



普通高等教育“十一五”国家级规划教材
现代计算机科学与技术教材系列

高级语言 程序设计

赵致琢 刘坤起 张继红 编著



国防工业出版社
National Defense Industry Press

普通高等教育“十一五”国家级规划教材
现代计算机科学与技术教材系列

高级语言程序设计

赵致琢 刘坤起 张继红 编著

国防工业出版社

·北京·

内 容 简 介

这是一本以 Pascal 为宿主语言,介绍高级程序设计语言及其程序设计基础,同时,用语言比较方法介绍 C 语言及其程序设计的教材。

本书基于计算机科学与技术一级学科人才培养科学理论,按照计算机科学与技术学科系列教材一体化设计的纲要,全面介绍了高级程序设计语言及其程序设计的基本内容,包括基本概念、基本结构、设施、成分和控制机制及程序设计的基本方法和技术。全书分两部分。第一部分以 Pascal 语言为宿主语言,介绍高级语言及其程序设计的基础内容;第二部分运用程序设计语言理论(原理),从语言比较学的角度,通过两种语言对比分析的方法介绍 C 语言及其程序设计。这样一种写作的方式,既总结了 Pascal 语言的内容,又展示了另一种不同风格的过程性语言,还传达了一种学习新型程序设计语言及其程序设计的方法,可加深读者对高级程序设计语言的认识,同时也为后续课程的学习提供了更为宽广的基础。

本书可作为计算机科学类专业和非计算机科学类专业“高级语言程序设计”课程的教材,也可供高等学校的教师、学生和广大工程技术人员参考。

图书在版编目(CIP)数据

高级语言程序设计/赵致琢,刘坤起,张继红编著. —北京:国防工业出版社,2010.7

(现代计算机科学与技术教材系列)

普通高等教育“十一五”国家级规划教材

ISBN 978-7-118-06950-1

I. ①高... II. ①赵...②刘...③. 张... III. ①PASCAL语言—程序设计—高等学校—教材②C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(2010)第 118981 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

涿中印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 26¼ 字数 605 千字

2010 年 7 月第 1 版第 1 次印刷 印数 1—4000 册 定价 45.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

总 序

民众多好饮酒,中外概莫能外。酒馆和酿酒坊伴随饮酒客而起,人类对酒的喜爱造就了酒文化和一个庞大的产业。好酒能卖好价钱,能使文人诗兴大发,催生佳作,还能解人间百难。于是,酿天下名酒自然成为不少人的毕生追求。

怎样才能酿出好酒呢?国人的看法不尽相同。崇信洋酒的人主张引进国外的生产工艺,学习洋人的生产和经营理念,而喜欢国酒的人则主张走自己的路,但不排除借鉴国外先进的科学技术和管理经验。这样的争论或许永远不会终结,但外国人重视科学酿酒,值得我们学习和借鉴。

计算机科学教育,如同酿酒工业的生产一样,科学办学迄今还只是部分学者的一种理想。与国内一样,国外的计算机科学教育并没有像他们的科学酿酒业一样,实现科学办学。也许科学办学要远比科学酿酒困难得多。譬如,怎么实现科学办学?甚至怎么推出一套科学的系列教材都是一篇大文章。

这套教材的创作始于教育部面向 21 世纪教育与教学改革 13-22 项目的研究。2000 年,在 13-22 项目研究工作即将完成之际,一些学者开始认识到面对计算机科学与技术的高速发展,我们亟需一套体现科学办学思想,反映内涵发展要求,服务教育与教学改革,参与构建学科人才培养科学体系的系列教材。强调系列教材是因为那时已经意识到计算机科学教育本质上是一项科学活动,但长期以来教师向学生传授科学技术知识的方式方法科学性不强。由于高等教育几百年来一直沿袭经验方式而非科学方式办学,大学教学的方式方法仍然还停留在古代作坊式的阶段,尽管教学使用的技术手段今天已经相当先进。在经验办学方式下,无论是研究型大学还是教学型大学,由于种种原因,教学活动的全过程存在着太多的漏洞和质量隐患。科学办学是对高等教育界传统办学方式的一个挑战,尽管在认识上,人们不难理解,科学办学是经验办学的最高形式,而经验办学应该成为科学办学的有益补充。

