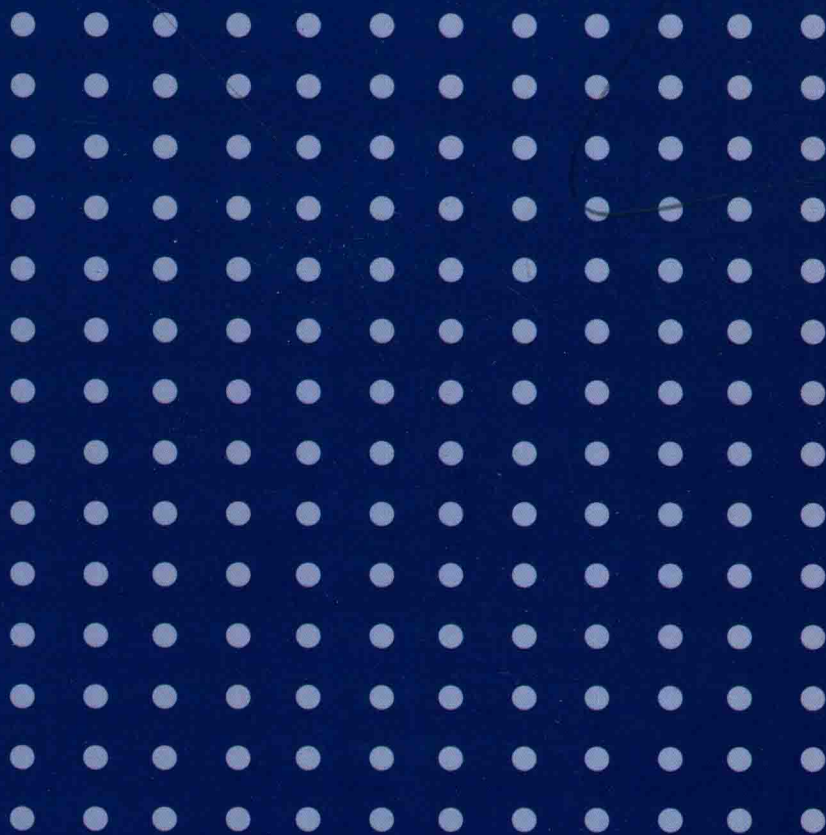


重点大学计算机专业系列教材

C++程序设计

(第2版)

