

高等学校计算机基础教育规划教材

C++程序设计

刘维富 编著

清华大学出版社

高等学校计算机基础教育规划教材

C++程序设计

刘维富 编著

清华大学出版社
北京

内 容 简 介

本书针对计算机基础教学,介绍标准 C++ 及其面向过程和面向对象编程,框架结构模块化,内容组织由浅入深、循序渐进,理论与实践融合,符合读者的认识规律和编程能力的形成规律,便于教学的组织、实施和考核,利于教学效果的巩固和教学质量的提高。本书不但适合作为高校文科各专业公共课教材和高校计算机等级考试参考书,而且适合有一定基础的读者自学和参考。

按知识链优化原则精选范例程序,精心阐述相关编程思想、方法、语法、算法、技巧、调试技术和操作技能,注重培养学生的实际编程能力和创新能力,解决学生难学、难入门的问题。

合理分散指针内容,分别从指针的概念和不同应用领域逐步理解、掌握和运用指针,使指针不再神秘,不再让人畏惧,充分释放指针的强大编程威力。

12 个同步实验逐步提高读者的实际编程能力。1 个主流编程环境介绍使读者尽快具备自主调试程序的能力。1 套规范的实验要求保证读者取得较好的实验效果。1 套完整课件大幅减少任课教师的备课时间,快速引导读者掌握要点。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++ 程序设计 / 刘维富编著. —北京:清华大学出版社,2011.9

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-25594-9

I. ①C… II. ①刘… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 096274 号

责任编辑:袁勤勇 战晓雷

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:22.25

字 数:529 千字

版 次:2011 年 9 月第 1 版

印 次:2011 年 9 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

产品编号:036922-01

《高等学校计算机基础教育规划教材》

编 委 会

顾 问：陈国良 李 廉

主 任：冯博琴

副 主 任：周学海 管会生 卢先和

委 员：（按姓氏音序为序）

边小凡	陈立潮	陈 炼	陈晓蓉	鄂大伟
高 飞	高光来	龚沛曾	韩国强	郝兴伟
何钦铭	胡 明	黄维通	黄卫祖	黄志球
贾小珠	贾宗福	李陶深	宁正元	裴喜春
钦明皖	石 冰	石 岗	宋方敏	苏长龄
唐宁九	王 浩	王贺明	王世伟	王移芝
吴良杰	杨志强	姚 琳	俞 勇	曾 一
战德臣	张昌林	张长海	张 莉	张 铭
郑世钰	朱 敏	朱鸣华	邹北骥	

秘 书：袁勤勇

前言

C++ 是兼容 C 的面向过程和面向对象的主程序设计语言,广泛用于编写系统软件和应用软件,也广泛用于程序设计、数据结构等课程的教学。C++ 程序设计是高校普遍开设的计算机核心基础课程,涉及编程思想、方法、语法、算法、调试技术和操作技能,理论性、综合性和实践性强,使不少人感到难学、难入门、难提高。

作者长期从事 C++ 程序设计教学、研究、建设和软件开发,了解初学者学习 C++ 的困难,了解文科学生的特点,了解软件开发人员的成长规律,因此在本书的编写中力求使更多的读者学好 C++。

本书针对计算机基础教学,以及大学生对计算机知识、技能和应用方面的要求,经过作者长期构思、精心写作而成,力求使大学生了解程序设计的基本思想和方法,培养大学生使用 C++ 语言编程解决本专业实际问题的能力,并为大学生学习本专业后续计算机课程(如网页设计、数据库技术及应用、管理信息系统等)打下良好的基础。

本书的总体思路是将 C++ 语言的理论教学、实践教学紧密融合,减少重复,减轻读者的学习负担和经济负担,提高读者的学习效率。

内容组织由浅入深,循序渐进,按实际编程能力形成的三个关键期,将 C++ 教学内容从面向过程到面向对象分为三大单元,即结构化程序设计(第 1~4 章)、模块化程序设计(第 5~9 章)和面向对象程序设计(第 10~13 章),符合读者的认知规律和编程能力的形成规律,便于教学的组织、实施和考核,利于教学效果的巩固和教学质量的提高。

内容取舍注重基础性、常用性和系统性。涉及标准 C++ 的内容包括基本数据类型、运算符、表达式、输入/输出、流程控制语句、函数、数组、指针、引用、结构、单向链表、类、构造函数、析构函数、友元、静态成员、运算符重载、继承、虚函数、编译预处理、文件等。

按知识链优化原则精选范例程序,使编程思想、方法、语法、算法、技巧、调试技术和操作技能有机结合起来,想方设法讲清讲透,注重培养学生的实际编程能力和创新能力,解决学生难学、难入门、难提高的问题。