13-22 项目组积极探索,率先倡导科学办学理念,初步构建了一个体现科学办学思想,反映内涵发展要求的计算机科学与技术一级学科人才培养科学理论体系,为学科专业教育探索新天地,走向科学办学和发展学科系列教材提供了一个认知基础。

长期以来,学术界一直在探索计算机科学与技术专业教育的规律。ACM 和 IEEE/CS 的专家小组在走访了全美 400 多位著名计算机科学家的基础上,以学科方法论作为切入点开展教学改革理论研究,于 1989 年发表了具有开创性意义的成果,尽管他们并未意识到自己的工作是以学科方法论的研究作为切入点,探讨内涵发展的道路。1990 年前后,在迷宫中探索行走的专家小组经大师和精英群的指点实际上已经摸到了走出迷宫的大门,却没有打开并进入一个崭新的天地。这一点从他们在 2000 年网上公布的 CC2001 报告最先删除了 CC1991 报告中有关学科方法论的内容便不难看出(注:后经中国人的提醒

又补充写入)。

与此同时,中外教材建设也一直没有停止探索,国内外出版社先后出版了种类繁多的计算机类专业教材。这些教材中不乏精品和上乘之作,但难觅具有鲜明特色,真正一体化设计并且符合科学办学要求的系列教材。多数丛书和系列教材基本上还只是出版社对出自作者个人创作的教材通过冠名“丛书”或“系列”的方法结集出版以求强势效应。尽管如此,不少优秀作者和学者理所当然地进入了编审委员会的视线。西方发达国家在计算机科学学科的领先优势曾使许多人不自觉地目光转向海外,试图从世界名牌大学使用的教材中去寻找蓝本。遗憾的是与国内一样,经验办学并没有使西方大学在教材建设方面摆脱“各自为政,各行其是”的阴影。此时,我们如梦初醒,毕竟科学办学是前无古人的一项创举。随着学科的不断深入,在迈向深蓝知识海洋的今天,外国人未必比中国人在科学办学方面占有更多天时地利的优势。不经意中的发现使我们惊喜和激动,同时深感责任不轻且平添担忧:即使能够写出系列教材的一体化设计,我们是否真能确认这项改革的正确性?真能推出科学的系列教材?可是,除了实践和试验,我们别无捷径可循。

从上个世纪的50年代起,我国几代学者苦苦追赶了西方发达国家半个世纪,靠引进、学习、跟踪、改进、创新的高新技术发展思维定式曾使我们付出了高昂的学费和沉重的代价。固然,在高新技术领域,依靠“引进”和“泊来”,我们取得了长足的发展和进步,填补了不少国内的“空白”,但在水准上始终与发达国家保持着一段差距,一种在行业内部看来时长时短,难于逾越的差距。这种差距主要表现在对高精尖学科的发展我们缺乏思想、概念、理论、方法、技术、制度、规范和设计的原始创新和发展模式的全面创新,研究工作总是跟在别人后面亦步亦趋。我们缺乏在发展中另辟蹊径,走自己道路的机制和氛围,迷信洋人,盲目追随西方学术发展道路的习惯思维方式几乎导致国人失去了创新的机能,这是一个国家和民族发展高新技术学科和产业致命的硬伤。

高新技术领域竞争的成败,关键取决于人才与文化。现代科学技术的创新,已不单纯是一个学术问题,还是一个与文化、人文密切相关的问题。科学教育求真求是,技术教育求实求精,人文教育求灵求善,艺术教育求美求新。没有科学技术知识,人的认识和生活难免停留在原始社会,而没有人文精神和艺术的陶冶,科学技术的创新必然失去力量的源泉。可见,走自己的道路,发展中国的科技创新体系,在某种程度和意义上,成败的关键在于大学能否真正培养一大批高素质的人才。高等学校要实现培养大量高素质计算机科学与技术专业人才的目标,需要在前进中不断地进行系统的、科学的总结和深刻的反省,需要对遇到的问题进行科学的分析和判断,作出正确的决策。