邵荣 编著



清华大学出版社

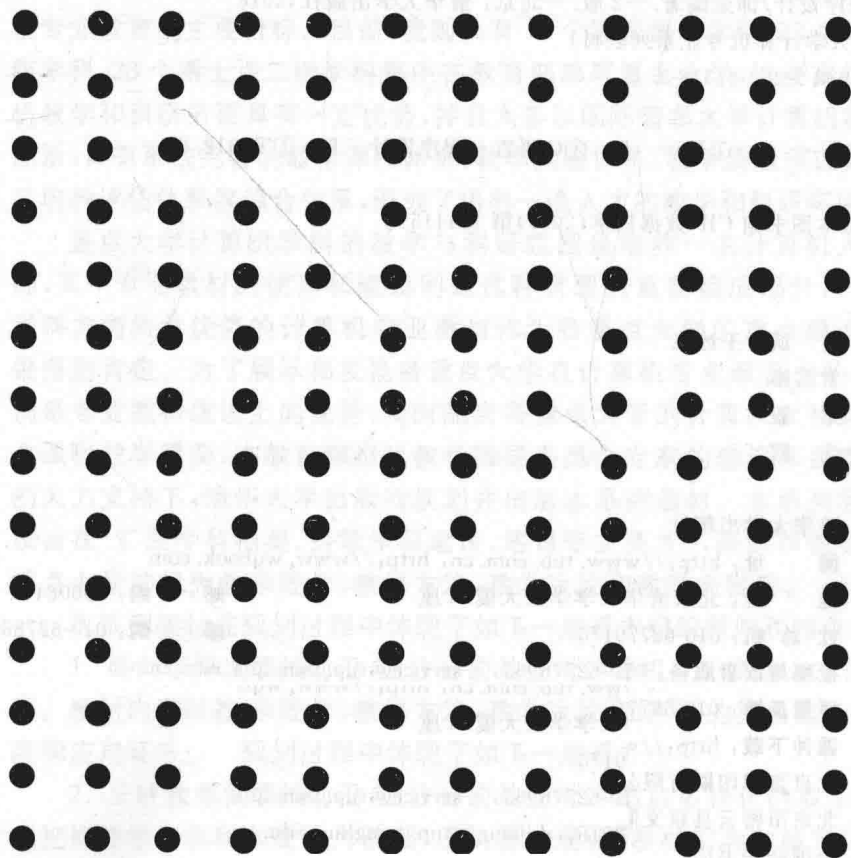


重点大学计算机专业系列教材

C++程序设计

(第2版)

邵荣 编著



清华大学出版社

北京

内 容 简 介

本书共分4篇,即基础编程、面向过程编程、面向对象编程和泛型编程,内容逐步深入。其中,基础编程篇介绍了选择和循环结构、数据和表达式的基本知识、数组结构和简单的指针使用;面向过程编程篇介绍了函数的使用、标识符的作用域和变量的存储类型、指针的高级用法和预处理;面向对象编程篇介绍了类与对象的封装性、继承性和多态性,以及对象的构造、运算符的重载、文件的使用、异常处理等;泛型编程篇则介绍了模板和简单的STL编程。

本书可作为高等院校低年级学生的教材,也可作为其他相关人员根据不同层次要求学习C++语言的参考书。

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/邵荣编著.—2版.—北京:清华大学出版社,2018

(重点大学计算机专业系列教材)

ISBN 978-7-302-47318-3

I. ①C… II. ①邵… III. ①C语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第124415号

责任编辑:贾 斌 王冰飞

封面设计:常雪影

责任校对:梁 毅

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

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:185mm×260mm

印 张:29.5

字 数:721千字

版 次:2013年9月第1版

2018年2月第2版

印 次:2018年2月第1次印刷

印 数:1~1500

定 价:59.80元

产品编号:072041-01

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

前言

一、本书的主要特点

本书的编写基于作者在多年的 C++ 教学中改进教学形式的尝试。作为 C++ 课程教材,作者除了将 C++ 的主要内容写入本书外,还进行了一些教材上的补充和尝试,包括以下方面。

- 本书的 C++ 内容兼顾 C++ 国际标准和目前还在使用的 C++ 旧版本,与大量 C++ 旧版本的书籍和某些 C++ 考试大纲相通,又顾及 C++ 的使用趋势。
- 本书的结构是从非常基本的 C++ 知识到泛型编程知识,共分 4 篇,第一篇是最基本的编写 C++ 程序的知识,用于 C++ 编程的快速入门,但也有足够丰富的内容;第二篇可以使读者编写较为复杂的 C++ 程序;第三篇可以使读者利用面向对象的方法编写大规模的程序;第四篇则提供了 C++ 国际标准提倡的泛型编程方法。
- 本书在附录部分介绍了计算机基础知识、Visual C++ 软件的基本操作和基本调试功能、C++ 旧版本与新标准的比较。
- 本书的程序示例分为 A、B、C 三类,A 类程序是最基本的 C++ 的用法,B 类程序是用于说明某个原理和用法的演示程序,C 类程序是比较经典具有实用性的 C++ 程序。当读者通过 B 类程序了解了程序的运行原理后,B 类程序几乎可以弃之不用。C 类程序是值得熟记的程序。
- 本书以提高读者编写程序的能力为主要目的,练习基本上是大量的编程题。全书 4 篇中的前 3 篇,每一篇的末尾都提供了综合练习题,以类似试卷的形式给读者提供熟悉课程考试和计算机等级考试的题型。

对于本书可以有层次地进行学习,当读者时间有限时可以只学习第一篇,能够基本掌握 C++ 编程知识并能编写各种不太复杂的程序,再学一下第 6 章就可以掌握大量以编程为工具的各种计算机知识。如果不考虑编写大规模程序,学到第二篇就可以了。第三篇是用于设计大规模程序的知识,使用了面向对象的编程方法。第四篇是 C++ 发展后期面向对象方法的新的发