对于 C++ 语言的特色内容、重点和难点内容——指针,首先从概念上纠正了以往教材中普遍存在的问题——“指针就是地址,地址就是指针”;其次从有利于读者形成稳固的编程知识链方面考虑,将指针合理分散到全书相关章节,尽早接触指针,使读者有足够的时间,分别从指针的不同应用领域逐步理解、掌握和运用指针,使指针不再神秘和可怕,充分释放指针的强大编程威力。

编程离不开实验,为此在相关章节精心设计了 12 个以调试技术、基本算法、综合编程为核心内容的同步实验,逐步提高读者的实际编程能力。附录 C 详细介绍了主流的 Visual C++ 6.0 程序开发环境,旨在帮助读者尽快具备自主调试程序的能力。附录 D 列出了规范的实验要求,可保证读者取得较好的实验效果。

本书的配套课件可从清华大学出版社网站下载。

本书的编写得到教育部高等学校文科计算机基础教学指导委员会的立项支持,在出版过程中得到清华大学出版社的支持和帮助,在此表示衷心感谢。

编 者

2011 年 5 月

目录

第 1 章 C++ 语言概述	1
1.1 C++ 语言的起源和发展	1
1.2 C++ 语言的特点	2
1.3 C++ 语言程序设计	2
1.4 C++ 程序的开发步骤和上机调试流程	7
习题一	7
第 2 章 数据类型、运算符和表达式	9
2.1 C++ 的字符集、关键字和标识符	9
2.1.1 C++ 的字符集	9
2.1.2 标识符和关键字	9
2.2 C++ 的基本数据类型	11
2.2.1 基本数据类型	11
2.2.2 常量	13
2.2.3 变量	16
2.3 运算符与表达式	17
2.3.1 算术运算符与算术表达式	19
2.3.2 ++(增 1)和--(减 1)运算符	21
2.3.3 赋值运算符与赋值表达式	21
2.3.4 数据类型转换	22
2.3.5 关系运算符与关系表达式	24
2.3.6 逻辑运算符与逻辑表达式	25
2.3.7 逗号运算符	26
2.3.8 条件运算符“?:”	26
2.3.9 sizeof 运算符	27
2.4 常用库函数	27
2.4.1 数学库函数	27
2.4.2 伪随机函数	29

习题二	29
第 3 章 输入/输出	31
3.1 C++ 的输入/输出	31
3.2 预定义输入流对象 cin	31
3.2.1 输入十进制整数和实数	31
3.2.2 输入字符数据	32
3.2.3 输入十六进制或八进制数据	33
3.3 预定义输出流对象 cout	34
3.3.1 输出字符或字符串	34
3.3.2 输出十进制整数	35
3.3.3 输出八进制数、十六进制数	36
3.3.4 输出实数	36
3.4 预定义格式控制符	38
3.5 程序举例	39
习题三	39
实验一 简单程序设计	40
第 4 章 流程控制	47
4.1 算法	47
4.1.1 算法的概念	47
4.1.2 算法举例	48
4.1.3 算法的表达	48
4.1.4 三种基本流程控制结构	49
4.2 C++ 语言的语句	50
4.3 选择语句	50
4.3.1 条件语句	50
4.3.2 开关语句	53
4.4 循环语句	55
4.4.1 while 语句	56
4.4.2 do...while 语句	57
4.4.3 for 语句	59
4.4.4 循环语句小结	61
4.5 转向语句	63
4.5.1 break 语句	64
4.5.2 continue 语句	65
4.5.3 exit 函数	65
4.6 程序举例	66

习题四	69
实验二 选择结构	70
实验三 循环结构	71
第 5 章 函数	73
5.1 模块化程序设计方法	73
5.2 函数定义和调用	77
5.2.1 函数定义	78
5.2.2 函数调用	79
5.2.3 函数原型	82
5.2.4 函数形参的默认值	84
5.2.5 内联函数	85
5.2.6 函数重载	86
5.3 标识符的作用域	87
5.3.1 块作用域(局部作用域)	87
5.3.2 文件作用域(全局作用域)	89
5.3.3 函数原型作用域	91
5.3.4 函数作用域	91
5.4 变量的存储种类	91
5.4.1 自动变量	92
5.4.2 寄存器变量	93
5.4.3 外部变量	93
5.4.4 静态变量	95
5.4.5 小结	97
5.5 指针基础	98
5.5.1 地址的概念	98
5.5.2 指针的概念	98
5.5.3 指针变量	99
5.5.4 指针做函数的参数	100
5.5.5 指针做函数的返回值	102
5.6 引用	103
5.6.1 引用型变量的说明和使用	103
5.6.2 引用做函数的参数	103
5.6.3 引用做函数的返回值	105
5.7 函数的参数传递方式小结	106
5.8 递归函数	106
习题五	110
实验四 函数	114