工欲善其事,必先利其器。倘若教师不能在思想上摆脱陈旧的思维定式,用先进的理念武装头脑,勇于探索前人没有走过的发展道路,那么,即使采用了世界一流大学的全套教材,恐怕也难于培养出世界一流的人才。中西文化、人文传统之间的差异之大,中外教育思想之间的差异之大使得中国教育的现代化决不是一个引进和模仿就可以轻易解决的问题。教师的职业不是贩卖知识。授业、传道、启蒙、解惑技能的高低,不仅取决于教师知识的广博和深厚,更重要的在于远见、卓识、探索、创新、敬业、求真的本领和身先垂范。

身处21世纪,面对国家的期望,处在科学技术发展浪潮之巅的计算机科学的教师,任

重道远。我们就像茫茫林海中的探险者,环顾苍翠的群山,犹如身陷迷宫一样。计算机科学教育,敢问路在何方?其实,我们的出路或许只有一条,那就是系统总结前人的经验和教训,设法努力登上山峰,居高眺望,探寻走出林海的希望之路。

2001年初,《现代计算机科学与技术教材系列》编审委员会正式成立,一年后,计算机科学一级学科系列教材一体化设计研究报告的第一部分在《计算机科学》杂志第6期长篇发表。编审委员会为系列教材的出版制定了严格、详细的操作程序,并通过多种渠道,选择了第一批教材的作者。在体制创新方面本次教材出版设立了学术编审人,跟踪编审教材的创作内容,力求教材的尽善尽美。可以预期,《现代计算机科学与技术教材系列》将是一套基于计算机科学一级学科人才培养科学理论体系,体现内涵发展和科学办学要求,反映一体化设计的系列教材建设的一个尝试。然而,就像任何新生事物一样,她难免存在缺点和不足,我们诚恳地希望关心和使用本套系列教材的师生、读者,在使用中将批评或建议留下来,帮助我们改进教材建设工作,修正存在的错误。

一些学者对于一级学科人才培养科学理论体系的能行性表示怀疑。带着这个问题,在贵州大学和国内部分高等学校和一大批知名学者的支持下,从1999年夏天起在贵州大学连续举办了“计算机科学与技术高级研讨班”,向(博士)研究生和中青年教师陆续开设了研究生核心学位课程“高等计算机体系结构”、“并行算法设计基础”、“分布式算法设计基础”、“高等逻辑基础”和“形式语义学基础”、“可计算性与计算复杂性”、“形式语言与自动机理论”,后来又进一步开设了本科生重点课程“算法设计与分析”、“数理逻辑基础”、“信息安全”、“密码学原理及其应用”等一系列课程。五年来,高级研讨班受到全国广泛关注、响应和支持,先后吸引了几十所大学三百多人次的师生参加听讲和学习,目前已经发展到由教育部批准资助,15所大学联合主办的高级研讨班。实践证明,高级研讨班为中国高等学校计算机教学改革和教育质量的提高正在发挥其独特的作用,并得到国内外一大批学者的充分肯定和好评。高级研讨班正在成为按照一级学科办学和教学改革的要求,对计算机科学系教师进行高起点、高标准、正规化研究生学位课程和本科重点课程培训的一个师资培训模式,有可能对未来计算机科学教育产生深远的影响。试想,如果高等学校的教师和培养的研究生普遍具有高级研讨班所开设的3-4门课程的基础,不仅科学办学面临的主要困难迎刃而解,各大学科研学术队伍的素质也将得到显著提高。

一些学者对高起点研究生学位课程的必要性提出疑问:是否这些课程都要学习?我们认为,应该看到,在高等教育界从来就存在着两种不同的教育观,一种是专才教育观,一种是通才教育观。持这两种教育观的人尽管都主张基础知识的重要性,但在对学以致用原则的理解和解释方面存在差异。一般地说,专才教育观主张在一定的基础上,通过深入钻研某一方向的学问,逐步扩展和加深自己的知识,缺什么基础补什么知识,学以致用,逐步成长为一个学科的专家。通才教育观则不同,它不主张在具备一定的基础后,就匆忙沿着某一方向钻研学问,单线独进,而是主张在一级学科的范围内,通过尽可能系统地掌握从事本学科各个重要方向的研究所需要的基础知识,能够站在学科的各个制高点上,沿着学科的一个方向,以单线独进、多线并进或整体推进的观点,逐步扩展和加深一级学科的知识,融会贯通,学以致用,逐步成长为一个学科的专家。两种教育观都有其代表人物。迄今为止,高等教育中研究生教育主要以培养专才为主,专才教育观是主流。但

