



普通高等教育“十一五”国家级规划教材

教育部普通高等教育精品教材



“十二五”江苏省高等学校重点教材



华章教育

面向CS2013计算机专业规划教材

程序设计教程

用C++语言编程

第3版

陈家骏 郑滔 编著
南京大学

Fundamentals of Programming with C++
Third Edition



机械工业出版社
China Machine Press



普通高等教育“十一五”国家级规划教材

教育部普通高等教育精品教材



“十二五”江苏省高等学校重点教材 (编号: 2014-1-029)



面向CS2013计算机专业规划教材

程序设计教程

用C++语言编程

第3版

陈家骏 郑滔 编著
南京大学

*F*undamentals of Programming with C++
Third Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

程序设计教程：用 C++ 语言编程 / 陈家骏，郑滔编著. —3 版. —北京：机械工业出版社，2015.5

(面向 CS2013 计算机专业规划教材)

ISBN 978-7-111-50123-7

I. 程… II. ① 陈… ② 郑… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2015) 第 091354 号

本书以 C++ 为编程语言，介绍程序设计的基本思想、方法和技术。本书内容围绕程序设计的基础知识、过程式和面向对象程序设计基本内容进行展开，主要包括：计算机基础知识、基本数据类型、常量、变量、操作符、表达式、流程控制、结构化程序设计、过程抽象、递归函数、复杂数据类型（数组、结构等）、数据抽象、继承、类属类型（STL）、输入/输出、异常处理以及事件驱动和基于 MFC “文档 - 视”结构的面向对象的 Windows 应用程序框架等。

本书可作为高等院校本科生第一门程序设计课程的教材，也可供程序设计初学者和有一定编程经验的人员参考。

出版发行：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：余 洁

责任校对：董纪丽

印 刷：北京诚信伟业印刷有限公司

版 次：2015 年 6 月第 3 版第 1 次印刷

开 本：185mm×260mm 1/16

印 张：24.75

书 号：ISBN 978-7-111-50123-7

定 价：45.00 元

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

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

第3版前言

自本教材的第2版出版以来,根据我们的教学实践以及广大读者的反馈意见,我们发现教材还存在一些不尽如人意的地方,现予以修订。

教材第3版的变动主要体现在以下几个方面:

1. 重新组织和调整了一些章节的内容,使得教材内容安排更加合理,并进一步突出教材对主程序设计思想、概念和技术的介绍。例如:把对整数的补码表示以及实数的浮点表示的描述集中放入1.1.3节“机内信息表示”中介绍;把基于断言的程序调试从第4章“过程抽象——函数”移至第10章“异常处理”中介绍;在第5章“复合数据的描述——构造数据类型”的子标题中显式指出每种类型的作用;把“操作符重载”从单独的一章(第2版的第7章)变成一节(6.6.5节)放入6.6节的“对象与类的进一步讨论”中;把C++的编译预处理命令(包括条件编译)、常用标准函数和STL算法以及MFC常用类的介绍放到附录中;围绕“消息驱动”和“文档-视”软件结构重新组织了第11章对基于MFC的面向对象程序设计的介绍;等等。

2. 针对C++新的国际标准(C++11),增加一些对程序设计有良好支持的C++内容。例如:增加了对 λ 表达式的介绍,包括 λ 表达式的定义(4.6.4节“匿名函数—— λ 表达式”)、实现(6.6.5节中的“函数调用操作符重载”)以及应用(5.5.6节“函数指针”和8.3节“C++标准模板库”),通过 λ 表达式可以实现匿名函数,它把函数定义和使用合二为一,以提高程序中“临时用一下”的小函数的灵活性;增加了对转移构造函数(6.6.4节“对象拷贝构造过程的优化——转移构造函数”)和转移赋值操作符重载函数(6.6.5节)的介绍,它们基于“右值引用”参数类型实现把资源从即将消亡的对象转移(而不是复制)到新创建的或已有的对象中,从而提高程序效率;等等。

3. 针对初学者,尤其是自学者,对一些内容的描述进行了完善,并对全书的语言文字和逻辑进行了优化,使得教材更加便于阅读;对教材例子中的程序代码增加了注释,使得它们更加容易理解。另外,尽量减少了在前面出现而在后面才会详细介绍的概念,以避免给初学者带来困扰。对于必须提前出现的概念以“将在……节……中介绍”的引用形式给出,而在后面用到前面介绍的内容时,将采用“参见……节的……”的引用形式。

