

高等学校计算机教育规划教材



普通高等教育“十一五”

国家级规划教材

程序 设计基础

何炎祥 石莹 王娜 编著

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

清华大学出版社





普通高等教育“十一五”国家级规划教材

高等学校计算机教育规划教材

程序 设计基础

何炎祥 石莹 王娜 编著

清华大学出版社

北京

内 容 简 介

本书主要介绍计算机程序设计基础方面的知识,内容包括:程序设计的基本概念、程序设计的发展、程序设计风格,程序设计语言、程序的结构、典型的数据结构、常用算法及算法描述,程序正确性证明,结构化程序设计、面向对象程序设计、网络程序设计、分布式程序设计,程序设计与平台的关系、程序的编译过程等。

本书取材新颖,内容完整,系统性强,注重实用。适合作为大专院校的教科书,也可供从事IT领域工作的科技人员参考。

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

程序设计基础 / 何炎祥, 石莹, 王娜编著. —北京: 清华大学出版社, 2006.11

(高等学校计算机教育规划教材)

ISBN 7-302-13370-0

I. 程… II. ①何… ②石… ③王… III. 程序设计-高等学校-教材 IV. TP311.1

中国版本图书馆 CIP 数据核字 (2006) 第 077346 号

出版者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社总机: 010-62770175 客户服务: 010-62776969

责任编辑: 张瑞庆

印刷者: 清华大学印刷厂

装订者: 三河市金元印装有限公司

发行者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 28 字数: 728 千字

版 次: 2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

书 号: ISBN 7-302-13370-0/TP · 8391

印 数: 1 ~ 5000

定 价: 35.00 元

编 委 会

名誉主任：陈火旺

主 任：何炎祥

副 主 任：王志英 杨宗凯 卢正鼎

委 员：（按姓氏笔画为序）

王更生 王忠勇 刘先省 刘腾红 孙俊逸

芦康俊 李仁发 李桂兰 杨健雷 陈志刚

陆际光 张焕国 张彦铎 罗 可 金 海

钟 珞 贲可荣 胡金柱 徐 苏 康立山

薛锦云

丛书策划：张瑞庆 汪汉友

本书主审：薛锦云

序 言

P R E F A C E

随着信息社会的到来,我国的高等学校计算机教育迎来了大发展时期。在计算机教育不断普及和高等教育逐步走向大众化的同时,高校在校生的人数也随之增加,就业压力随之加大。灵活应用所学的计算机知识解决各自领域的实际问题已经成为当代大学生必须具备的能力。为此,许多高等学校面向不同专业的学生开设了相关的计算机课程。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。抓好计算机专业课程以及计算机公共基础课程的教学,是提高计算机教育质量的关键。现在,很多高等学校除计算机系(学院)外,其他系(学院)也纷纷开设了计算机相关课程,在校大学生也必须学习计算机基础课程。为了适应社会的需求,满足计算机教育的发展需要,培养基础宽厚、能力卓越的计算机专业人员和掌握计算机基础知识、基本技能的相关专业的复合型人才迫在眉睫。为此,在进行了大量调查研究的基础上,通过借鉴国内外最新的计算机科学与技术学科和计算机基础课程体系的研究成果,规划了这套适合计算机专业及相关专业人才培养需要的、适用于高等学校学生学习的《高等学校计算机教育规划教材》。

“教育以人为本”,计算机教育也是如此,“以人为本”的指导思想则是将“人”视为教学的主体,强调的是“教育”和“引导”,而不是“灌输”。本着这一初衷,《高等学校计算机教育规划教材》注重体系的完整性、内容的科学性和编写理念的先进性,努力反映计算机科学技术的新技术、新成果、新应用、新趋势;针对不同学生的特点,因材施教、循序渐进、突出重点、分散难点;在写作方法上注重叙述的逻辑性、系统性、适用性、可读性,力求通俗易懂、深入浅出、易于理解、便于学习。