展——泛型编程方法,利用类型作为参数,可以编写更加精致、高效的程序。

书中有少量加“*”的内容,是选讲或选做的内容。

二、第 2 版的改进

本书第 1 版经过 3 年的教学使用,发现了一些内容上的不足,针对这些不足,本书做了改进,主要有以下几个方面。

- 考虑到自学的需要,本书增加了书上部分题解和相关程序编写中容易出现的错误作为附录 A,在没有教师指导的情况下,附录 A 的内容可以作为读者做题的参考,可使读者了解到编程常见的错误,从而提高自己的编程能力。附录 A 中选择的题目都是编程时容易犯错的题目,或者是常见算法的题目,或者是特别难的题目。前三篇末尾的综合练习题答案和附录 B 的练习答案也放在附录 A 的后面。
- 本书将第 1 版的 Visual C++6.0 程序全部改成了第 2 版的 Visual Studio 2008 程序,剔除了少量多余或重复的程序示例。本书中的内容也针对不同版本做了相应改动,第 15 章内容改动相对较多。
- 本书在第 8 章“指针的高级用法”中增加了一小节内容,专门讲容易混淆误用的指针表达式以及指针的各种复杂关系。
- 本书在第 10 章到第 14 章的程序中,对象成员函数如 show()、get()等均用 const 说明为常成员函数,以顺应目前 C++ 的编程特点。第 12 章之后程序使用到的字符串都改为 C++ 的 string 类字符串。
- 本书中的练习做了少量改动或次序调整,难一些的题目给出了提示。
- 针对第 1 版内容有些地方不够详细,而且全书知识重点不够突出的情况,第 2 版增加了很多注解,并以粗体楷体字的形式突出显示,这些注解集中在 C++ 最主要的知识点、易错的知识点、拓展的知识点方面。

三、致谢

在本书再版的过程中得到了张强老师、黄达明老师的热心帮助,他们的知识和经验融入到了本书第 2 版的内容中,作者在此向他们及其他直接、间接对本书出版提供过帮助的人表示衷心的感谢。

作者

2017 年 10 月

目录

第一篇 基础编程

第 1 章 C++ 概述	3
1.1 C++ 历史背景	3
1.2 C++ 语言的特点	5
练习题	5
第 2 章 简单的 C++ 程序	6
2.1 流程框图	6
2.2 简单的顺序结构程序	7
2.3 有分支和循环结构的程序	16
2.3.1 有分支结构的程序	16
2.3.2 有循环结构的程序	20
2.3.3 结构化的程序设计及结构化流程图	24
练习题	26
第 3 章 数据与运算	28
3.1 引言	28
3.2 字符集与标识符	30
3.3 基本数据类型	31
3.4 常量	34
3.4.1 字面常量	34
3.4.2 符号常量	36
3.5 基本运算及表达式	37
3.5.1 算术运算、比较运算和逻辑运算	38
3.5.2 增量/减量运算、赋值运算和逗号运算	38

3.5.3 字符数据的处理	40
3.6 对表达式的进一步说明	41
3.6.1 整数除和算术溢出	41
3.6.2 比较运算的特殊用法	42
3.6.3 前置与后置的增量和减量	44
3.6.4 类型的转换	45
3.6.5 短路表达式	48
3.7 位运算	50
*3.8 数据的输出格式控制	52
练习题	55
第 4 章 选择结构和循环结构	59
4.1 if 语句及其嵌套	59
4.2 条件表达式	62
4.3 switch 语句	63
4.4 while 语句和 do...while 语句	65
4.5 for 语句	67
4.6 break 语句和 continue 语句	70
*4.7 goto 语句及语句标号	74
练习题	75
第 5 章 复合类型数据	78
5.1 数组	78
5.1.1 一维数组	78
5.1.2 多维数组	82
5.1.3 字符数组及字符串处理	86
5.1.4 字符串处理函数	90
5.1.5 常用数组算法	91
5.2 指针	99
5.3 引用	107
5.4 结构与联合	108
5.4.1 结构	108
*5.4.2 联合	113
5.5 枚举	115
5.6 用 typedef 定义类型	117
练习题	119
第一篇综合练习题	123