4. 补充了一些习题,使得读者能更好地理解 and 掌握核心内容,并有针对性地进行程序设计训练。

5. 修正了教材第2版中的一些错误。

本书修订过程中得到了很多人的帮助,在第3版出版之际向他们表示感谢。特别感谢刘奇志老师和黄书剑老师,他们与作者一起承担程序设计课程的教学过程中发现了本教材的一些问题,并对本教材的修订提出了很多很好的建议,作者获益良多。另外,还要感谢策划编辑朱劼对我们的鼓励与鞭策,并为教材编写出谋划策。

作者于南京大学
2015年2月



第2版前言

本书第1版自2004年出版以来,得到了广大读者的热情关注和支持,很多读者还提出了宝贵的建议,我们深表感谢。

在近几年的教学中,我们也发现了本书的一些不足之处。首先,编写该教材的初衷是介绍程序设计的基本思想、概念和技术,C++语言是作为编程实现语言的角色出现,然而,在教材某些内容的表述上违背了这个初衷,教材的一些地方出现了C++语言“喧宾夺主”的情况。其次,教材在一些内容的表达上过于“精炼”,使初学者有“看天书”的感觉。再次,教材对现在比较流行的C++标准模板库(STL)以及它所支持的泛型程序设计没有给出足够的介绍,从而给读者学习使用STL带来了困难。此外,教材中还存在少量的错误。

针对上述问题,我们对教材进行了修订。第2版的变动主要体现在以下几个方面:

1) 重新组织了一些章节的内容,并调整了相应章节(主要是节)的标题和次序,进一步突出了程序设计的主流思想、概念和技术。

2) 对教材的文字进行了润色,补充了例子,并为例子中的程序代码增加了注释,使之更加容易理解。

3) 补充了对STL的介绍,包括一些常用的容器和算法以及它们的使用实例,有利于读者更好地进行泛型程序设计。

4) 增加了对计算机内部信息表示的介绍,使得读者能更好地理解程序设计中涉及的二进制。

5) 补充了一些习题,使读者有更多的机会进行有针对性的训练。

6) 对一些重要的程序设计术语用不同的字体加以突出的标注并给出了它们的英文对照,突出了对程序设计重要概念的介绍。

7) 修正了上一版中的一些错误。

在教材的修订过程中,得到了很多人帮助,在教材第2版出版之际向他们表示感谢,并希望继续得到大家的支持,使教材进一步得到完善。

作者于南京大学

2009年2月

李天意南于书社

R 5 平 2105

第1版前言

随着计算机应用领域的不断扩大、应用层次的不断加深,社会对计算机软件的需求急剧增长,这就导致了软件的规模不断扩大、复杂程度不断提高。如何设计出大量的满足用户需求的高质量软件是软件工作者所面临的严峻挑战。

作为计算机软件的主要表现形式——计算机程序不同于其他程序(如音乐会程序),它是由计算机来执行的。这就使得计算机程序的编制(程序设计)不能完全以人的思维模式和习惯来进行,它往往要受到计算机解决问题的方式和特点的限制。除此之外,要设计出解决各种问题的程序,程序设计者往往还需要了解与问题领域有关的知识。这些都给程序设计带来一定的难度。

从程序设计的发展历史来看,程序设计经历了从低级语言到高级语言、从以编码为中心到面向软件生存周期的软件工程、从过程式到面向对象的发展过程。这一过程体现了人们对程序设计活动的不断认识和改进的过程,特别是从过程式程序设计到面向对象程序设计的发展,体现了人们对以自然的方式来描述和解决问题的需求,它使得解题过程更接近于人的思维方式。

有人认为程序设计是一门艺术,而艺术基于人的灵感和天赋。对于一些小型程序的设计而言,上述的说法可能有一些道理。但是,对于大型、复杂的程序设计问题,灵感和天赋是不能很好地解决问题的,几十年的程序设计实践已证明了这一点。不可否认,程序设计需要灵感和天赋,它们往往在程序的一些局部设计上发挥着作用。但从总体上讲,程序设计是一门科学,它是有规律和步骤可循的。通过对程序设计的基本思想、概念和技术的学习,再加上必要的训练和实践,程序设计的规律和步骤是可以掌握的,这正是本书的主旨所在。本书强调准确的程序设计基本概念、良好的程序设计风格和实际动手能力的训练与培养。

