



重点大学 计算机基础课程教材

计算机程序设计基础

赵宏 主编

吴文虎 主审



清华大学出版社 • 北京交通大学出版社



TP311.1
40

重点大学计算机基础课程教材

计算机程序设计基础

赵 宏 主编

吴文虎 主审

清华大学出版社

北京交通大学出版社

• 北京 •

内 容 简 介

本书是根据教育部非计算机专业计算机基础课程教学指导分委员会提出的高等学校计算机基础课程教学基本要求组织编写的。

为适应新的人才培养的要求,结合程序设计技术和信息技术的发展,对程序设计基础课程教学内容和体系结构进行了重大改革。本书以 C/C++ 语言为背景,全面介绍了结构化程序设计和面向对象程序设计的基本原理和方法,内容由浅入深,循序渐进,每章都有丰富的例题和习题,并有配套的实验教材指导学生上机实践,以及丰富的数字化资源。

本书可作为高等学校各专业的程序设计基础课程教材,也可用于网络化教学及培训机构和读者自学程序设计课程的教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

计算机程序设计基础 / 赵宏主编. —北京:清华大学出版社;北京交通大学出版社, 2005.1
(重点大学计算机基础课程教材)

ISBN 7-81082-341-8

I. 计… II. 赵… III. 程序设计-高等学校-教材 IV. TP311

中国版本图书馆 CIP 数据核字(2004)第 090107 号

责任编辑:陈 芳

出版者:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京东光印刷厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:19.25 字数:474 千字

版 次:2005 年 1 月第 1 版 2005 年 9 月第 2 次印刷

书 号:ISBN 7-81082-341-8 / TP·134

印 数:5 001~9 000 册 定价:26.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@center.bjtu.edu.cn。

《重点大学计算机基础课程教材》

编 委 会

(排名不分先后)

吴文虎 (清华大学)

黄刘生 (中国科学技术大学)

叶晓风 (南京大学)

阮秋琦 (北京交通大学)

谢柏青 (北京大学)

郑 骏 (华东师范大学)

施伯乐 (复旦大学)

管会生 (兰州大学)

钱 能 (浙江工业大学)

谢步赢 (同济大学)

朱 敏 (东南大学)

汪 卫 (复旦大学)

杨小平 (中国人民大学)

李丽娟 (湖南大学)

王立福 (北京大学)

何炎祥 (武汉大学)

王行恒 (华东师范大学)

马建峰 (西安电子科技大学)

衷克定 (北京师范大学)

薛永生 (厦门大学)

出版说明

进入 21 世纪,随着国家信息化步伐的加快及各行业信息化进程的不断加速,社会对专业(非计算机专业)人才的信息技术能力要求越来越高。为了适应社会对专业人才的要求,全国各高校在重视专业知识培养的同时也非常注重计算机应用能力的训练,即信息技术能力的培养。计算机应用水平已成为衡量高校毕业生综合素质的突出标志之一。

为此,各高校加大了使用计算机科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向现代信息社会学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新其教学内容、改革课程体系,使学科专业的教育与社会信息化发展趋势相适应。计算机基础课程教学在改造传统学科向现代信息社会学科转变起到了至关重要的作用,学科专业中的计算机基础课程设置、内容体系和教学手段及方法等也具有不同于以前传统学科的鲜明特点。

为了配合各高校现代学科专业(非计算机专业)的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机基础课程教材。但是计算机基础教育的发展只有短短的二十多年时间,其覆盖的专业门类繁多,涉及的学校类型各异,不同的高校在开展计算机基础教育时还存在各自的认识。目前,非计算机专业的计算机课程教材的建设工作仍滞后于教学改革的实践,如:现有的计算机课程教材中有不少内容陈旧,重理论、轻实践,不能满足教学计划及课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等。这些都不利于学生自学能力的提高和全面素质的培养。可见,高等学校计算机基础教育和教材建设正面临新的形势和任务。