本系列教材突出计算机科学与技术学科的特点,强调理论与实践紧密结合,注重能力和综合素质的培养,并结合实例讲解原理和方法,引导学生学会理论方法的实际运用。

本系列教材在规划时注重教材的立体配套,教学资源丰富。除主教材外,还配有电子课件、习题集与习题解答、实验上机指导等辅助教学资源。有些课程将开设教学网站,提供网上信息交互、文件下载,以方便师生的教与学。

《高等学校计算机教育规划教材》覆盖计算机公共基础课程、计算机应用技术课程和计算机专业课程。既有在多年教学经验和教学改革基础上新编著的

教材，也有部分已经出版教材的更新和修订版本。这套教材由国内三十余所知名高校从事计算机教学和科研工作的一线教师、专家教授编写，并由相关领域的知名专家学者审读全部书稿，多数教材已经经受了教学实践的检验，适用于本科教学，部分教材可用于研究生学习。

我们相信通过高水平、高质量的编写和出版，这套教材不仅能够得到大家的认可和支持，也一定能打造成一套既有时代特色，又特别易教易学的高质量的系列教材，为我国计算机教材建设及计算机教学水平的提高，为计算机教育事业的发展和高素质人才的培养作出我们的贡献。

《高等学校计算机教育规划教材》编委会

程序设计是高等学校计算机及相关专业一门重要的核心课程,任何理论或者应用,只要是在计算机中实现,最终必须依靠程序,所以程序设计的学习非常重要。程序设计是一门基础学科,也是一门综合学科,之所以称其为基础学科是因为程序设计的课程一般安排在大学本科或者专科的前两年;称其为综合学科是因为程序设计中包含了数学、统计学、计算机组成原理、操作系统、数据结构、算法设计、编译原理等知识,如果从更具体的角度看,则包含更多的领域知识,如网络程序设计还必须熟悉网络构架及通信的基本流程,嵌入式程序设计则必须了解嵌入式环境等。

正因为程序设计的综合性,如何去学习是一个难题。对于程序设计的初学者,用一种具体的语言进行教学是比较通用的办法,因为学习一门具体的语言比较快,而且会增加学习者的信心。但同时也存在一些问题,初学者容易受某一种程序设计语言的具体技术影响,而忽略了程序设计的整体思想和方法。计算机科学与技术是一门快速发展的学科,各种语言的更新换代非常快,所以学习程序设计的重点应该放在程序设计的方法和思想上。程序设计从本质上可以归结为四大要素:问题、概念、逻辑和技巧,而语言的学习则是更多地停留在技巧的层面上。为了更好地、有意识地学习和掌握程序设计的概念和思维方法,我们提倡初学者以全面系统地学习程序设计的基础知识为主,同时以一种具体的程序语言作为工具来学习。这也是我们编写这本教材的初衷。

全书分为 12 章。

第 1 章绪论,介绍程序设计的一些基本概念和基础知识,包括程序设计的应用、程序设计的基本概念、程序设计的环境、程序设计的发展以及程序设计的要素等。

第 2 章程序的结构,是进一步学习编程的基础,掌握好本章的内容,对深入学习后面的编程知识具有重要的意义。内容包括程序的基本结构、程序语言的字符集、基本数据类型、常量和变量、表达式等基本概念,以及典型的逻辑结构。