是,两种不同的教育观各有其特点。一般地说,当一个学科的发展处于早期时,专才教育比较容易跟上学科的发展步伐,比较容易出成果,也比较容易迅速地达到较深的学术层面。而当一个学科的发展比较成熟,发展速度比较平稳时,通才教育的优势就比较明显。因为,通才教育培养的人才可以在一级学科的范围比较容易地向任何一个方向转向。特别,在胜任高难度重大创新人才的培养方面,在出综合性的大成果方面,在创立一套科学理论和开辟一个研究方向方面,通才教育的多种优势往往是专才教育所不具备的。当然,两种教育观谁优谁劣迄今并无定论,根据两种教育观的特点和现实情况,选择哪一种教育观实际仅反映了师生的一种选择策略。不过,实践告诉我们,尽管通才教育观的操作实现比较困难,但作为师资补充的来源,通才教育培养的人才更容易适应大学教学与科研的双重要求,理应更多地受到研究型大学的青睐。在科学技术日益深化、高度分化又高度综合的今天,放眼未来,在高精尖学科中,通才教育观无疑有着更为宽广而美好的发展前景。

伴随着学科教学改革理论研究与实践探索的推进,社会热切地期待着一套与教学改革方案相配套的高质量系列教材问世。总结过去教材建设成功和失败的经验和教训,使我们清楚地认识到:教材建设必须建立在科学研究基础之上,按照科学的运作程序,动员在第一线从事科学研究,功底深厚,学有所长的优秀教师参与到教材的创作中来,才有可能推出高质量并符合学科发展要求的系列教材。我们的主张是:“让大学中的科学家来创作教材。”

冬去春来,年复一年。当我们终于从跟踪、学习、盲从西方大学教程的发展模式中走出时,感受到了一种从未有过的释然与激动,一种走自己的发展道路,独立自主的自豪与喜悦。这条道路虽然艰难,但前景光明。连续五年在贵州大学举办的全国计算机科学与技术高级研讨班的成功实践更进一步地坚定了我们内涵发展模式与科学办学之路的认识与追求。

今天,经过编审委员会、作者和出版社的共同努力,《现代计算机科学与技术教材系列》终于开始陆续出版发行。在新年即将到来的时候,我们怀着喜悦的心情,向祖国和人民,向计算机科学界献上一份完全由华夏学者按照他们对高等教育和计算机科学的理解与认识,倾力创作的新年贺礼,一份建立在科学研究基础之上的教育与教学改革最新成果。

新的世纪已经到来,愿《现代计算机科学与技术教材系列》的出版能够为新一代的莘莘学子攀登现代科学技术的高峰成就未来。

《现代计算机科学与技术教材系列》编审委员会

程序设计的教学改革要有整体的思考和设计

——《高级语言程序设计》前言

2001年夏天,在计算机科学与技术暑期高级研讨班(贵阳·花溪)上,我和坤起多次讨论“高级语言程序设计”课程的教学问题,感到有些问题确实需要认真深入探讨。当时的情况是,大多数学校普遍将“高级语言程序设计”课程开设逐步提前,有些已经进入了一年级的第一学期,而且,随着C++的兴起,Pascal语言渐渐淡出教学内容,取而代之的是C语言,后来更有部分学校直接采用C++进行教学,但普遍遇到不少困难和问题,教学效果并不好。那么,究竟应该以怎样的方式进行这门课程的教学,才是一个比较好的选择呢?