关于C++语言是否适合作为介绍程序设计时的编程实现语言,目前存在不同的看法。持反对意见的人认为C++太灵活,以至于会使初学者感到无所适从。本书之所以选择C++语言作为实现语言,首先,因为C++语言是一种流行的高级语言,很多人都在用C++编写实际的程序;其次,C++支持大部分基本的程序设计思想、概念和技术,其中包括对过程式及面向对象两种程序设计范型的支持。再次,与其他高级语言相比,C++语言具有灵活和高效等特点,这使得一些程序设计思想、概念和技术能够更好地实现。本书是以介绍基本的程序思想、概念和技术为主旨,C++服务于这个主旨,而不是相反,这样,初学者在使用C++时能够做到有的放矢。因此,本书对C++的一些特殊的、用于解决非主流的程序设计问题的成分和技巧不予重点介绍。特别地,本书对一些属于C++语言“文化”范畴的内容不予过分强调。

本书的内容分成两大部分:第1章至第5章为第一部分,主要对一些基本的程序设计思想、概念和技术以及过程式程序设计的基本内容进行介绍,其中包括:数据类型、表达式、流程控制、子程序(函数)、递归等;第6章至第12章为第二部分,重点介绍面向对象程序设计的基本内容,其中包括:类/对象、继承、操作符重载、类属(模板)、输入/输出、异常处理以及面向对象的Windows应用程序基本框架等。

本书既适合于程序设计的初学者使用,同时,对具有一些程序设计经验的人也有一定的参考价值。本书可以作为一学年的程序设计课程使用,其中,第一学期介绍第1章到第5章的全部内容以及第10章的部分内容;第二学期介绍第6章到第9章全部内容、第10章的部分内容以及第11章到第12章的全部内容。如果读者已学过过程式程序设计(如C语言程序设计等)的基本内容,则本书也可作为一学期的面向对象程序设计课程使用,重点介绍第6章到第12章的内容。书中加“*”标记的节在初次阅读时可以跳过。

本书的编写和完成与很多人的帮助是分不开的。首先,要感谢郑国梁教授对本书编写工作的精心指导。在内容的选取、安排、用语的规范性等方面,郑老师都事无巨细地给予了考虑,并检查了全文(包括每个例子程序)。值得一提的是,作者编写本书所必备的专业知识和专业素质是在郑老师的长期熏陶下获得的,这些知识和素质使得作者能够完成本书的编写。其次,非常感谢尹存燕老师和戴新宇在本书习题的设计和文字易读性方面所做的大量工作;非常感谢孙明欣和周明对书中内容所做的检查工作,特别是对本书初稿中一些概念上的模糊与谬误、内容安排的合理性与易读性以及遵守C++标准规范方面所提出的建议;感谢胡昊和徐锋,作者对一些基本概念的理解是在与他们就相关问题的讨论中获得的。另外,还要感谢机械工业出版社的温莉芳总编辑对本书编写工作的鼓励和支持。

最后,要感谢我们的家人对本书编写工作的理解和支持,本书的编写占用了大量本应与他们共度的家庭欢乐时间。感谢所有支持和帮助过本书编写工作的人们。

由于作者水平有限,加之时间仓促,错误和疏漏在所难免,恳请广大读者不吝指教,以便于我们在今后的版本中进行改进。

陈家骏

南京大学计算机科学与技术系

chenjj@nju.edu.cn

郑浩

南京大学软件学院

zt@nju.edu.cn

2004年4月

教学建议

教学章节	教学要求	课时	
		两学期	一学期
第1章	- 了解冯·诺依曼体系结构计算机及其硬件和软件构成 - 了解计算机内的信息表示 - 了解程序设计范式 - 了解程序设计语言及其翻译（编译和解释） - 了解程序设计的基本步骤 - 了解 C++ 语言的特点、C++ 程序的构成以及 C++ 程序的开发环境 - 了解 C++ 语言的词法	4	0.5
第2章	- 了解数据类型的概念 - 掌握 C++ 的各种基本数据类型 - 掌握常量和变量的使用以及变量值的输入 - 掌握 C++ 操作符的使用，对操作数的类型转换有一定的认识 - 掌握表达式的使用（包括优先级与结合性、类型转换等）以及表达式的输出 - 了解表达式的副作用问题	6	0.5
第3章	- 了解程序流程控制的基本思想（算法） - 掌握 C++ 语言的顺序控制、选择控制、循环控制以及无条件转移控制语句的使用 - 掌握基于循环的问题求解方法（迭代法和穷举法），区分计数循环和事件循环 - 了解程序设计风格和结构化程序设计	6	0.5
第4章	- 了解基于过程抽象的过程式程序设计基本思想 - 掌握 C++ 函数的定义、调用以及值参数传递 - 掌握局部变量与全局变量以及 C++ 程序的多模块结构 - 掌握标识符的作用域与变量的生存期概念 - 了解“分而治之”的程序设计思想和掌握递归函数的使用 - 了解内联函数、带默认值的形式参数、函数名重载以及匿名函数的基本内容 - 了解 C++ 的标准函数库	6	0.5
第5章	- 掌握枚举类型及其运用 - 掌握一维、二维数组，以及字符串类型及其运用 - 掌握结构类型及其运用 - 了解联合类型的作用 - 掌握指针类型的基本操作，以及把指针应用于参数传递和动态变量 - 了解指针与数组的配合使用 - 了解函数指针的运用 - 掌握引用类型的使用	18	2
第6章	- 了解数据抽象的基本思想 - 掌握 C++ 类的定义、类成员的访问控制、对象的创建和操作、对象的初始化以及消亡处理 - 了解 this 指针的作用 - 掌握成员对象的初始化以及拷贝构造函数的使用	14	14

(续)

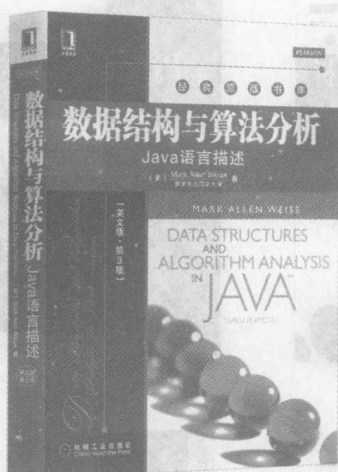
教学章节	教学要求	课时	
		两学期	一学期
第 6 章	- 了解对常量对象的访问、对象之间的数据共享（静态成员）、提高对私有成员访问效率（友元）以及拷贝构造的优化等问题 - 了解基于类的程序模块结构 - 了解操作符重载的基本思想和实现方法 - 了解 C++ 几种特殊操作符重载的实现和具体应用	14	14
第 7 章	- 理解类继承作为软件复用的手段 - 掌握 C++ 单继承的使用 - 了解 C++ 的 protected 访问控制的作用 - 了解 C++ 继承方式的作用 - 掌握派生类对象的初始化和消亡处理 - 了解聚集的概念以及与继承的比较 - 掌握通过虚函数实现消息处理的动态绑定机制 - 掌握抽象类的使用 - 了解多继承及其问题	8	8
第 8 章	- 了解类属类型（泛型）的基本思想以及模板的作用 - 掌握函数模板和类模板的基本使用 - 了解模板的复用原理 - 了解 C++ 标准模板库 STL 的基本思想 - 学会使用 STL 的容器、算法以及迭代器来解决一些程序设计问题	6	6
第 9 章	- 了解 C++ 中输入 / 输出的基本思想 - 了解基于 scanf 和 printf 的控制台输入 / 输出 - 掌握基于 cin 和 cout 的控制台输入 / 输出 - 掌握文件的组织方式和面向文件的输入 / 输出 - 了解面向字符串变量的输入 / 输出 - 掌握抽取操作符“>>”和插入操作符“<<”的重载	6	4
第 10 章	- 了解异常处理的基本思想 - 掌握 C++ 异常处理机制	2	2
第 11 章	- 了解事件驱动的程序结构 - 了解 Windows 应用程序的对象组成 - 掌握基于 MFC 和“文档－视”结构的应用框架，能够进行 Windows 应用程序的开发	8	8
总课时		84	46

说明：

1) 本书可作为一学年（两学期）的程序设计课程使用，也可作为一学期的面向对象程序设计课程使用。

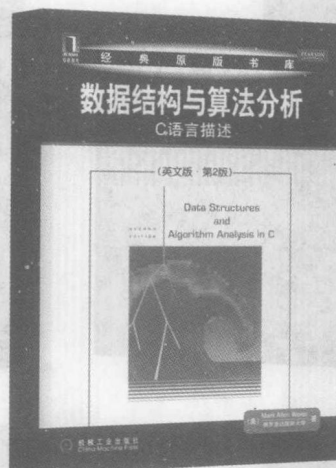
2) 本书授课学时包含理论讲授、课堂讨论以及习题讲解等必要的课内教学环节。

推荐阅读



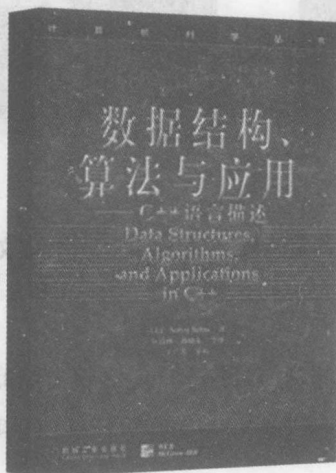
数据结构与算法分析：Java语言描述（英文版·第3版）

作者：Mark Allen Weiss ISBN：978-7-111-41236-6 定价：79.00元



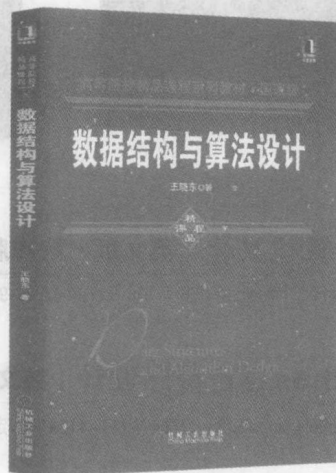
数据结构与算法分析：C语言描述（英文版·第2版）

作者：Mark Allen Weiss ISBN：978-7-111-31280-2 定价：45.00元



数据结构、算法与应用：C++语言描述

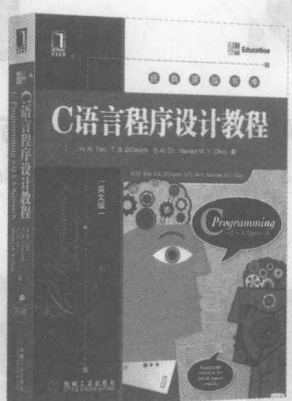
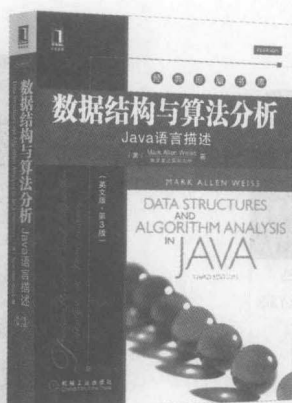
作者：Sartaj Sahni ISBN：7-111-07645-1 定价：49.00元



数据结构与算法设计

作者：王晓东 ISBN：978-7-111-37924-9 定价：29.00元

推荐阅读



计算机文化 (英文版·第15版)

作者: June Jamrich Parsons 等 ISBN: 978-7-111-42803-9 定价: 79.00元

数据结构与算法分析: Java语言描述 (英文版·第3版)

作者: Mark Allen Weiss ISBN: 978-7-111-41236-6 定价: 79.00元

Python语言程序设计 (英文版)

作者: Y. Daniel Liang ISBN: 978-7-111-41234-2 定价: 79.00元

C++程序设计 (英文版·第3版)

作者: Y. Daniel Liang ISBN: 978-7-111-42505-2 定价: 79.00元

C语言程序设计教程 (英文版)

作者: H. H. Tan 等 ISBN: 978-7-111-40432-3 定价: 49.00元

Java语言程序设计 (英文版·第8版)

作者: Y. Daniel Liang 基础篇 ISBN: 978-7-111-36122-0 定价: 89.00元

进阶篇 ISBN: 978-7-111-36125-1 定价: 89.00元

目 录

第3版前言	1
第2版前言	1
第1版前言	1
教学建议	1
第1章 概述	1
1.1 计算机的工作模型	1
1.1.1 冯·诺依曼体系结构	1
1.1.2 硬件与软件	2
1.1.3 机内信息表示	4
1.2 程序设计	8
1.2.1 程序设计范式	8
1.2.2 程序设计步骤	9
1.2.3 程序设计语言	11
1.3 C++ 语言	13
1.3.1 概述	13
1.3.2 C++ 程序的构成	14
1.3.3 C++ 语言的词法	15
1.3.4 C++ 程序的运行步骤	17
1.4 小结	18
1.5 习题	19
第2章 简单数据的描述——基本数据类型和表达式	21
2.1 概述	21
2.2 基本数据类型	22
2.2.1 整数类型	22
2.2.2 实数类型	23
2.2.3 字符类型	23
2.2.4 逻辑类型	24
2.3 数据的表现形式	24
2.3.1 常量	25
2.3.2 变量	27
2.3.3 变量值的输入	29
2.4 数据的基本运算——操作符	30

2.4.1 操作符概述	30
2.4.2 算术操作符	30
2.4.3 关系与逻辑操作符	31
2.4.4 位操作符	34
2.4.5 赋值操作符	36
2.4.6 其他操作符	37
2.4.7 操作数的类型转换	38
2.5 数据基本运算的实现——表达式	42
2.5.1 表达式的构成与分类	42
2.5.2 操作符的优先级和结合性	43
2.5.3 表达式中操作数的类型转换	44
2.5.4 带副作用操作符的表达式计算	45
2.5.5 表达式结果的输出	45
2.6 小结	46
2.7 习题	47
第3章 程序的流程控制(算法)描述——语句	49
3.1 概述	49
3.2 顺序执行	50
3.2.1 表达式语句	50
3.2.2 复合语句	51
3.2.3 空语句	52
3.3 选择执行	53
3.3.1 if 语句	53
3.3.2 switch 语句	58
3.4 循环(重复)执行	61
3.4.1 迭代与穷举	61
3.4.2 while 语句	62
3.4.3 do-while 语句	63
3.4.4 for 语句	64
3.4.5 计数循环和事件循环	65
3.4.6 循环程序设计实例	68
3.5 无条件转移	73
3.5.1 goto 语句	73
3.5.2 break 语句	75