第 3 章程序设计语言翻译基础,每一种程序语言因为其语法不同而有不同的翻译规则。学习语言的翻译基础有助于深入理解和掌握程序语言的语法和语义,了解程序的执行过程。

第 4 章算法,是程序设计的重要内容,介绍算法的概念、算法的常见描述方法、算法的正确性和复杂度分析,以及几种经典的算法设计方法。

第5章数据类型，讨论数据的存储、数据的表示以及常用的数据类型，为了不与普通的数据结构教材内容重叠，本章重点从程序设计的角度进行讨论。

第6章程序设计中常用的数据结构，介绍数据结构的基本概念以及几种在程序设计中常用的数据结构，如链表、队列、树等。

第7章堆栈与递归程序设计，主要介绍堆栈的定义、应用，以及递归的定义、递归程序的设计方法。

第8章输入输出，介绍流和文件的概念及读写、数据库的概念、数据库在程序设计中的应用。

第9章程序正确性证明，介绍程序测试、数理逻辑基础、部分正确性证明、终止性证明、结构化程序的正确性证明和递归程序的正确性证明。

第10章结构化程序设计，介绍结构化分析/结构化设计编程思想及结构化程序设计思想，同时介绍函数的定义、函数的调用、作用域规则，以及库函数、编译预处理等概念。

第11章面向对象程序设计，介绍面向对象程序设计的内容，面向对象的思想产生和基本概念，以及面向对象建模的概念和常用的面向对象建模技术，如OOD方法、OMT方法和UML建模语言等，还将介绍面向对象编程的知识、构件、分布对象技术及分布式对象中间件等概念。

第12章网络程序设计，网络程序设计是在网络环境下的程序设计，与单机程序设计不同，网络程序设计需要网络和通信知识作为背景，需要了解基本的网络原理，涉及更多通信及消息传递的知识。这也是本书将网络程序设计作为单独一章介绍的原因。

本书取材新颖，内容丰富，系统性强，注重实用。适合作为高校学生学习程序设计的教科书，也可供从事IT领域工作的广大科技人员参考。

由于作者的水平所限，书中难免存在错误与不妥之处，诚恳地希望读者批评指正。

作者

于武汉大学计算机学院

目 录

CONTENTS

第 1 章 绪论	1
1.1 程序设计的应用	1
1.1.1 科学计算	2
1.1.2 信息处理	3
1.1.3 计算机辅助系统	4
1.1.4 计算机控制与仿真系统	5
1.1.5 人工智能	5
1.2 程序设计的基本概念	6
1.2.1 程序设计的本质	6
1.2.2 问题求解的基本流程	8
1.3 程序设计的环境	10
1.3.1 程序设计与编程环境的关系	10
1.3.2 程序设计环境的发展	11
1.4 程序设计的发展	11
1.4.1 程序设计简史	12
1.4.2 程序设计的发展趋势	14
1.5 程序设计的要素	15
1.5.1 程序设计的辅助知识	15
1.5.2 标准化基础知识	16
1.5.3 良好的程序设计风格	17
1.6 本书所用的工具语言	18
1.6.1 C 程序设计语言简述	18
1.6.2 Java 程序设计语言简述	19
小结	20
习题	20
第 2 章 程序的结构	22
2.1 程序的基本结构	22
2.2 程序语言的字符集	25
2.2.1 字符集的概念	25
2.2.2 关键字	26

2.3	常量和变量	26
2.3.1	常量	26
2.3.2	变量	27
2.4	基本数据类型	28
2.4.1	数值类型	28
2.4.2	字符型	30
2.4.3	布尔型	31
2.4.4	枚举型	32
2.5	表达式	34
2.5.1	运算符	34
2.5.2	赋值表达式	37
2.5.3	输入输出语句	38
2.6	典型的逻辑结构	41
2.6.1	顺序	41
2.6.2	条件	41
2.6.3	循环	46
2.6.4	goto 语句	53
	小结	54
	习题	55
第3章	程序设计语言翻译基础	58
3.1	什么是编译程序	58
3.2	与编译器相关的程序	60
3.3	编译过程概述	62
3.3.1	词法分析	62
3.3.2	语法分析	63
3.3.3	语义分析	63
3.3.4	中间代码生成	64
3.3.5	代码优化	64
3.3.6	目标代码生成	64
3.3.7	符号表管理	65
3.3.8	出错处理	65
3.4	编译器中的主要数据结构	65
3.5	编译器结构中的其他问题	67
3.6	编译运行时的环境	69
3.6.1	程序执行时的存储器组织	70
3.6.2	完全静态运行时环境	71
3.6.3	基于栈的运行环境	72
3.6.4	动态存储器	73
3.7	参数传递机制	74