面对这一疑问,我曾有过认真的思考。我以为,“高级语言程序设计”课程是学生入学后第一次系统地学习与高级语言和程序设计有关的内容,可以说,对学生能否学好专业、顺利完成学业影响深远。现在,绝大多数学校把该课程安排在一年级,少数学校安排在二年级(上),可谓是各有各的道理,但多数人未必手执真理。要想真正弄清楚这个问题,建立科学的认识,还是应该从整体上来考虑问题。

众所周知,科学办学要有科学的理论指导,教学改革必须要有整体的思考和设计,将先进的教学理念贯穿在整个培养方案的始终,具体的改革才能获得准确的定位,并可从实际的成效判断改革是否正确。如果我们大家认同在教学改革中,局部服从于整体的原则,那么,不难看出,“高级语言程序设计”课程安排在一年级,对于培养高素质创新型人才,不是一个正确的选择。

首先,大学本科教育重在提高人的素质,为未来成长为专门人才打下一个厚实的基础,让学生获得自主更新知识,初步学会分析问题和解决问题的能力,以此训练学生科学的思维方法,致其在某个专业方面具备一定的专业技能。而且,一、二年级又是本科学习打基础的阶段,电子信息科学类专业教学中学生普遍需要面对数学、物理等基础课程繁重的学习任务,再加上英语、政治等公共课程,很少有时间来认真研读“高级语言程序设计”课程并较好地完成实验任务。一旦对创新人才的培养提升数理基础训练的要求和标准,那么,过早地安排专业基础课程,其对数理基础等课程教学的冲击等弊端必将显现。2010年全国硕士研究生入学考试科目“数学(一)”的考试结果和计算机科学与技术专业全国统考“专业科目”的考试成绩可以作为一个佐证。

其次,“高级语言程序设计”课程的教学,本身的重点不在于如何解决各种实际问题,而是在问题已经有了计算方法(甚至算法)之后,如何掌握使用高级语言实际编制解决问题的程序,为后续课程提供必要的高级语言与程序设计基础。高级语言种类繁多,究竟使用哪一种语言更合适,还是应该从语言是否适合初学者和该语言在本科教学中的地位来决定取舍。Pascal语言作为一种语法描述和程序结构比较规范,语言概念和成分有代表

性,语言规模大小比较合适的高级语言,将来可以与“数据结构”课程中的类 - Pascal、CPU 设计中的 VHDL 语言、程序设计语言原理和程序理论等很好地实现对接,用于作为“高级语言程序设计”课程中的宿主语言是恰当的。C 语言也是一种重要的程序设计语言,是当今计算机软硬件工程师应该掌握的一种基本的、实用的程序设计语言。但是,C 语言的内容比较庞杂,语言描述本身也不如 Pascal 规范,真正较好地掌握该语言,不经过长期、大量的实践是不现实的,而针对“高级语言程序设计”的初学者,其作为宿主语言并非最佳选择。

第三,在推免研究生复试中,常可以看到,许多在第一学年开设“高级语言程序设计”课程的学校,推免学生的数理基础训练和程序设计训练都受到了不同程度的影响,两者都没有学好。而且,在教学中,也常可以看到,不少学生到了高年级,还在钻研和“恶补”高级语言与程序设计的知识,直接干扰和影响了对后续专业(基础)课程的学习。

当然,应该看到,在重视数理基础课程训练和“高级语言程序设计”课程安排在二年级第 1 学期(如厦门大学、仰恩大学)的学校中,也存在一部分学生两者都没有学好的情况。但是,这与重视基础训练,按照教学要求进行学习,取得较好学习成效的同学不同,这些学生多是对专业不感兴趣,属于高考中调剂录取的学生;或是偏听偏信错误的观点,不重视数理基础课程的训练,没有养成良好的学习习惯,导致学习逐步陷入被动的学生;或是自以为是,受各种因素的影响,放纵自己,本身就没有认真对待学习的学生;还有其他因素的干扰。从我们近年来的教学改革实践看,几乎还找不到重视数理基础训练,学习态度认真,学习比较努力,但在程序设计方面薄弱的学生。在教学改革中,每个学生的情况不同,需要具体情况具体分析,区别对待。很明显,在“计算机科学与技术导论”课程中给予学生们正确的认识,坚定重视数理基础课程的训练,按部就班地依教学要求教学,是培养高素质创新人才的第一重要环节。尽管,这属于另外一个问题的范畴。