3.5.3 continue 语句	76	5.1.2 枚举类型的操作	124
3.6 程序设计风格	77	5.2 数组类型——由多个同类型元素构成的复合数据描述	126
3.6.1 结构化程序设计	78	5.2.1 一维数组类型——线性复合数据的描述	127
3.6.2 关于 goto 语句	78	5.2.2 一维字符数组——字符串类型的一种实现	131
3.7 小结	79	5.2.3 二维数组类型——二维复合数据的描述	134
3.8 习题	80	5.2.4 数组类型的应用	138
第4章 过程抽象——函数	82	5.3 结构类型——由若干属性构成的复合数据描述	143
4.1 概述	82	5.3.1 结构类型的定义	143
4.1.1 功能分解与复合	82	5.3.2 结构类型的操作	145
4.1.2 子程序及子程序间的数据传递	83	5.3.3 结构类型的应用	148
4.2 C++ 函数	84	5.4 联合类型——用一种类型表示多种类型的数据	152
4.2.1 函数的定义	84	5.4.1 联合类型的定义与操作	152
4.2.2 函数的调用	86	5.4.2 联合类型的应用	153
4.2.3 值作为参数传递	88	5.5 指针类型——内存地址的描述	156
4.2.4 局部变量与全局变量	89	5.5.1 指针类型的定义	156
4.2.5 基于函数的过程式程序设计	91	5.5.2 指针类型的基本操作	158
4.3 标识符的作用域与变量的生存期	92	5.5.3 指针作为参数类型	162
4.3.1 程序的多模块结构	92	5.5.4 指针与动态变量	169
4.3.2 标识符的作用域	94	5.5.5 指针与数组	181
4.3.3 命名空间	99	5.5.6 函数指针	185
4.3.4 变量的生存期(存储分配)	101	*5.5.7 多级指针	188
*4.3.5 基于栈的函数调用的实现	104	5.6 引用类型——变量的别名	189
4.4 递归函数	106	5.6.1 引用类型的定义	190
4.4.1 什么是递归函数	106	5.6.2 引用作为参数类型	190
4.4.2 “分而治之”的程序设计	107	5.7 小结	193
4.4.3 递归与循环的选择	109	5.8 习题	194
4.4.4 递归函数应用实例	109	第6章 数据抽象——对象与类	198
4.5 标准函数库	111	6.1 概述	198
4.6 C++ 函数的进一步讨论	112	6.1.1 数据抽象与封装	198
4.6.1 内联函数	112	6.1.2 面向对象程序设计	201
4.6.2 带默认值的形式参数	114	*6.1.3 面向对象程序设计与过程式程序设计的对比	203
4.6.3 函数名重载	115	6.2 类	206
4.6.4 匿名函数——λ表达式	118	6.2.1 数据成员	206
4.7 小结	119	6.2.2 成员函数	207
4.8 习题	120		
第5章 复合数据的描述——构造数据类型	123		
5.1 枚举类型——自定义值集的数据描述	123		
5.1.1 枚举类型的定义	123		

