，一个完整的人是由鼻子、耳朵等部位组成的。那么，用户便可以将这些组成部位看做是一组相关联的数据组合成的类，而人则是这个类的实例对象。

2. 继承

继承是指在 C++语言中，两个单独的类之间所发生的联系，即一个类具有另一个类的全部或部分属性。例如，两个独立的人 A 和 B，A 的基因中包含了 B 基因中的全部或部分基因，则用户可以认为 A 是继承于 B 的。此时，A 称为子类，B 称为父类或基类。用户在程序中使用继承，可以避免重复定义相关类的属性，使类定义达到最简化。

3. 多态

当父类中以特殊形式进行声明的函数或属性被子类继承后，这些函数根据子类对象调用的不同方式而表现出不同的反应形式，称为多态性。例如，人类和家禽同属于动物类，但是人类的手可以用来拿东西、打字等，而家禽却不能实现这些功能。

4. 虚函数

虚函数是多态性编程的基础，其具体是指在基类中使用 C++关键字 `virtual` 进行声明的函数。

往往这一类函数在声明时并没有具体的实现方法,但是子类继承这些函数后则可以实现不同的功能。

在本小节中,主要向用户介绍了有关面向对象编程技术中的相关特性,并对各个特性进行了举例说明。用户通过对这些特性的了解,可以深刻理解面向对象编程的基本思想。



提示: 其中有部分知识在这里只是粗略地概括一下,在后面相关的章节中将为用户详细地进行讲解。

1.1.2 C++程序的基本结构

C++程序的基本结构与C语言的基本结构一致,只是将C语言中的输入/输出函数替换为C++语言中的标准输入/输出。即在C++语言中,使用标准输入/输出设备 `cin` 和 `cout` 分别代替C语言中的输入/输出函数 `scanf()` 和 `printf()`。同时,将C语言中的输入/输出头文件 `stdio.h` 替换为 `iostream.h`。

下面将通过一段C语言代码,向用户介绍C语言的基本结构。

【实例 1.1】 现在,有一段完整的C语言代码,其功能为将一串字符显示在屏幕上。其代码如下所示:

【本例代码位置: 源文件\01\实例 1.1\1.1.cpp】

```
01 #include<stdio.h>           //包含C语言标准输入/输出头文件
02 main()                     //主函数开始
03 {
04     char ch[256]={0};        //定义并初始化字符数组
05     printf("请输入字符: \n"); //提示用户输入字符
06     scanf("%s",ch);          //用户输入字符串
07     printf("您输入字符是: %s\n",ch); //将用户输入字符显示在屏幕上
08     return 0;                //程序正常结束
09 }
```

【运行程序】 编译并运行该程序,结果如图 1.1 所示。

【深入学习】 第 07 行代码的作用主要是向屏幕输出一串由用户自行输入的字符串。其中,符号 `\n` 和 `%s` 分别表示换行符和字符串标识符。

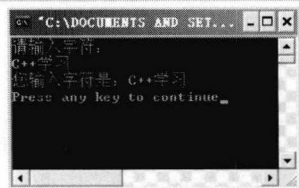


图 1.1 获取并输出字符串



提示: 用户在第 06 行代码中使用标准输入函数 `scanf()` 时,应当尽量避免使用符号 `%n`。

C 语言代码可以准确、快速地被转换为 C++ 代码。

【实例 1.2】 用户需要将实例 1.1 修改为具有 C++ 语言风格的代码,则按照前面所介绍的方法即可实现转换。转换后的 C++ 代码如下所示: