



励志改变人生
编程改变命运



零基础学 C++

(第2版)

11.5小时多媒体教学视频

杨彦强 刘袁红 王浩 等编著

本书特色

- ◎ 由浅入深，循序渐进，从零开始学C++，一点都不难
- ◎ 编程基础、编程进阶、编程应用、项目实战、上机练习、面试指南
- ◎ 266个实例、42个练习题、68个面试题

超值、大容量DVD

- ◎ 本书教学视频、本书源代码
- ◎ 本书教学PPT、本书习题答案



机械工业出版社
China Machine Press



零基础学

C++

11.5小时多媒体教学视频

杨彦强 刘袁红 王浩 等编著



机械工业出版社
China Machine Press



阅读和学习本书并不要求读者有C++语言的基础,通过几百个简洁高效的代码,本书将带领读者循序渐进地领略C++语言的魅力所在。本书采用从无到有、由浅入深、逐级递进的编写方式,尤其适合无C++语言基础或C语言基础薄弱的程序员阅读。

本书共分为六篇21章,介绍了Visual C++6开发环境的搭建、C++语言的基本数据类型、C++语言的表达式、函数、函数重载、类的继承、多态、内存管理方式、文件存储、错误与异常、RTTI、String类、编码风格与程序编译等内容,最后还对世界500强中IT公司常见面试题进行了讲解。

本书非常适合C++语言的初、中级学者阅读,并可作为开发人员的参考手册。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

零基础学C++ / 杨彦强等编著. —2版. —北京:机械工业出版社, 2012.3
(零基础学编程)

ISBN 978-7-111-37396-4

I. 零… II. 杨… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2012)第019204号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:陈佳媛

三河市杨庄长鸣印刷装订厂印刷

2012年3月第2版第1次印刷

185mm×260mm · 35.5印张

标准书号: ISBN 978-7-111-37396-4

ISBN 978-7-89433-324-7 (光盘)

定价: 69.00元(附光盘)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱: hzjsj@hzbook.com

前言

欢迎进入C++语言的世界，虽然说C++是在C语言的基础上发展而来的，但不同于C语言的面向结构编程机制，C++语言引入了面向对象编程和泛型编程机制，所以，完全可以把C++语言当成一门全新的语言来看，因此，在阅读本书时，并不要求读者有C语言的基础。

C++语言是由贝尔实验室的Stroustrup于20世纪80年代初开发的，可以说C++是一门很年轻的语言，对C++的研究也一度十分活跃，各种新的特性被提出并尝试加入到这门语言中，为了C++的有序发展，ANSI/ISO C++国际标准第1版于1998年正式发布，2003年又发布了C++国际标准第2版，新标准在老标准的基础上补充了很多内容，现今市面上大部分相关书籍，包括网络上的一些学习资料还停留在1998年的老标准上，虽然现有编译器对1998年的标准和2003年的标准都提供了支持，但从效率和未来的发展方向来看，让初学者从开始就接触较新标准，并掌握C++语言的思维方式是非常重要的，本书就是在这样一个背景下编写的。

为了使读者能循序渐进地掌握C++的语法机制和编程思想，笔者精心编写了本书。本书根据读者的一般学习习惯，以循序渐进的方式，从最简单的“Hello,World”程序写起，逐步深化、细化，对书中每个知识和技术要点都给予了详细的程序示例及代码分析，这些示例代码不仅一针见血地指明了技术要点的本质，而且短小精练，方便复制和调试。

本书特色

本书合理控制了知识点的深度和讲解的进度，给读者展示出C++语言全貌，书中所给示例代码条理清晰、简洁且直透本质，读者可以迅速掌握技术要点的内涵。笔者从事多年项目开发，经验丰富，对技术要点进行了深入浅出的阐述。本书面向初、中级读者，以大量的示例进行示范和解说，其特点主要体现在以下几个方面。

- 本书的编排采用循序渐进的方式，每章主题鲜明，要点突出，适合初、中级学者逐步掌握C++的语法规则和编程思想。
- 范例丰富，包含几百个代码示例，关键知识点都辅以示例帮助读者理解。范例程序简洁，不是简单的代码罗列，而是采用短小精练的代码紧扣所讲的技术细节，并配以详细的代码解释和说明，使读者印象鲜明，理解透彻。
- 内容全面，兼顾了C++语言所有的特性。
- 所有示例都十分简洁，具有代表性和理解意义，剪除旁枝末节，揭示技术要点的本质，每个示例集中说明一个概念或要点。
- 对于学习中经常遇到的问题与需要注意的关键点予以特别注释。
- 大量使用了图例、表格等直观的表达方式。
- 本书采用技术要点、详细介绍、示例运行等多种方式进行讲解，系统性强、可用性强，



能够给读者留下深刻的印象。

- 精选习题，为了给读者在学习知识之余提供更多的实践机会，每一章都配备了各种类型的习题。习题紧扣本章知识点，突出知识的理解和应用。同时，在习题中配备了上机操作题，结合本章的重点来考查，给读者应用所学知识的机会。
- 紧扣职场，为了给有可能参加职场面试的朋友提供更多的帮助，在本书最后一章还专门提供了常见面试题。在该章中精选了多年来著名IT公司面试题中与C++语言相关的题目，除了提供最佳答案之外，还对题目进行了深入浅出的分析，帮助读者了解题目特点，把握面试重点。

本书内容

本书分为六篇，共21章。

第一篇（第1~2章）是基础知识介绍，涵盖了程序设计语言的基本概念、Visual C++6开发环境的搭建、C++程序的构成、变量、常量、运算符与表达式、数据类型转换以及流程控制语句等内容。

第二篇（第3~7章）介绍的是面向过程的C++程序开发，分别介绍了数组和字符串，指针和引用，以及结构、共用体和链表的使用，详细且视角独到地讲述了函数的使用，详细讨论了变量的类型、生存期、作用域和可见域，总结了在编写函数时经常出现的问题，提供了函数编写的建议。

第三篇（第8~11章）介绍的是面向对象（OOP）的C++程序开发，结合精练的代码讲述了类的设计和使用、多态、虚函数、继承、代码复用、友元等内容，向读者展示了C++面向对象编程的全貌和独特魅力。

第四篇（第12~13章）讨论的是泛型编程，这是由模板来实现的，使数据结构和算法的定义可以脱离开具体类型的限制，据此，C++提供了标准模板库STL，模板和STL是该篇介绍的重点。

第五篇（第14~20章）介绍了文件存储和编程规范方面的内容，文件存储从基本的高层I/O机制讲起，而后介绍了C++流类库的使用、异常与错误机制、RTTI及转换类型符的应用，程序编码风格和C++程序编译相关的内容。

第六篇（第21章）讲解了面试题精选方面的内容，介绍了著名的IT公司关于C++语言程序开发的各种常见面试题，用来帮助读者提高笔试能力，找到满意的工作。

本书读者

本书作为C++语言的基础教程，适合于：

- C++的初学者
- 想从C语言跨越到C++语言的人员
- 了解C++但所学不全面的人员
- 想了解C++技术及最新进展的其他人员
- 高等院校理科学习C++课程的学生

- ☐ 使用C++语言进行毕业设计的学生
- ☐ 使用C++进行项目开发的人员
- ☐ 供熟悉其他语言的开发人员参考
- ☐ 企业和相关单位的培训班学员
- ☐ 掌握C++编程技术想通过此找工作的人员
- ☐ 拥有很多实践经验的高级读者，可作为参考资料

本书作者

本书主要由杨彦强、刘袁红、王浩编著，其他参与编著和资料整理的人员有冯华君、刘博、刘燕、叶青、张军、张立娟、张艺、彭涛、徐磊、戎伟、朱毅、李佳、李玉涵、杨利润、杨春娇、武鹏、潘中强、王丹、王宁、王西莉、石淑珍、程彩红、邵毅、郑丹丹、郑海平、顾旭光。

作者

2012年1月

目 录

前言

第一篇 C++基础

第1章 C++概述	1
1.1 了解计算机	1
1.1.1 计算机硬件	2
1.1.2 计算机软件	3
1.2 程序设计语言	3
1.2.1 机器语言	3
1.2.2 汇编语言	4
1.2.3 高级语言	5
1.2.4 数据结构和算法	6
1.2.5 面向过程的程序设计 (POP)	6
1.2.6 面向对象的程序设计 (OOP)	6
1.3 C++简介	7
1.3.1 C++语言发展历程	7
1.3.2 C++语言特点	8
1.3.3 C++程序开发基本过程	8
1.4 C++开发环境的搭建	9
1.4.1 Visual C++ 6开发环境简介	9
1.4.2 开发步骤	10
1.5 第一个C++程序	11
1.6 小结	12
1.7 上机实践习题	12
第2章 开始C++之旅	13
2.1 C++程序的结构	13
2.1.1 注释	14
2.1.2 编译预处理与新旧标准	14
2.1.3 主函数	14
2.1.4 名称空间	16
2.1.5 C++语素	17

2.2 变量与基本类型	18
2.2.1 变量	18
2.2.2 整型	20
2.2.3 浮点型	21
2.2.4 基本字符型	21
2.2.5 宽字符型	22
2.2.6 布尔型	22
2.3 常量	22
2.3.1 整型常量和浮点型常量	23
2.3.2 字符型常量	23
2.3.3 字符串常量	25
2.3.4 符号常量	26
2.3.5 枚举常量	27
2.4 运算符与表达式	28
2.4.1 算术运算	28
2.4.2 逻辑运算	30
2.4.3 短路表达式	30
2.4.4 关系运算	31
2.4.5 大有文章：变量是否为“0”	31
2.4.6 条件运算	32
2.4.7 位运算	32
2.4.8 赋值运算	35
2.4.9 ++和--	37
2.4.10 逗号表达式	38
2.4.11 sizeof运算符与sizeof表达式	38
2.4.12 运算符的优先级和结合性	38
2.5 类型转换	40
2.5.1 赋值转换	40
2.5.2 表达式中的转换	43
2.5.3 强制类型转换	44
2.5.4 函数调用和传递参数时的类型转换	45
2.6 流程控制语句	45
2.6.1 if...else...选择结构	45
2.6.2 switch结构	47
2.6.3 if...else...结构和switch结构的比较	51
2.6.4 for循环结构	51
2.6.5 for循环结构嵌套	53
2.6.6 while循环结构	54
2.6.7 do...while循环结构	54
2.6.8 循环语句的效率	55
2.6.9 流程转向控制语句之break	55



2.6.10 流程转向控制语句之continue	56
2.6.11 自由转向语句goto	56
2.6.12 程序中止函数exit()	57
2.7 小结	58
2.8 上机实践习题	58

第二篇 C++过程开发

第3章 数组和C风格字符串	61
3.1 什么是数组	61
3.2 一维数组	62
3.2.1 一维数组的声明	62
3.2.2 初始化一维数组	63
3.2.3 一维数组应用举例	64
3.2.4 数组操作注意事项	65
3.3 C 风格字符串	65
3.3.1 C 风格字符串的声明	65
3.3.2 字符数组的cin和cout	66
3.3.3 get()函数和getline()函数	67
3.3.4 cin与get()/getline()函数的搭配问题	68
3.3.5 访问C风格字符串中的某个元素	69
3.3.6 C风格字符串处理函数	69
3.4 多维数组	71
3.4.1 声明一个多维数组	71
3.4.2 初始化多维数组	72
3.4.3 多维数组应用举例	72
3.4.4 多维数组在内存中是如何排列元素的	73
3.5 小结	74
3.6 上机实践习题	74
第4章 指针和引用	76
4.1 指针的定义与使用	76
4.1.1 声明一个指针变量	76
4.1.2 初始化指针变量	79
4.1.3 指向指针的指针	80
4.1.4 指针赋值	80
4.2 指针的运算	81
4.2.1 指针与整数的加减	81
4.2.2 同类型指针间的比较	82
4.2.3 同类型指针相减	82
4.3 动态内存分配	83

4.3.1 使用new动态分配内存	83
4.3.2 使用delete动态释放及动态申请的内存	83
4.3.3 使用new申请动态数组	83
4.3.4 不要使用或释放已经释放的内存块	85
4.3.5 使用malloc和free动态申请内存	85
4.3.6 动态内存申请并不一定能成功	87
4.4 指针和const	87
4.4.1 禁止改写指针（常量指针或常指针）	87
4.4.2 禁止改写间接引用	88
4.4.3 既禁止改写指针，又禁止改写间接引用	88
4.5 指针与数组	88
4.5.1 数组名指针	88
4.5.2 数组元素的指针形式	90
4.5.3 指向数组的指针	90
4.5.4 指针数组	92
4.6 引用	95
4.6.1 引用的声明	95
4.6.2 引用的特点	95
4.6.3 引用的使用限制	96
4.6.4 其他要说明的问题	97
4.7 小结	97
4.8 上机实践习题	98
第5章 结构、共用体和链表	99
5.1 结构	99
5.1.1 如何定义一个结构	99
5.1.2 结构变量的声明和使用	100
5.1.3 结构变量的初始化	102
5.1.4 结构变量间是否可以相互赋值	102
5.1.5 结构体变量的sizeof	103
5.2 共用体	105
5.2.1 共用体的定义	105
5.2.2 共用体和结构的区别	105
5.2.3 共用体变量的声明和初始化	107
5.2.4 共用体使用举例	107
5.2.5 共用体的sizeof	109
5.3 结构数组和共用体数组	109
5.3.1 结构数组的声明和初始化	109
5.3.2 共用体数组的声明和初始化	110
5.4 指向结构的指针	111
5.4.1 声明一个结构指针	111



5.4.2 结构指针的初始化	112
5.4.3 使用指针访问结构成员	114
5.5 链表	114
5.5.1 链表的结构	115
5.5.2 创建链表	115
5.5.3 链表与数组的区别	116
5.5.4 链表的遍历和查找	116
5.5.5 链表的插入和删除	118
5.6 小结	121
5.7 上机实践习题	122
第6章 用函数合理组织程序	124
6.1 模块化带来的好处	124
6.1.1 函数的调用过程	124
6.1.2 抽象和封装	125
6.1.3 实现一个函数	126
6.2 函数定义	126
6.2.1 函数头	127
6.2.2 函数体	127
6.2.3 函数定义补充说明	128
6.3 函数声明	129
6.3.1 为什么要进行函数声明	129
6.3.2 如何声明一个函数	130
6.3.3 分割程序文件	131
6.4 函数调用	132
6.4.1 形参和实参	133
6.4.2 参数类型转换	133
6.4.3 值传递	134
6.4.4 指针传递	135
6.4.5 引用传递	136
6.4.6 对3种传递的补充	137
6.4.7 默认参数调用	138
6.4.8 inline函数	140
6.5 递归	141
6.6 函数的重载	143
6.6.1 何时使用函数重载	144
6.6.2 如何实现函数重载	144
6.6.3 陷阱：隐式转换导致重载函数出现二义性	144
6.7 C++如何使用内存	145
6.7.1 自动存储（栈存储）	145
6.7.2 静态存储（编译器预分配）	148

6.8 作用域与可见域	152
6.8.1 作用域	153
6.8.2 可见域	153
6.8.3 函数的作用域和可见域	156
6.9 小结	156
6.10 上机实践习题	157
第7章 关于函数的高级专题	158
7.1 内存使用错误剖析	158
7.1.1 内存泄露	158
7.1.2 野指针	159
7.1.3 试图修改常量	160
7.1.4 用错sizeof	161
7.1.5 内存越界访问	161
7.1.6 变量的初始化	162
7.2 重申：函数参数传递和返回机制	162
7.2.1 参数传递时的“副本”	162
7.2.2 函数返回时的“副本”	164
7.3 函数与指针	165
7.3.1 指向函数的指针	165
7.3.2 typedef	167
7.3.3 通过函数指针将函数作为另一个函数的参数	168
7.3.4 函数指针数组	169
7.3.5 返回函数指针的函数	171
7.3.6 带参主函数	171
7.4 函数与数组	172
7.4.1 数组名作为函数参数	172
7.4.2 通过指针得到多于1个的回传值	175
7.5 函数与结构体、共用体及类对象	176
7.5.1 3种参数调用	176
7.5.2 3种返回机制	177
7.6 函数编写的建议	177
7.6.1 合理使用const	177
7.6.2 检查输入参数的有效性	178
7.6.3 函数返回类型的判断	179
7.7 小结	179
7.8 上机实践习题	179