6.2.3 成员的访问控制——信息隐藏	208	7.2.6 类之间的聚集关系	274
6.3 对象	211	7.3 消息 (成员函数调用) 的动态绑定	276
6.3.1 对象的创建	211	7.3.1 消息的多态性	276
6.3.2 对象的操作	212	7.3.2 虚函数与消息的动态绑定	277
6.3.3 this 指针	213	7.3.3 纯虚函数和抽象类	280
6.4 对象的初始化和消亡前处理	216	*7.3.4 虚函数动态绑定的一种实现	285
6.4.1 构造函数	216	7.4 多继承	286
6.4.2 析构函数	219	7.4.1 多继承的必要性	286
6.4.3 成员对象的初始化	222	7.4.2 多继承派生类的定义	288
6.4.4 拷贝构造函数	223	7.4.3 命名冲突	288
6.5 类作为模块	227	7.4.4 重复继承——虚基类	289
6.5.1 类模块的组成	227	7.5 小结	291
*6.5.2 Demeter 法则	228	7.6 习题	291
6.6 对象与类的进一步讨论	229	第 8 章 类属类型 (泛型) ——模板	295
6.6.1 对常量对象的访问——常 (const) 成员函数	229	8.1 概述	295
6.6.2 同类对象之间的数据共享——静态 (static) 数据成员	231	8.2 模板	296
6.6.3 提高对象私有数据成员的访问效率——友元 (friend)	233	8.2.1 函数模板	296
6.6.4 对象拷贝构造过程的优化——转移构造函数	237	8.2.2 类模板	299
6.6.5 操作符重载	238	8.2.3 模板的复用	301
6.7 小结	259	8.3 C++ 标准模板库	303
6.8 习题	260	8.3.1 概述	303
第 7 章 继承 (类的复用) ——		8.3.2 容器	305
派生类	264	8.3.3 迭代器	308
7.1 概述	264	8.3.4 算法	308
7.1.1 类之间的继承关系——基类与派生类	264	8.4 小结	314
7.1.2 继承的作用	264	8.5 习题	315
7.2 单继承	265	第 9 章 输入 / 输出	316
7.2.1 单继承派生类的定义	265	9.1 概述	316
7.2.2 在派生类中访问基类成员——protected 访问控制	266	9.2 面向控制台的输入 / 输出	317
7.2.3 派生类对基类成员的访问控制——继承方式	269	9.2.1 基于函数库的控制台 I/O	318
7.2.4 派生类对象的初始化和赋值操作	271	9.2.2 基于类库的控制台 I/O	320
7.2.5 单继承的应用实例	273	9.2.3 抽取 / 插入操作符 “>>” 和 “<<” 的重载	323
		9.3 面向文件的输入 / 输出	325
		9.3.1 文件概述	325
		9.3.2 基于函数库的文件 I/O	326
		9.3.3 基于类库的文件 I/O	332
		9.4 面向字符串变量的输入 / 输出	338
		9.5 小结	340
		9.6 习题	340

第 10 章 异常处理 341

10.1 概述 341

10.1.1 什么是异常 341

10.1.2 异常处理的基本手段 342

10.2 C++ 异常处理机制 343

10.2.1 try、throw 以及 catch 语句 343

10.2.2 异常的嵌套处理 346

10.3 基于断言的程序调试 347

10.4 小结 348

10.5 习题 349

第 11 章 Windows 应用程序设计 350

11.1 Windows 应用程序的基本
结构 350

11.1.1 Windows 应用程序简介 350

11.1.2 消息驱动的程序结构 351

11.2 面向对象的 Windows 应用程序
结构 354

11.2.1 Windows 应用程序中的对象 354

11.2.2 微软基础类库 (MFC) 355

11.2.3 基于 MFC “文档 - 视”结构
的应用框架 357

11.3 小结 359

11.4 习题 359

附录 A ASCII 字符集及其编码 360

附录 B IEEE 浮点数的内部表示 361

附录 C C++ 标准函数库中的常用
函数 363

附录 D C++ 编译预处理命令 365

附录 E C++ 标准模板库 (STL)
常用功能 370

附录 F MFC 一些常用类的功能 377

参考文献 382

CHAPTER 1

第1章

概述

自1946年第一台数字电子计算机(ENIAC)问世以来,计算机在理论、技术以及应用等方面都有了很大的发展。特别是计算机应用,它已从早期的数值计算应用拓广到现在大量的非数值计算应用,如:信息管理系统、文字处理系统、基于Internet的应用(如Web浏览器、电子邮件等)以及嵌入式应用系统(如家电的计算机控制)等。现在,计算机已经渗透到人类社会活动的各个领域并发挥着巨大的作用。

一台计算机由硬件和软件两部分构成。硬件是指计算机的物理构成,软件主要是指计算机程序(指令序列)。没有硬件就没有计算机,但如果只有硬件没有软件,那可以说计算机几乎什么事情也做不了。因此,要想用计算机来解决各种问题,必须要有相应的软件。

随着计算机应用领域不断扩大、应用层次不断加深,社会对计算机软件的需求急剧增长,从而导致软件的规模不断扩大、复杂程度不断提高。如何设计出大量的满足用户需求的高质量软件是软件工作者面临的严峻挑战。

本章从计算机的工作模型、程序设计的基本思想以及具体的程序设计语言三个方面,对与计算机程序设计相关的基本内容进行简单概述。

1.1 计算机的工作模型

计算机程序不同于其他程序(如会议程序、菜谱等),它是由计算机来执行的,因此,编制计算机程序(程序设计)通常要考虑到计算机解决问题的方式和特点,需要对计算机的工作模型有一定的了解。下面将分别从计算机的体系结构、硬件与软件以及计算机内部的信息表示等几个方面来介绍计算机的工作模型。