上述三个理由具有充分性,可以很好地解释为什么课程安排在一二年级,选择 C 语言进行教学并不是一个好的选择,而教学实践的结果也充分说明了这一点。

(1) 把“高级语言程序设计”课程安排在一二年级的学校,一般说来,加上其他课程,周学时已经超过了 30 学时(含实验),如果排课还不科学,必然导致学生少有时间来开展读书、思考、做习题、归纳、总结、预习等课外学习活动,消化基础课程的内容,每天似乎很忙,常处在疲于应付的状态,但忙得没有意义。加上 C 语言内容的特点和学生的智力深度还存在不足,结果是数理基础也难以学好,高级语言与程序设计也没有学好。而一旦数理课程没有夯实基础,今后要成长为高素质创新人才,眼下已经处于劣势。

(2) 即使是把课程安排在二年级第 1 学期的学校,如果采用 C 语言作为宿主语言进行教学,也并非是一个好的选择。因为 Pascal 自有其在整个学科中特殊的地位、作用和优势,不仅能够让学生比较容易地学习、掌握高级语言及其程序设计,支撑其更好地学习后续课程,而且,能够帮助他们以一种更严谨的态度和更高的观点来看待高级语言与程序设计的内容,不会对英语、数理基础、电子技术等基础课程的学习产生消极影响。

2001 年 12 月,在鼓浪屿召开的“全国计算机科学与技术教育与教学改革工作会议”上,一些代表也提出了类似的问题。对此,计算机科学家、计算语言学家陆汝占教授关于“向后教、向后学”的观点,为我们的判断提供了有力的支持和新的参考。以后,汕头大学的于津教授也曾告诉我,她多年来一直坚持以 Pascal 语言为宿主语言讲授“高级语言程

序设计”课程,效果比较好。没有想到,这位来自重视数理基础课程训练的吉林大学,曾师从编译专家的学者与我们的观点居然惊人地一致。

众所周知,在电子信息科学类专业的一年级,数理基础课程和外语的学习比“高级语言程序设计”更重要。良好的数理基础训练和外语教学训练,提高了一年级学生的智力水平、思维深度和外语能力,有助于学生更好地学习这门课程和后续课程。基于上述分析和认识,我们在教学改革理论体系中提出了当学生具备了一定的数理基础和计算机操作实验基础后,在二年级第1学期开设“高级语言程序设计”课程的安排,以Pascal语言作为宿主语言介绍高级语言与程序设计的内容,然后逐步过渡到C语言程序设计内容的构想(见绪论)。

带着这种认识和构想,我们经过近九年的素材收集和创作,完成了这本教材的初稿(内部讲义)。为了提高教材的质量,2009年秋天,结合仰恩大学计算机科学与技术专业的综合教育与教学改革项目,在2008级学生中进行了改革试验,由陈发强教授担任授课和实验指导。经过一个学期的教学实践,从理论教学与实验教学两个方面,初步证实了这样一种构想的可行性,同时也发现、修正和改进了原稿中存在的错误和不足,使得讲义得以在2010年公开出版。

通过这一次的教学试用和总结,我们提出以下几点意见供使用者参考:

(1) 课程安排课堂讲授72学时是必要而且恰当的,其中,50~52学时用于讲授第一部分,即Pascal语言及其程序设计的内容,用剩余的时间讲授第二部分,即C语言及其程序设计的内容是可行的。实验安排48学时左右,比较适中。其中,随开课程学期安排28学时,最好能安排32学时,重点是基础部分的实验;剩余16~20学时推迟到下个学期初的前几周执行,实验内容转向文件系统的使用和用C语言实现一个以文件为基础的信息管理系统。

(2) 本教材的起点比较高,实例的难度相对比较大,对教师和学生都具有挑战性,更突显了数理基础课程训练的极端重要性。当然,教师在授课时,可以选择一些相对简单的实例进行讲授,而把书中的例子留给学生自己研读。我们也打算把实例和难度降低,另外套写一本同类教材以适应不同类别的学校。