第三篇 面向对象的C++

第8章 面向对象技术基础	181
8.1 面向对象基本概念	181

8.1.1 类的概念	182
8.1.2 类是分层的	182
8.1.3 类和对象的关系	183
8.2 C++类的定义	183
8.2.1 类定义的基本形式	183
8.2.2 类定义示例	184
8.2.3 class和struct	185
8.3 C++类的实现	185
8.3.1 在类定义时定义成员函数	185
8.3.2 在类定义的外部定义成员函数	186
8.4 C++类的使用	187
8.4.1 声明一个对象	188
8.4.2 对象的作用域、可见域和生存期	189
8.5 对象的创建和撤销	189
8.5.1 构造函数的作用	189
8.5.2 构造函数可以有参数	190
8.5.3 构造函数支持重载	191
8.5.4 构造函数允许按参数默认方式调用	192
8.5.5 初始化表达式	192
8.5.6 析构函数	194
8.5.7 显式调用析构函数	195
8.6 复制构造函数	196
8.6.1 复制构造函数调用机制	196
8.6.2 默认复制构造函数带来的问题	198
8.6.3 解决方案——显式定义复制构造函数	199
8.6.4 关于构造函数和复制构造函数	201
8.7 特殊数据成员	202
8.7.1 const数据成员	202
8.7.2 引用成员	204
8.7.3 类对象成员	206
8.7.4 特别说明	210
8.7.5 static数据成员	210
8.8 特殊函数成员	212
8.8.1 静态成员函数	212
8.8.2 const与成员函数	214
8.9 对象的组织	215
8.9.1 const对象	215
8.9.2 指向对象的指针	217
8.9.3 对象的大小	218
8.9.4 this指针	219

8.9.5 对象数组	220
8.9.6 对象链表	221
8.10 为对象动态分配内存	221
8.10.1 使用new和delete为单个对象分配/释放动态内存	221
8.10.2 使用new和delete为对象数组分配/释放动态空间	222
8.10.3 malloc和free能否为对象动态申请内存	224
8.11 小结	224
8.12 上机实践习题	225
第9章 关于对象的高级专题	227
9.1 类的作用域	227
9.2 类定义的作用域与可见域	229
9.3 对象的生存期、作用域和可见域	232
9.3.1 先定义,后实例化	232
9.3.2 对象内存的释放与堆内存	232
9.4 友元	232
9.4.1 友元的非成员函数	233
9.4.2 友元的成员函数	234
9.4.3 友元函数的重载	236
9.4.4 友元类	236
9.4.5 友元是否破坏了封装性	237
9.5 运算符重载	238
9.5.1 运算符重载规则	238
9.5.2 以成员函数形式重载运算符	239
9.5.3 以友元函数形式重载运算符	242
9.5.4 友元函数形式和成员函数形式的比较	245
9.5.5 对运算符重载的补充说明	246
9.6 运算符重载范例	247
9.6.1 赋值运算符	247
9.6.2 函数调用运算符	250
9.6.3 下标运算符	251
9.7 类型转换	253
9.7.1 由其他类型向定义类的转换	253
9.7.2 由自定义类向其他类型的转换	255
9.7.3 隐式转换带来的二义性	257
9.8 重载函数选择规则	257
9.8.1 重载解析	258
9.8.2 什么是完全匹配	258
9.9 小结	259
9.10 上机实践习题	259