3.7.1	值传递	75
3.7.2	引用传递	75
3.7.3	值结果传递	75
3.7.4	名字传递	75
3.8	并行编译基础	76
3.8.1	并行计算机	76
3.8.2	并行编译系统的结构	77
	小结	78
	习题	78
第4章	算法	80
4.1	算法的概念	80
4.1.1	算法设计在程序设计中的角色	80
4.1.2	算法的特征	81
4.1.3	算法的复杂度	81
4.2	算法的描述	82
4.2.1	用自然语言描述算法	82
4.2.2	用流程图描述算法	83
4.2.3	用 N-S 图描述算法	84
4.2.4	用 PAD 图描述算法	84
4.2.5	用伪代码描述算法	85
4.3	算法分析	89
4.3.1	算法的正确性	89
4.3.2	算法的复杂度	91
4.4	P 与 NP 问题简介	94
4.5	算法设计	97
4.5.1	回溯法	97
4.5.2	分支限界法	102
4.5.3	贪婪法	105
4.5.4	分而治之法	110
4.5.5	动态规划法	116
4.5.6	局部搜索法	123
	小结	126
	习题	127
第5章	数据类型	130
5.1	程序设计中的数据对象	130
5.2	数据类型的基本概念	132
5.2.1	数据类型的描述	132
5.2.2	数据类型的实现	133
5.2.3	声明	134

5.2.4	类型检查和类型转换	136
5.3	构造数据类型	138
5.3.1	数组	138
5.3.2	指针	148
5.3.3	记录（结构体）	163
5.4	抽象数据类型	166
5.4.1	数据的抽象	167
5.4.2	抽象数据类型的定义	168
5.4.3	抽象数据类型的实现	170
小结		173
习题		173
第 6 章	程序设计中常用的数据结构	177
6.1	数据结构的基本概念	177
6.1.1	数据的逻辑结构	177
6.1.2	数据的存储结构	178
6.1.3	数据的运算	179
6.2	链表	180
6.2.1	结点	181
6.2.2	链表的构造	182
6.2.3	链表的相关运算	184
6.2.4	双链表和循环链表	187
6.3	队列	189
6.3.1	队列的定义	189
6.3.2	队列的运算	191
6.3.3	队列在程序设计中的应用	193
6.4	树	193
6.4.1	树的定义	193
6.4.2	二叉树的概念	194
6.4.3	二叉树的遍历	197
6.4.4	线索二叉树	200
6.4.5	二叉树在程序设计中的应用	203
小结		205
习题		206
第 7 章	堆栈与递归程序设计	208
7.1	堆栈	208
7.1.1	堆栈的定义	208
7.1.2	堆栈的应用	211
7.2	递归程序设计	216
7.2.1	递归的定义	216

7.2.2	递归程序的设计	219
7.2.3	递归函数的调用与堆栈	228
7.2.4	递归程序的性能	229
7.2.5	非递归化	231
	小结	232
	习题	233
第8章	输入输出	234
8.1	流和文件	234
8.1.1	流和文件的概念	234
8.1.2	流和文件的读写	235
8.2	数据库	244
8.2.1	数据库的概念	244
8.2.2	数据库在程序设计中的应用	246
	小结	266
	习题	267
第9章	程序正确性证明	269
9.1	概述	269
9.2	程序测试	271
9.2.1	黑盒测试	271
9.2.2	白盒测试	275
9.3	数理逻辑基础	277
9.3.1	命题逻辑	278
9.3.2	谓词逻辑	279
9.4	部分正确性证明	281
9.4.1	不变式断言法	281
9.4.2	子目标断言法	286
9.4.3	公理化方法	289
9.5	终止性证明	294
9.5.1	良序集法	294
9.5.2	计数器法	298
9.6	结构化程序的正确性证明	300
9.6.1	结构化程序	300
9.6.2	正确性定理	302
9.6.3	证明程序正确性的代数方法	304
9.6.4	产生循环不变式的一种方法	311
9.7	递归程序的正确性证明	313
9.7.1	递归程序的模型	313
9.7.2	结构归纳法	315
9.7.3	良序归纳法	316