重点大学的教学与科研氛围是培养面向信息社会一流专业人才的基础,其中教材的使用和建设则是这种氛围的重要组成部分,一批具有特色优势的非计算机专业的计算机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非计算机专业上计算机教育的优势,同时以教材展示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,在相关教学指导委员会专家的指导和建议下,我们规划并组织出版了本系列教材,以满足非计算机专业计算机课程教学的需要。

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。

一、强调应用。本系列教材面向非计算机专业学生,从应用目的出发,强调计算机在各专业中的应用。在教材内容上坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

二、内容新颖。计算机科学和技术的发展日新月异,本系列教材力求介绍这一领域的新技术、新发展,放弃对一些过时的概念和使用价值较小的技术的介绍。教材涉及的计算机软件应具有典型性,在保持通用性的前提下介绍最新版本的特点。

三、体现案例教学。在兼顾基础性和系统性的前提下,重视教材内容的案例编排,力求从内容和结构上突出案例教学的要求,以适应教师指导下学生自主学习的教学模式。

四、实施精品战略，突出重点，保证质量。本系列教材规划的重点在公共基础课和专业基础课的教材建设；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出版，力求逐步形成精品教材；鼓励教师编写体现专业计算机教学内容和课程体系改革成果的教材。

五、依靠一线教师，择优落实。本系列教材的作者全部来自全国各重点大学的一线授课教师。在落实选题和作者时，引入竞争机制，通过申报和进行严格评审后再进行确定。书稿完成后认真实行审稿程序，确保出书质量。

计算机科学与技术的发展突飞猛进，本系列教材也应动态发展。在教材使用过程中，希望广大的读者积极地向我们提出意见与建议，我们将及时改正和更新。

《重点大学计算机基础课程教材》编委会

2005 年 7 月

前 言

从 20 世纪 70 年代的结构化程序设计, 到 90 年代的面向对象技术, 程序设计技术在不断发展。目前国内各高校开设的程序设计课程种类繁多, 选取的第一门程序设计课程也不相同。

教育部 2003 年颁发了计算机基础教育白皮书“关于进一步加强高校计算机基础教学的几点意见”, 在教育部 1997 年 155 号文件的基础上对计算机基础教学的目标重新定位, 提出了适应新形势的课程体系。按照教育部新课程体系的要求, 程序设计基础课程作为理工科大学生的必修课, 既要讲结构化程序设计, 也要讲面向对象程序设计。

本书以解决实际问题的程序设计思想作为出发点进行编写, 并选择 C/C++ 语言为背景进行讲解, 突出程序设计方法, 介绍常用的数组、链表等数据结构, 讲解了递归、递推、排序、查找等算法的设计。学习目标是理解和掌握程序设计的基本概念和方法, 具备运用程序设计语言、数据结构和算法进行基本的结构化程序设计和面向对象程序设计的能力。

本教材主要特色有以下几点。

一、教学内容既注重基础理论又突出实用性。突出结构化程序设计和面向对象程序设计的基本原理、概念和方法, 特别强调培养学生分析问题、解决问题和动手实践的能力, 重点突出, 精选例题和习题, 由浅入深逐步展开进行讲解。

二、充分体现教师的教学思想。参加编写的教师均是多年从事计算机基础课教学和科研工作的一线教师, 积累了一定的经验, 教材编写时突出程序设计方法, 把教师的教学思想融入教材中, 由浅入深、循序渐进, 语言和语法的讲解完全融会贯通在程序设计及实例中。

三、重点内容突出。舍弃一些次要内容, 如位运算、多级指针、多维数组与指针等, 使该教材通俗易懂。与该书配套的《计算机程序设计基础辅导与实验教程》中对每章都编写了补充阅读内容, 作为本书的扩展。