第10章 继承	262
10.1 什么是继承	262
10.1.1 简单示例	262
10.1.2 继承的层次性	264
10.2 派生类	264
10.2.1 public派生与private派生	264
10.2.2 protected成员与protected派生	265
10.3 多基派生	266
10.3.1 多基派生的声明和定义	266
10.3.2 二义性问题	267
10.3.3 解决方案	268
10.4 虚基类	268
10.4.1 共同基类带来的二义性	268
10.4.2 解决方案	270
10.4.3 虚基派生二义性与多基派生二义性不同	271
10.5 派生类的构造函数和析构函数	271
10.5.1 派生类的构造函数	271
10.5.2 派生类的析构函数	273
10.5.3 多基派生类的构造函数和析构函数	275
10.5.4 虚基派生类的构造函数和析构函数	275
10.6 分清继承还是组合	278
10.6.1 继承不是万金油	278
10.6.2 组合	278
10.7 基类与派生类对象间的相互转换	280
10.7.1 类型适应	280
10.7.2 多基继承时的情况	282
10.7.3 公共基类	282
10.7.4 虚基类的情况	284
10.8 小结	284
10.9 上机实践习题	285
第11章 多态	287
11.1 多态与虚函数	287
11.1.1 静态联编	287
11.1.2 动态联编	287
11.1.3 为什么需要虚函数	287
11.1.4 虚函数的声明和定义	290
11.2 虚函数的访问	290
11.2.1 对象名访问	291
11.2.2 指针访问	292
11.2.3 引用访问	292

11.2.4 类内访问	293
11.2.5 在构造函数或析构函数中进行访问	294
11.3 纯虚函数与抽象类	295
11.3.1 纯虚函数的声明和定义	295
11.3.2 抽象类	296
11.3.3 另一种抽象类：类中只定义了protected型的构造函数	299
11.3.4 延伸：构造函数能否为private型	301
11.3.5 虚析构函数	302
11.4 虚函数引入的二义性	305
11.4.1 多基派生	305
11.4.2 共同基类和虚继承	307
11.5 重载、覆盖与隐藏	309
11.5.1 重载	309
11.5.2 覆盖	309
11.5.3 隐藏	310
11.6 小结	311
11.7 上机实践习题	312

第四篇 泛型编程

第12章 模板	315
12.1 为什么要定义模板	315
12.1.1 类型参数化	315
12.1.2 模板的定义	317
12.2 函数模板	318
12.2.1 函数模板的使用	318
12.2.2 隐式实例化	318
12.2.3 显式实例化	320
12.2.4 特化	321
12.2.5 重载	322
12.2.6 优先级与执行顺序	322
12.3 类模板	324
12.3.1 定义类模板	324
12.3.2 隐式实例化	326
12.3.3 显式实例化	326
12.3.4 显式特化	327
12.3.5 部分特化	328
12.3.6 重载和优先级	328
12.4 模板的嵌套	329
12.4.1 函数成员模板	329



12.4.2 对象成员模板	330
12.5 模板参数	331
12.6 小结	332
12.7 上机实践习题	333
第13章 标准模板库	335
13.1 理解STL	335
13.1.1 容器	335
13.1.2 适配器	336
13.1.3 迭代器	336
13.1.4 算法	336
13.2 使用序列式容器	336
13.2.1 序列式容器的创建和元素的访问	336
13.2.2 所有容器都支持的特征	338
13.2.3 序列式容器中元素的插入和删除	339
13.2.4 vector容器	344
13.2.5 deque容器	344
13.2.6 list容器	344
13.3 使用关联式容器	344
13.3.1 set容器	345
13.3.2 multiset容器	346
13.3.3 map容器	346
13.3.4 multimap容器	348
13.4 关联式容器支持的成员函数操作	349
13.4.1 元素的插入	349
13.4.2 元素的删除	352
13.4.3 元素的查找与访问	354
13.5 迭代器	355
13.5.1 理解迭代器本质	355
13.5.2 迭代器类型	355
13.5.3 为什么要定义这么多迭代器	357
13.5.4 容器中定义的迭代器类型与5种类型的对应	358
13.5.5 流迭代器	358
13.5.6 迭代器失效	360
13.6 泛型算法	360
13.6.1 什么是函数对象	360
13.6.2 算法分类	361
13.7 适配器	365
13.7.1 容器适配器	366
13.7.2 迭代器适配器	367
13.7.3 函数适配器	369