小结	317
习题	318
第 10 章 结构化程序设计	321
10.1 结构化分析/结构化设计编程思想	321
10.1.1 结构化分析/结构化设计思想简介	321
10.1.2 结构化程序设计思想简介	327
10.2 函数的定义	329
10.3 函数的调用	333
10.3.1 函数调用的一般格式	333
10.3.2 嵌套	337
10.3.3 递归	338
10.4 作用域规则	340
10.5 库函数	346
10.5.1 标准库函数	346
10.5.2 自定义库函数	347
10.6 编译预处理	348
10.6.1 宏	348
10.6.2 编译	351
小结	353
习题	354
第 11 章 面向对象程序设计	356
11.1 面向对象的思想	356
11.1.1 面向对象的思想及其产生过程	356
11.1.2 面向对象的基本概念	359
11.2 面向对象建模	362
11.2.1 面向对象建模的概念	362
11.2.2 面向对象分析	363
11.2.3 面向对象设计	365
11.2.4 常用的面向对象建模技术	366
11.3 面向对象程序设计	376
11.3.1 面向对象程序语言	376
11.3.2 程序语言中的 OOP 机制	378
11.4 面向对象技术的发展	393
11.4.1 构件	393
11.4.2 分布式对象中间件	396
11.4.3 基于面向对象技术的应用软件体系结构	397
11.4.4 并发面向对象技术	398
11.5 面向对象技术面临的问题	400
小结	400
习题	401

第 12 章 网络程序设计	404
12.1 网络程序设计的发展.....	404
12.1.1 联机多用户程序设计.....	404
12.1.2 对等网络程序设计.....	404
12.1.3 客户机/服务器程序设计.....	405
12.1.4 远程过程调用程序设计.....	405
12.1.5 浏览器/服务器程序设计.....	406
12.1.6 分布式程序设计.....	406
12.2 UNIX 环境网络编程.....	406
12.2.1 基本概念.....	406
12.2.2 socket 编程.....	409
12.2.3 TCP 和 UDP 协议编程.....	413
12.3 Windows 环境下 socket 编程.....	419
12.3.1 Winsock 套接字.....	420
12.3.2 Windows Sockets 编程.....	420
小结.....	425
习题.....	425
参考文献.....	427

第 1 章

绪论

程序设计是按照一定的规则，使用计算机指令控制计算机为人们提供服务的过程。从 1946 年第一台数字电子计算机诞生以来，程序设计伴随着计算机科学的发展不断进步，特别是 20 世纪 50 年代第一代高级程序设计语言出现后，出现了成千上万种程序设计语言和各种各样的程序设计方法：从 0 和 1 组成的机器语言到 FORTRAN、ALGOL、COBOL，到 C、PASCAL、Java；从结构化程序设计到面向对象程序设计，到构件。

这些程序语言和方法丰富了程序设计的内容，使程序设计的应用更加广泛，同时也增加了学习程序设计的困惑：无论打开哪一个搜索引擎搜索“程序设计”，都会出现多种不同的关键词，如网络程序设计、汇编程序设计、并行程序设计、极限程序设计、嵌入式程序设计……众多的概念使人们不知道从哪里开始学习；刚学习完了 C 程序设计又出现了 C# 语言，刚学会 ASP 又开始使用 ASP. Net，快速的更新使人们不知道如何追赶。实际上，所有的计算机程序语言以及方法的最终目的都是一样的，就是良好地控制计算机按照人们的意愿去工作，共同的目的使各种各样的语言和方法有了共同的知识基础，所以在学习程序设计的时候，只要首先掌握程序设计的基础方法和规则，就能够触类旁通、举一反三。

本章将介绍程序设计的一些基本概念和知识，后续章节将详细讨论程序设计的各个方面，目的是分析程序设计的本质和特性，从更抽象的角度介绍程序设计知识，并不限于某种特殊的语言或者方法，当然在这个过程中会用一些具体的例子来使这些知识容易理解。