第 6 章 编译预处理	116
6.1 文件包含	116
6.2 宏	118
6.2.1 不带参数的宏	118
6.2.2 带参数的宏	120
6.3 条件编译	121
习题六	123
第 7 章 数组	124
7.1 一维数组	124
7.1.1 一维数组的定义、引用和初始化	124
7.1.2 数组做函数参数	126
7.1.3 排序	131
7.1.4 查找	133
7.2 二维数组	136
7.3 字符数组	140
7.3.1 字符数组的定义和初始化	140
7.3.2 字符数组的输入/输出	142
7.3.3 字符串处理函数	143
7.3.4 应用举例	146
习题七	151
实验五 数组	153
第 8 章 指针	156
8.1 指针的运算	156
8.1.1 指针的赋值	156
8.1.2 指针的算术运算	157
8.1.3 指针的关系运算	158
8.2 指针与数组	159
8.2.1 指针与一维数组	160
8.2.2 指针与多维数组	161
8.2.3 指针与字符串	164
8.3 指针数组和指向指针的指针变量	166
8.3.1 指针数组	166
8.3.2 指向一维数组的指针变量	168
8.3.3 指向指针的指针变量	170
8.3.4 指针的引用	171

8.4 指针与函数	172
8.4.1 数组做函数的参数	172
8.4.2 带参数的 main 函数	174
8.4.3 指向函数的指针	176
8.5 new 和 delete 运算符	179
8.5.1 new 运算符	180
8.5.2 delete 运算符	181
8.6 指针(或引用)参数与数据安全	182
习题八	183
实验六 指针(一)	185
实验七 指针(二)	187
第 9 章 结构体和单向链表	189
9.1 结构体	189
9.1.1 结构体类型的定义	189
9.1.2 结构体变量的定义	190
9.1.3 结构体变量的使用	192
9.2 单向链表	196
9.2.1 单向链表的概念	196
9.2.2 单向链表的建立和基本操作	197
9.3 定义类型别名	202
习题九	203
实验八 结构体与单向链表	203
第 10 章 类和对象	205
10.1 类和对象	206
10.1.1 类的定义	206
10.1.2 类成员的访问控制	208
10.1.3 类的函数成员	208
10.1.4 对象	209
10.1.5 对象数组	211
10.1.6 this 指针	212
10.2 构造函数和析构函数	214
10.2.1 构造函数的作用	214
10.2.2 构造函数的定义和调用	214
10.2.3 构造函数的重载	216
10.2.4 默认的构造函数	217
10.2.5 构造函数的类型转换功能	218

10.2.6	对象成员与构造函数	219
10.2.7	析构函数	221
10.2.8	复制构造函数	223
10.3	友元	226
10.3.1	友元函数	226
10.3.2	友元类	228
10.4	静态成员	230
10.4.1	静态数据成员	230
10.4.2	静态函数成员	232
10.5	常数据成员、常对象和常函数成员	234
10.5.1	常数据成员	234
10.5.2	常对象	234
10.5.3	常函数成员	234
习题十		236
实验九	类和对象	238
第 11 章	运算符重载	240
11.1	运算符重载	241
11.1.1	重载二元运算符为类运算符	242
11.1.2	重载一元运算符为类运算符	244
11.1.3	重载运算符为友元运算符	246
11.2	常用特殊运算符的重载	249
11.2.1	转换函数	249
11.2.2	赋值运算符	251
11.2.3	“++”和“--”运算符	252
11.2.4	下标运算符	254
11.3	自定义字符串类	256
11.4	小结	260
习题十一		260
实验十	运算符重载	261
第 12 章	继承和派生	263
12.1	继承	263
12.1.1	基本概念	263
12.1.2	单一继承	263
12.1.3	多重继承	268
12.1.4	初始化基类成员和对象成员	270
12.1.5	应用举例	273