四、教学资源丰富。构建了包括教材、实验教材、电子教案、源程序文件包、网络教程等在内的立体化教学资源。

全书共 12 章, 其中前 8 章为结构化程序设计, 主要包括程序设计概述、程序设计初步知识、程序的控制结构、模块化程序设计、构造数据类型、指针、动态数据结构、文件; 第 9 章为结构化程序设计到面向对象程序设计的过渡; 最后 3 章围绕面向对象程序设计的封装性、继承性、多态性, 讲解了类和对象、构造和析构函数、静态成员、友元、继承与派生、多态性与虚函数等内容。

本书包含了大量的程序例子, 并附有运行结果。凡在程序开头带有程序名编号的, 都是完整的程序, 这些程序都在 Visual C++ 6.0 环境下调试通过。程序及电子教案等资源可以在北京交通大学出版社网站 <http://press.bjtu.edu.cn/> 的【FTP 下载专区】中查找下载。

全书由北京交通大学的 5 位教师集体编写完成。第 1、2、4、9、10 章由赵宏编写; 第 3、8 章由李会霞编写; 第 5、6 章由靳小燕编写; 第 7 章由鲍志斌编写; 第 11、12 章由翟高

寿编写。全书由赵宏统稿。

清华大学计算机系吴文虎教授以广博的知识，耐心细致地审阅了全稿，提出了许多宝贵的修改意见，在此表示衷心的感谢。

在编写此书时还得到了许多同事的关心和支持，在此表示深深的谢意。

由于时间仓促，作者水平有限，本教材编写中难免有不足和疏漏，欢迎读者提出宝贵意见和建议，以供再版时改进。

作 者

2005 年 1 月

目 录

第 1 篇 结构化程序设计

第 1 章 概述	1
1.1 计算机语言和计算机程序设计	1
1.1.1 计算机语言	1
1.1.2 计算机程序设计	2
1.2 程序设计方法	2
1.2.1 结构化程序设计基本思想	3
1.2.2 3 种基本结构	3
1.2.3 结构化程序设计的过程	4
1.2.4 用自顶向下、逐步细化的方法构造算法实例	8
1.3 程序的实现环境	10
1.3.1 硬件环境	10
1.3.2 软件环境	11
1.4 C 语言与 C++ 的发展	12
1.5 本章小结	13
思考题	13
习题	13
第 2 章 程序设计初步	14
2.1 程序的基本结构	14
2.1.1 简单程序: 输出一行文本	14
2.1.2 简单程序: 计算两个整数的和	15
2.1.3 简单程序: 用多函数方法计算两个整数的和	16
2.1.4 C/C++ 语言结构化程序的基本结构	17
2.2 数据类型	19
2.2.1 基本数据类型	19
2.2.2 常量	20
2.2.3 变量和内存的概念	23
2.2.4 变量定义和赋初值	24
2.3 算术运算符和算术表达式	25
2.4 赋值运算符和赋值表达式	26
2.5 数据类型的转换	27
2.6 数据的输出和输入	28

2.6.1	格式输出函数	28
2.6.2	格式输入函数	30
2.7	数学函数	32
2.8	简单程序设计	32
2.9	本章小结	35
	思考题	35
	习题	35
第3章	程序控制结构	37
3.1	选择结构程序设计	37
3.1.1	引例	37
3.1.2	关系运算和逻辑运算	38
3.1.3	if 选择结构	40
3.1.4	switch 多分支选择结构	46
3.1.5	程序设计举例	48
3.2	循环结构程序设计	49
3.2.1	引例	49
3.2.2	自增和自减运算符	51
3.2.3	while 循环和 do-while 循环	52
3.2.4	for 循环	54
3.2.5	循环嵌套	56
3.2.6	break 和 continue 语句	57
3.2.7	循环应用举例	59
3.3	本章小结	62
	思考题	62
	习题	63
第4章	模块化程序设计	64
4.1	模块化程序设计概述	64
4.2	函数的声明、定义和调用	65
4.2.1	函数说明	66
4.2.2	函数定义	66
4.2.3	函数调用及参数的传递	67
4.2.4	带自定义函数的程序设计	70
4.3	函数的多级调用	73
4.3.1	嵌套调用	73
4.3.2	递归调用	75
4.3.3	递归与递推	83
4.4	变量的作用域和存储类别	84
4.4.1	局部变量及存储类别	85
4.4.2	全局变量及存储类别	87