第二篇 面向过程编程

第 6 章 函数	129
6.1 函数的定义	129
6.2 函数的参数传递	133
6.3 递归函数、重载函数和默认参数函数	137
6.3.1 递归函数	137
6.3.2 重载函数	140
6.3.3 默认参数函数	141
6.4 内联函数	143
练习题	144
第 7 章 作用域和存储类型	147
7.1 标识符的作用域	147
7.1.1 局部作用域和全局作用域	147
7.1.2 标识符的定义与说明以及混合作用域	150
7.1.3 语句作用域	151
7.1.4 名空间及名空间作用域	153
7.1.5 标识符的重名	156
7.1.6 函数说明的隐藏规则	158
7.2 程序的内存映像	159
7.3 变量的存储类型	163
7.3.1 auto 类型和 static 类型	163
7.3.2 register 类型	166
7.3.3 无名变量——函数值	167
7.3.4 多文件程序中的变量和函数	168
练习题	169
第 8 章 指针的高级用法	171
8.1 指针与函数	171
8.1.1 指针作参数	171
8.1.2 指针传递数组和返回指针	173
8.2 不定参数	175
8.3 各类指针	177
8.3.1 字符指针	177
8.3.2 行指针	177
8.3.3 指针数组与指向指针的指针	180
8.3.4 函数指针	186

* 8.3.5	void 指针	189
8.3.6	const 与指针	191
8.3.7	容易混淆的指针使用	194
8.4	跨函数引用	195
8.4.1	引用型参数	195
8.4.2	返回引用	196
8.5	链表和二叉树	198
8.5.1	单链表	198
8.5.2	双向链表	203
* 8.5.3	二叉树	204
8.6	程序中的指针错误用法	207
	练习题	211
第 9 章	预处理命令	214
9.1	包含命令	215
9.2	宏定义	216
9.2.1	无参宏定义	217
9.2.2	带参宏定义	219
9.3	条件编译命令	221
9.4	程序的编译与连接	225
	练习题	226
第二篇	综合练习题	228

第三篇 面向对象编程

第 10 章	类与对象	235
10.1	类与对象的定义	236
10.1.1	结构的演变	236
10.1.2	类的定义与使用	237
10.1.3	内联成员函数	240
10.1.4	this 指针	240
10.1.5	对象的使用	241
10.2	类的构造与析构	243
10.2.1	构造函数与析构函数	243
10.2.2	复制构造函数	249
10.2.3	临时对象	252
10.2.4	调用函数时数据传递中的对象	253
10.3	动态对象与对象数组	255
10.3.1	对象数组的构造与析构	255

10.3.2	动态对象与动态对象数组	257
10.3.3	对象的生命周期	259
10.4	常成员和对象类成员	261
10.5	内嵌类型	263
	练习题	265
第 11 章	类的静态成员与类的友元	267
11.1	静态成员	267
11.1.1	静态成员变量	267
11.1.2	静态成员函数	269
11.1.3	静态成员变量与全局变量的比较	271
11.2	类的友元	272
11.2.1	友元函数	272
11.2.2	友元类	273
	练习题	275
第 12 章	运算符的重载	277
12.1	类成员的运算符重载	277
12.2	非成员的运算符重载	279
12.3	赋值运算符的重载	281
12.4	转换函数	285
12.5	特殊的运算符重载	287
12.5.1	重载++和--	287
12.5.2	重载<<和>>	290
12.5.3	重载[]、*(指向)和->	291
12.5.4	重载()和函数对象	294
12.6	string 类的使用	296
	练习题	298
第 13 章	类的继承性	301
13.1	单一派生	302
13.1.1	派生类的定义	302
13.1.2	公有派生、私有派生和保护派生	304
13.1.3	重名的优先级和隐藏	308
13.2	派生类的构造与析构	310
13.3	多重派生	313
13.4	重名的使用规则	315
13.5	派生类的类型兼容	317
13.6	虚基类及复杂类的构造与析构	320