12.2	歧义、优先规则和赋值兼容规则	276
12.2.1	歧义	276
12.2.2	优先规则	278
12.2.3	赋值兼容规则	279
12.3	虚基类	280
12.4	虚函数	283
12.4.1	虚函数的定义	283
12.4.2	虚函数与多态性	283
12.4.3	虚函数的特殊性	285
12.4.4	纯虚函数	287
	习题十二	288
	实验十一 继承和派生	289
第 13 章	文件	291
13.1	基本概念	291
13.1.1	字节流	291
13.1.2	文件	291
13.1.3	缓冲	292
13.2	文件	292
13.2.1	文件流的用法	292
13.2.2	文件的打开	293
13.2.3	文件的关闭	294
13.2.4	文本文件的使用	294
13.2.5	二进制文件的使用	296
13.3	重载提取和插入运算符	299
13.4	应用举例	300
	习题十三	302
	实验十二 文件	302
附录 A	ASCII 码表	303
附录 B	C++ 语言常用标准库	304
B.1	数学库函数	305
B.2	字符处理函数	306
B.3	字符串处理函数	307
B.4	内存操作函数	308
B.5	通用函数	308
B.6	时间和日期函数	309
B.7	输入/输出类库常用函数成员	310

附录 C Visual C++ 6.0 程序调试技术	312
C.1 Visual C++ 的集成开发环境	312
C.1.1 菜单栏	313
C.1.2 工具栏	317
C.1.3 工作区窗口	319
C.1.4 文本编辑器和编辑菜单	320
C.1.5 文件菜单	322
C.1.6 Build 菜单	323
C.2 Visual C++ 开发 C++ 程序的基本过程	324
C.2.1 创建 C++ 程序的方式	324
C.2.2 源程序的编辑与格式化	326
C.2.3 程序的编译、链接和运行	327
C.3 程序调试	328
C.3.1 程序的错误类型	328
C.3.2 编译错误、链接错误的查看和修改	329
C.3.3 运行错误、逻辑错误的判断与调试	330
C.3.4 基本调试手段	331
C.3.5 集成开发环境的跟踪调试功能	333
C.4 Visual C++ 的帮助功能	338
附录 D 实验要求	340
参考文献	342

C++ 语言概述

1.1 C++ 语言的起源和发展

随着计算机科学技术的迅速发展,程序设计技术和程序设计语言也得到了不断发展,经历了面向机器的程序设计、面向过程的程序设计和面向对象的程序设计几个阶段。目前,面向对象程序设计是软件开发领域的主流技术,而 C++ 又是主流的面向对象程序设计语言。

C++ 语言是在 C 语言的基础上发展起来的。早在 20 世纪 60 年代,Martin Richards 为便于软件人员开发系统软件设计出 BCPL(Basic Combined Programming Language)语言。1970 年,Ken Thompson 在吸收 BCPL 语言优点的基础上设计了 B 语言,但 B 语言功能有限。1972 年,贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 在 B 语言的基础上设计出了 C 语言,以编写 UNIX 操作系统。随着 UNIX 的成功和流行,C 语言赢得了人们的青睐。到 20 世纪 80 年代,C 语言成为非常流行的结构化程序设计语言,应用领域从系统软件延伸到应用软件。

C 语言的主要特点如下:

(1) 语言结构化,简洁,规模小,数据类型丰富,使用灵活方便。既适用于设计和编写大型系统软件、大型应用软件,又适用于编写小程序。

(2) 兼有高级语言和汇编语言的特点,目标程序质量高,程序执行效率高,大大缩小了汇编语言的使用范围,提高了编程效率。

(3) 语言接口开放,不但便于 C 语言编译系统提供丰富的通用库函数,而且便于用户创建自己的专用库函数,也为第三方提供更为广泛的各类专业库函数打开了方便之门。由于有大量的库函数可供使用,因此,用户在编写程序时,不必事事从零开始,大大提高了编程效率。其实,库函数是出色的代码重用机制之一,是 C 语言广泛应用的重要基础。

(4) 程序的可移植性好。一是源于其本身已将与硬件有关的语言成分尽可能剥离(如输入/输出),由库函数实现;二是源于其有丰富的预编译命令支持;三是源于其标准化程度高,有 ISO 国际标准。

随着 C 语言的广泛应用,它的一些不足受到人们的关注,如对数据类型检查较弱,没有对面向对象技术的支持,随着软件工程规模的扩大,难以适应开发特大型的程序等。

1980 年贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充,并把 Simula 67 中类的概念引入 C 中,1983 年将其正式命名为 C++ (C Plus Plus),其含义是 C 语言的扩充。后来又把运算符的重载、引用、虚函数、模板等功能加入到 C++ 中,使 C++ 的功能日趋完善。目前,C++ 已成为面向过程和面向对象的主流通用程序设计语言。