1.1.1 冯·诺依曼体系结构

虽然现在计算机的计算能力与早期的计算机相比已有了很大的提升,但是,目前计算机基本上采用的还是传统的冯·诺依曼体系结构。冯·诺依曼结构的计算机(简称为冯·诺依曼计算机)由存储、运算、控制、输入以及输出五个单元构成(如图1-1所示),其中,存储单元用于存储程序(指令序列)和数据^①;运算单元用于进行算术/逻辑运算;控制单元用于控制程序的执行和根据指令向其他单元发出控制信号;输入单元和输出单元作为计算机与外界接口,用于实现系统的输入和输出功能。

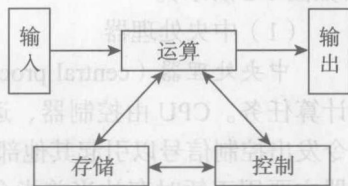


图1-1 冯·诺依曼体系结构

① 冯·诺依曼体系结构的计算机又称为存储程序式计算机,程序和数据存储在同一个存储器中。在冯·诺依曼体系结构出现之前,计算机存储单元只存储数据,而程序则是以一种外插的形式(如附加的电子线路板)接入系统。

冯·诺依曼计算机的工作过程：把待执行的程序装入存储单元中，控制单元从存储单元中逐条地取出和执行程序中的指令，把其中的运算指令交给运算单元完成；程序执行过程中从输入单元或存储单元获得所需要的数据；并将产生的临时结果保存在存储单元中，程序的最终执行结果通过输出单元输出。冯·诺依曼计算机的本质是通过不断地改变程序的状态来完成程序的功能，程序的状态由存储单元中的数据构成，状态的转换是通过程序中的指令对数据的操作来实现的。

冯·诺依曼计算机所能执行的指令通常有：

- 算术指令：进行加、减、乘、除等运算。
- 逻辑指令：比较两个数据的大小等逻辑运算。
- 数据传输指令：实现各单元之间的数据传输。
- 流程控制指令：指定下一条指令在存储单元中的地址。通常情况下，计算机从某个存储地址开始依次取指令来执行（顺序执行）。流程控制指令可以用来改变指令执行顺序，它包括转移、循环以及子程序调用等。

从冯·诺依曼计算机所能执行的指令可以看出，程序设计的任务是十分艰巨的，它要把各种应用问题落实到用一些简单的指令来解决！

1.1.2 硬件与软件

一台计算机由硬件和软件两部分构成。硬件是计算机的物质基础，软件是计算机的灵魂，两者相辅相成。从某种意义上讲，一台计算机的性能主要由硬件决定，而它的功能则主要由软件来提供。

1. 硬件

硬件（hardware）是指构成计算机的元器件和设备。计算机元器件的发展经历了电子管、晶体管、集成电路以及超大规模集成电路等几个阶段，其集成度和速度仍在不断提高。

目前，冯·诺依曼计算机的硬件一般按中央处理器、内存以及外围设备三个部分来组织，它们之间通过用于传递数据、地址和控制信号的总线来连接（如图 1-2 所示）。

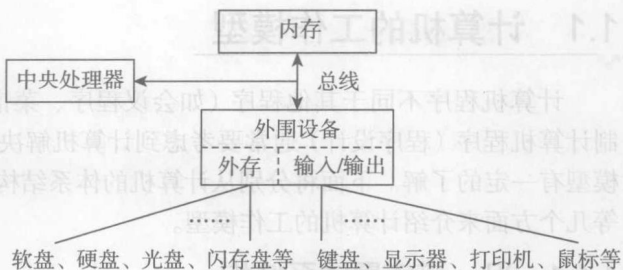


图 1-2 典型的计算机硬件组成

（1）中央处理器

中央处理器（central processing unit, CPU）是计算机的核心部件，它用于执行指令以完成计算任务。CPU 由控制器、运算器以及寄存器等构成。控制器负责从内存中取指令并根据指令发出控制信号以引起其他部件的动作。运算器执行运算指令所规定的算术和逻辑运算。寄存器主要用于暂时存放当前指令所需要的数据和计算结果（供后续指令使用）、记录当前指令的执行状态以及下一条指令的内存地址等，其主要作用是减少访问（存/取）内存中内容的次数，以提高指令的执行效率。

（2）内存

内存（memory）是内部存储器（或主存储器）的简称，它用于存储正在运行的程序和正