练习题.....	325
第14章 类的多态性	329
14.1 虚函数.....	329
14.2 虚函数的使用规则和内部实现原理.....	334
14.2.1 虚函数的使用规则.....	334
14.2.2 虚函数的内部实现原理.....	339
14.3 纯虚函数与抽象类.....	341
练习题.....	347
第15章 C++输入/输出系统	351
15.1 C++输入/输出系统的结构	352
15.2 C++流的状态	353
15.3 输入/输出成员函数及插入符<<和提取符>>	355
*15.4 C++流的格式控制	360
15.5 文件流.....	365
15.5.1 文件的打开与关闭.....	366
15.5.2 文本文件的使用.....	368
15.5.3 二进制文件的使用.....	371
*15.6 字符串流.....	376
练习题.....	378
第16章 异常	380
16.1 异常机制.....	380
16.2 异常的使用.....	381
16.3 异常类.....	385
练习题.....	386
第三篇综合练习题	387
第四篇 泛型编程	
第17章 模板	395
17.1 函数模板.....	395
17.2 类模板.....	398
练习题.....	401
第18章 STL编程简介	402
18.1 STL的简单例子	402
18.2 STL容器与迭代器	403

18.2.1 顺序容器.....	403
18.2.2 关联容器.....	406
18.3 STL 算法	407
练习题.....	411
附录 A 部分题解及常见错误.....	412
附录 B 计算机基础知识	444
附录 C 使用 Visual C++ 编写控制台应用程序	452
附录 D C++ 旧版本和 C++ 标准的常见区别	456
参考文献.....	458

第 1 章 C++ 概述
第 2 章 编译 C++ 程序
第 3 章 运算符与表达式
第 4 章 进阶结构体和联合体
第 5 章 用户自定义类型

基础编程

第一篇

1.1 C++ 历史背景

第 1 章 C++ 概述

第 2 章 简单的 C++ 程序

第 3 章 数据与运算

第 4 章 选择结构和循环结构

第 5 章 复合类型数据

C++ 概述

第 1 章

1.1 C++ 历史背景

1946 年,世界上第一台电子数字计算机——ENIAC 在宾夕法尼亚大学研制成功,该事件标志着人类进入了计算机时代。

在计算机处理问题时,首先需要人们编写解决问题的程序。最早期的程序是直接计算机的二进制指令编写的,计算机的二进制指令集称为机器语言。

后来,为了更好地记忆和使用这些二进制指令编写程序,人们将指令符号化,称为用符号表示的指令,这些符号化的指令集称为汇编语言。用汇编语言编写的指令只是指令的符号表示,不能直接在计算机上执行,需要将这些符号转换成二进制指令才能执行,这个转换工作是由软件来完成的,称为汇编过程。

用汇编语言编写程序,一个简单的功能(如计算一个多项式的值)可能需要许多指令才能完成,因此用汇编语言编程是一个低效的工作。而且汇编语言所用的指令是机器能实现的基本操作,机器的基本操作与人们认可的基本操作是不一样的。例如一个多项式的计算,对于人们而言是基本操作,而对于机器而言就不是一个基本操作,需要很多指令才能实现。机器语言、汇编语言等完全面向机器的计算机语言称为计算机低级语言。

为了解决低级语言编程低效的问题,人们将一些对人而言基本的操作作为指令组合,用相应含义的语句形式表示,这样一些语句的全体及使用规则称为高级语言。由于高级语言是一种面向人的计算机语言,因此人们编程的效率和可靠性得到了巨大的提高。用高级语言编写的程序虽然适合人们阅读理解,但是无法被机器执行,需要用软件进行转换。与汇编语言的一对一的符号指令转换相比,一句高级语言的语句要对应许多的指令,因此转换更加复杂。高级语言语句的转换过程是分两步来实现的,即编译和连接。其中,编译就是单纯地将语句转换成对应的指令组合,连接则是将指令和数据统一安排分配,构成完整的指令数据集合,然后保存为可执行文件。该可执