4.5	计算机随机模拟方法	91
4.5.1	伪随机数的产生	92
4.5.2	蒙特卡罗方法	94
4.6	编译预处理	95
4.7	本章小结	99
	思考题	99
	习题	99
第5章	构造数据类型	102
5.1	数组概述	102
5.2	一维数组	102
5.2.1	引例	102
5.2.2	一维数组的定义	103
5.2.3	数组的初始化	104
5.2.4	数组的引用	104
5.3	数组做函数的参数	105
5.4	一维数组的应用	106
5.4.1	排序	106
5.4.2	数据查找	111
5.4.3	数据统计分析	114
5.5	多维数组	116
5.5.1	二维数组的定义	116
5.5.2	二维数组的初始化	116
5.5.3	二维数组的引用	117
5.5.4	二维数组元素在内存中的排列顺序	118
5.5.5	多维数组举例	118
5.6	字符数组与字符串	119
5.6.1	字符数组的定义及初始化	119
5.6.2	用字符数组存放字符串	120
5.6.3	字符数组的输入输出	120
5.6.4	用于字符串处理的函数	123
5.6.5	应用举例	126
5.7	结构体类型	127
5.7.1	结构体类型的定义	128
5.7.2	结构体变量的定义和引用	129
5.7.3	结构体变量的初始化	131
5.7.4	结构体应用举例	131
5.8	共用体类型	133
5.8.1	共用体类型的定义	133
5.8.2	共用体类型变量的定义和引用	134

5.8.3 共用体应用举例	134
5.9 本章小结	135
思考题	135
习题	136
第6章 指针	137
6.1 变量的地址和指针	137
6.1.1 数据在内存中的存储	137
6.1.2 访问变量的方式	137
6.1.3 指针变量	137
6.2 指针变量的定义及引用	138
6.2.1 指针变量的定义	138
6.2.2 指针变量的引用	138
6.3 指针与函数	141
6.3.1 指针作为函数参数	141
6.3.2 函数返回指针	142
6.3.3 函数指针	143
6.4 指针与数组	145
6.4.1 指向数组的指针	145
6.4.2 引用数组元素的方法	145
6.5 字符指针	147
6.5.1 字符串的表示形式	147
6.5.2 程序设计举例	148
6.6 指针数组	152
6.6.1 指针数组的概念	152
6.6.2 指针数组的初始化	153
6.7 本章小结	155
思考题	156
习题	156
第7章 动态数据结构	158
7.1 从静态数据结构到动态数据结构	158
7.2 动态内存分配	158
7.2.1 ANSI C 中用于动态操作的标准函数	158
7.2.2 C++中用于动态操作的运算符——new 和 delete	162
7.3 链表	164
7.3.1 链表的定义	164
7.3.2 链表的建立	165
7.3.3 链表结点的插入	167
7.3.4 链表结点的删除	171
7.3.5 循环链表	173

7.4 本章小结	173
思考题	174
习题	174
第8章 文件	176
8.1 文件概述	176
8.1.1 文件的概念和分类	176
8.1.2 文件的操作过程	177
8.1.3 缓冲区和文件指针	177
8.2 文件的打开和关闭	178
8.2.1 文件的打开	178
8.2.2 文件的关闭	179
8.3 文件的顺序操作	179
8.3.1 文件的字符读写函数	179
8.3.2 文件的字符串读写函数	182
8.3.3 文件的数据块读写函数	184
8.3.4 文件的格式化读写函数	186
8.4 文件的随机操作	187
8.4.1 文件定位	187
8.4.2 文件的随机读写	188
8.5 文件的检测	189
8.6 本章小结	190
思考题	190
习题	190