C++ 语言受到软件厂商的极大支持,纷纷推出他们的商业化 C++ 编译系统,从早期的 Turbo C++、Borland C++、Watcom C++、Quick C++ 到目前流行的 Visual C++ 和 C++ Builder。C++ 语言也受到开放源代码组织的积极支持,他们也纷纷推出自己的非商业化的 C++ 编译系统,如 GNU C++、DEV C++、Code::Blocks 等。

值得一提的是,互联网上也出现了在线免费使用的 C++ 编译器,如:

<http://www.dinkumware.com/exam/default.aspx>

<http://www.comeaucomputing.com/tryitout>

<http://code.vcer.net>

这些在线编译器不必购买和安装,读者不妨上网试一试。

C++ 语言的标准化工作始于 1989 年,于 1994 年制定了 ANSI C++ 标准草案,并经过不断修改完善,于 1998 年 11 月被国际标准化组织(ISO)批准为国际标准。

1.2 C++ 语言的特点

C++ 语言除了具备 C 语言的特点外,还具有以下的特点:

(1) 全面兼容 C 语言,全面支持面向过程的结构化程序设计。C++ 语言是 C 的超集,大多数的 C 程序代码略作修改或不作修改就可在 C++ 编译系统下编译通过。这样,既保护了用 C 语言开发的丰富软件资源,也保护了丰富的 C 语言软件开发人力资源。

(2) 全面支持面向对象程序设计。以对象为基本模块,使程序模块的划分更合理,模块的独立性更强,程序的可读性、可理解性、可重用性、可扩充性、可测试性和可维护性等更好,程序结构更加合理。

(3) 全面支持面向过程和面向对象的混合编程,充分发挥两类编程技术的优势。

1.3 C++ 语言程序设计

目前,C++ 语言的编程环境通常是集多种编程工具于一体的,包括源程序的编辑、编译、链接、运行、调试等,使用非常方便,这种编程环境称为集成开发环境(IDE)。

C++ 的编程环境不仅支持 C++ 程序的编译和调试,而且支持 C 程序的编译和调试。通常,C++ 的编程环境约定:当源程序文件的扩展名为“.c”时,则为 C 程序;而当文件的扩展名为“.cpp”时,则为 C++ 程序。

本书主要介绍标准 C++ 语言及其程序设计,书中所有程序都在 Visual C++ 6.0 编

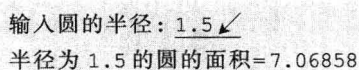
程环境下调试运行。当然,读者也可以选用其他的 C++ 编程环境,如 Dev C++ 等。除非特别声明,书中所有例题的源程序文件扩展名均为“.cpp”。

为了对 C++ 程序的基本结构有所了解,例 1.1、例 1.2 和例 1.3 分别给出了用面向过程和面向对象的程序设计方法计算圆面积的 C++ 程序。

例 1.1 面向过程程序设计。输入圆的半径,计算并输出该圆的面积。

```
1  /*****
2      源程序名: ex1_1.cpp
3      功    能: 计算并输出圆的面积
4      计算方法: 圆的面积= $\pi \times r \times r$ 
5      输入数据: 圆的半径
6      输出数据: 圆的面积
7      程序设计: 刘维富
8      设计日期: 2010.2.8
9  *****/
10     # include<iostream>
11     using namespace std;
12
13     int main(void/* 无参 */)           //每个 C++程序有且仅有一个 main 函数
14     {
15         float r;                       //定义浮点型变量 r,存放圆的半径
16         cout<<"输入圆的半径: ";       //显示提示信息,方便用户输入数据
17         cin>>r;                         //从键盘上输入圆的半径送给变量 r
18         cout <<"半径为"<<r             //输出运算结果
19             <<"的圆的面积="<<3.14159f*r*r<<"\n";
20         return 0;
21     }
```

程序运行结果为:



输入圆的半径: 1.5 ✓
半径为 1.5 的圆的面积=7.06858

例 1.1 程序说明

(1) 程序第 13 行。其中用 main 代表“主函数”的名字,由于它是整个程序执行的入口,因此,这个名字不能改变,而且每个 C++ 程序无论规模大小都有且仅有一个 main 函数。main 函数在程序中的位置不限,但为了便于阅读,通常置于程序的头部或尾部。一个 C++ 程序至少包含一个 main 函数,其余为库函数和自定义函数。main 前面的“int”的作用是声明函数的类型为整型,main 后面的“void”的作用是声明函数没有参数。

(2) 程序第 14~21 行由一对大括号{ }括起来的部分是函数体。函数体内可由零个或若干语句组成,每一语句均以“;”结束。

主函数内首先声明了一个浮点数类型的变量 r,用于存放圆的半径值,注意分号“;”是一个语句结束的标志,不能遗漏。