1.1 程序设计的应用

学习任何一门学科之前，首先要明确学习它的意义，而了解程序设计的应用是了解程序设计的意义的最好方法。

计算机学科是一门与应用紧密结合的学科，计算机学科的发展扩展了其应用范围，而计算机的应用也促进了学科自身的发展。计算机学科和数学、物理学、统计学、逻辑学等学科紧密相连，可以说计算机学科是一门包容性强的综

合性学科，而无论计算机应用在哪个方面，往往还是需要通过程序的运行来实现，从某个角度说，计算机的应用也就是程序设计的应用，而随着手机和一些嵌入式设备的普及，程序设计的应用已经不仅仅限于传统的计算机。本节将从多个方面介绍程序设计的应用。

1.1.1 科学计算

科学计算是程序设计最早也是最重要的应用领域。实际上，在计算机技术发展的初期，计算机主要用于军事领域，操作计算机的也都是经过训练的专家。“科学计算”一词首先出现在 1983 年，由美国国防部、能源部、国家科学基金会以及一个由美国著名数学家拉克斯 (P.Lax) 为首的不同学科的专家委员会向美国政府提出的报告中，强调“科学计算是关系到国家安全、经济发展和科技进步的关键性环节，是事关国家命脉的大事。”当时轰动美国朝野，总统科学顾问随即到国会作证，政府迅速采取措施。

1984 年美国大幅度增加科学计算经费，美国国家科学基金会 (NSF) 建立了“先进科学计算办公室” (OASC)，制定全面高级科学计算发展规划，连续 5 年累计拨款 2.5 亿美元。新建成 5 个国家超级计算中心 (分别在普林斯顿大学、圣地亚哥、伊利诺大学、康奈尔大学和匹兹堡)，配备当时最高性能的计算机，建立 NSFnet 新网络。

1987 年起 NSF 把“科学与工程计算”、“生物工程”和“全局性的科学”作为三大优先重点支持的领域。

1990 年美国国家研究委员会发表了《振兴美国数学：90 年代的计划》，建议对由计算引发的数学给予特殊的鼓励和资助。因为这种数学将成为有效地使用已在运转的或已设计好的许多超级计算机所必需的工具。关于算法或计算方法的研究是高度数学性的，涉及数学科学的许多分支。计算机为数学提供了一条通往科学和工程技术各个领域的重要通道，也开辟了一个数学时代。报告指出，由于大容量存储的高速计算机的使用，已经导致了科学和技术方面的两大突出进展：一是大量用于设计工作的实验逐步被数学模型的研究所取代，如航天飞机设计、反应堆设计、人工心脏瓣膜设计等；二是能获取和存储空前大量的数据，并能提取隐秘的信息，如计算机层析 X 射线摄影、核磁共振等。

1991 年以美国总统的名义提出了“高性能计算与通信 (HPCC) 计划”。这是为了保持和提高美国在计算和网络等先进领域中的领导地位而制定的。该计划为期 5 年，由美国 8 个重要部门负责实施。投资的重点 (43%) 是发展先进的软件技术与并行算法，关键技术是可扩展的大规模并行计算。要求到 1996 年高性能计算能力提高 14 倍，达到万亿次每秒浮点运算速度 (TOPS)；计算机网络通信能力提高 100 倍，达到 10^9 位每秒 (Gbps)。该计划中列举的“挑战”项目有：磁记录技术、药物设计、催化、燃烧、海洋模拟、臭氧洞、空气污染、高速民用运输机、数字解剖、蛋白质结构设计和金星成像等。

1993 年初美国总统发布“发展信息高速公路” (NII) 的总统令。

1994 年 4 月美国总统发布“建立国家 (地球) 空间数据基础实施” (NSDI) 的总统令。

从上面这组数据中可以看出科学计算的重要性。设计科学计算的程序需要扎实的数学知识，有一些特定的设计方法，强调程序的算法、复杂度和精度。有关科学计算的理论和方法已经成为数学学科中的一个分支，即计算数学。常用于科学计算的程序设计工具有