第2篇 面向对象程序设计

第9章 从结构化程序设计到面向对象程序设计	191
9.1 C++对C的改进和扩充	191
9.1.1 输入输出流	191
9.1.2 函数的重载	193
9.1.3 带默认参数的函数	195
9.1.4 内联函数	196
9.1.5 一元作用域运算符	197
9.1.6 引用与函数	197
9.2 面向对象的程序设计方法	200
9.2.1 面向对象的概念	200
9.2.2 类和对象	201
9.2.3 面向对象程序设计的特性——封装性、继承性和多态性	204
9.3 本章小结	205
思考题	206

习题	206
第 10 章 类和对象	207
10.1 从结构到类	207
10.2 类和成员函数的定义	209
10.2.1 类的定义	209
10.2.2 在类外定义成员函数	210
10.2.3 接口与实现方法的分离	210
10.3 对象的定义与成员的访问	212
10.3.1 对象的定义	212
10.3.2 对象成员的访问	212
10.3.3 this 指针	213
10.4 构造函数和析构函数	214
10.4.1 构造函数	214
10.4.2 析构函数	216
10.4.3 默认构造函数和默认析构函数	217
10.4.4 拷贝构造函数	218
10.5 静态成员	220
10.5.1 静态数据成员	220
10.5.2 静态成员函数	223
10.6 友元	224
10.6.1 友元函数	224
10.6.2 友元类	225
10.7 对象数组	226
10.8 复合类	227
10.9 本章小结	229
思考题	230
习题	230
第 11 章 继承与派生类	231
11.1 继承与派生类概述	231
11.2 单一继承	232
11.2.1 派生类的定义与实现	232
11.2.2 单一继承与访问控制	235
11.2.3 继承类别及派生类对基类成员的访问权	237
11.3 多重继承	239
11.3.1 派生类的定义与实现	239
11.3.2 多重继承与访问控制	240
11.3.3 虚基类	244
11.4 综合实例	251
11.5 本章小结	254

思考题	254
习题	254
第 12 章 多态性	255
12.1 多态性概述	255
12.2 函数重载与歧义性	255
12.3 运算符重载	256
12.3.1 运算符重载意义及要领	256
12.3.2 运算符重载实现方式	256
12.3.3 特殊运算符重载	259
12.3.4 类类型转换	268
12.4 虚函数	269
12.4.1 赋值兼容原则及虚函数的引入	269
12.4.2 虚函数的定义及使用	274
12.4.3 在基类构造函数与析构函数中调用虚函数	276
12.4.4 多重继承与虚函数	278
12.5 纯虚函数与抽象类	280
12.6 本章小结	282
思考题	282
习题	282
附录 A 运算符的优先级和结合性	284
附录 B ASCII 字符集	286
附录 C 常用库函数	287
参考文献	290

第 1 篇 结构化程序设计

第 1 章 概 述

本章涉及的内容非常广泛，首先回顾了计算机语言的发展，讲述了计算机程序设计的基本概念，然后介绍面向过程程序设计的基本方法，详细讲述了程序实现的软、硬件环境以及程序上机实现的一般步骤等内容。通过这些内容的学习，可以对程序设计的概貌有一个了解，为以后各章的学习打下基础。

1.1 计算机语言和计算机程序设计

1.1.1 计算机语言

计算机是人们处理信息的一种重要工具，它在人的控制下，按照人的意志正确地工作。人给机器一个指令，机器就完成一个操作。如果把一系列指令输入计算机存储起来，计算机就能按照指令序列实现操作的自动化。人和计算机之间的这种通信必须使用人工设计的语言，即计算机语言。

当今使用的计算机语言大致可以分为 3 类：机器语言、汇编语言和高级语言。

1. 机器语言

本质上计算机只能识别“0”和“1”这样的二进制信息，机器语言的程序全部由“0”和“1”表示出来，例如，一个 16 位的计算机，由 16 个二进制数组成一条指令，这些指令叫机器指令。16 个 0 和 1 可以组成 2^{16} 个不同的指令或信息，这些指令的集合叫机器语言。机器语言是计算机能直接识别和执行的惟一语言。

人们用这种语言编写出的程序非常烦琐，而且不论阅读程序还是调试程序都非常困难。另外，机器语言是与机器有关的，特定的机器语言只能用在特定的一类机器上，不是通用的。

2. 汇编语言

为了克服机器语言的缺点，人们用一些特殊的符号（即助记符）来表示机器指令，例如用 ADD 代表加，用 SUB 代表减。这些助记符的使用在一定程度上提高了汇编语言的可读性。汇编语言的语句与计算机硬件操作有一一对应关系，每种汇编语言都是支持这种汇编语言的计算机所独有的。下面的汇编源程序计算 a、b 的和并存储在 a 中：

LOAD a, 4	(将地址 4 中的数据取出放到变量 a 中)
LOAD b, 5	(将地址 5 中的数据取出放到变量 b 中)

ADD a, b	(对 a、b 求和放到 a 中)
STORE a, 6	(将 a 中数据写到地址 6 中)

汇编语言出现后计算机的用途迅速扩大,但基本上有多少种计算机就有多少种汇编语言,因此汇编语言同机器语言一样也是面向机器的,通用性较差。尽管如此,汇编语言一直被人们所使用,主要是由于其执行速度快、占用存储空间小、对硬件操作灵活等特性。

3. 高级语言

为加速程序开发的进程,人们创造出了高级语言,它非常接近于人类的自然语言和数学语言,它的一个语句往往对应几条机器指令。用高级语言编写计算 a、b 的和并存储在 c 中的语句如下:

c=a+b;

一般说高级语言不再是面向机器的了,而是面向过程的语言,即把解题过程的每一步用计算机语言的语句描述出来,再配上适当的语言处理程序,计算机就能执行。因此,这种语言也称“算法语言”。高级语言有很多种,如 Fortran、Basic、Pascal、C、C++、Java 等,不管哪种高级语言,源程序都必须经过相应的语言处理程序翻译成机器指令才能执行。

1.1.2 计算机程序设计

程序并不是计算机程序中独有的概念,在日常生活中我们也常见到这个词,例如一个会议的日程、一场演出的节目单等,这些程序都是由人的一项一项活动组成,有序地完成每一项活动也就实现了程序的目标。

计算机工作方式有两种,一种是交互式的,即人给机器一个指令,机器就完成一个操作;另外一种程序控制式的,即把计算机要完成的操作一条一条指令按序排好,计算机一步一步执行了这个指令序列,也就完成了我们希望它实现的功能,而且整个指令序列执行过程中不需要人来干预。为了解决某一特定问题用某一种计算机语言编写的指令序列称为程序。程序是程序设计的结果,在执行程序前必须先排好程序,因此排定以时间为进程的必须完成的各种操作叫程序设计。

在用高级语言进行程序设计时要注意以下 3 个概念。

(1) 语法。每种计算机语言都有自己的语法规则,这些规则是非常严格的。在进行编译时系统会按语法规则严格检查程序,如有不符合语法规则的地方,计算机就会显示语法有错信息。

(2) 语义。即某一语法成分的含义。例如, C 语言中用“int”定义整型变量,用“char”定义字符型变量,用“while”语句实现循环,用“+”表示加法,用“!=”表示不等运算等。在使用时必须了解每一种语法成分的正确含义。

(3) 语用。即正确使用语言。要善于利用语法规则中的有关规定和语法成分的含义有效地组成程序,以达到特定的目的。

1.2 程序设计方法

在程序设计发展过程中,特别是在 20 世纪 70 年代初期,各种大型的、复杂的软件系统