(3) 在从Pascal语言过渡到C语言的过程中,虽然我们用各种不同的表格列出了两种语言之间的异同点,但完全由一年级的学生通过语言比较,自己去学习和掌握C语言及其程序设计基础依然比较困难。我们建议教师在讲授中,可按照表格内容的顺序,概要性地介绍C语言及其程序设计的内容,同时,针对各知识点,结合前面章节给出的具有综合性特点的实例,逐一分析和讲解如何将一个Pascal程序改造成在功能上等价的C语言程序,帮助学生从语言比较学的角度快速掌握一种新的高级语言及其程序设计的基础内容。

(4) 我们最初选用的例子,多采用标准Pascal语言和标准C语言设计程序,但许多学校使用的编译环境一般为Turbo Pascal和Turbo C,两种版本语言的编译程序之间还存在一定的差别,师生在引用书中的实例时还需注意这一点。而实际上,从这次教学试验来看,从一开始就采用Turbo Pascal和Turbo C进行教学,不仅可以节省时间,减轻学生的负担,而且更符合大多数学校的实际情况。

由于本书入选了教育部“普通高等教育‘十一五’国家级规划教材”,受到出版时间的

限制,暂时已没有时间来对教材进行系统的修改,只能把这一工作留待将来改版时进行。在教材即将付梓之际,我们要特别感谢陈发强教授,他在 70 岁高龄的时候,在我们边创作、边修改、边交付内部印刷使用的情况下,不辞辛苦,依然担负起这一重任,并在课程教学改革实践过程中做了精心的授课准备,创造性地圆满完成了这项试验任务,教学质量受到同学们的高度评价。特别是他的认真备课和准备,等同于严格地审核了本书的初稿;他写下了大量教学笔记,运行了其中的许多程序,纠正了不少错误,提出了一系列修改意见。从他的身上,可以看到老一辈学者严谨的工作作风和敬业精神。

我们还要感谢厦门大学教务处领导和国防工业出版社编辑的耐心和支持,因为大家都知道,要创作一本好的教材是多么的不容易!此外,在教材创作的过程中,还得到陆汝占教授、康立山教授、余扬政教授、吕子玄副处长、闫敬文教授、杨承中教授、孙长圣教授等专家学者的帮助和鼓励,宋文教授抽空阅读了书稿,提出了许多修改意见,在此一并向他们表示衷心的感谢。在教材的试用过程中,仰恩大学计算机科学系 2008 级“教学改革试点班”的学生使用了教材的初稿——内部讲义,帮助陈发强教授发现了其中的一些错误,对此,也向他们表示由衷的感谢。

赵致琢

2010 年 2 月于武汉

目 录

绪论 对“高级语言程序设计”课程的认知与导学	1
0.1 对“高级语言程序设计”课程的认知	1
0.1.1 高级语言与程序设计的起源	1
0.1.2 程序设计语言的描述	4
0.1.3 计算模型、计算机与程序设计语言	6
0.1.4 数理逻辑和代数是程序设计语言与程序设计技术的基础	7
0.1.5 程序设计语言与程序设计属于科学的范畴	8
0.1.6 语言问题和程序设计问题均是计算机科学与技术学科中的核心问题 ..	8
0.1.7 对语言与程序设计技术的熟练掌握是计算机科学与技术工作者的 基本功	10
0.1.8 “高级语言程序设计”课程在学科专业教育中的地位和作用	11
0.1.9 本课程的宿主语言为什么是 Pascal 语言而不是其它语言?	11
0.2 内容的选取、组织与本课程的导学	12
0.2.1 内容的选取与界定	12
0.2.2 本教材内容组织的线索	14
0.2.3 与本课程相关的课程	14
0.2.4 对本课程的导学	14
第一章 引论	18
1.1 计算模型、高级语言与程序设计	18
1.2 程序设计语言 Pascal 简介	20
1.2.1 Pascal 语言的发展	20
1.2.2 Pascal 语言的特点	23
1.2.3 Pascal 语言的符号、约定	23
1.2.4 Pascal 语言源程序的结构	26
本章小结	30
习题	30
第二章 基本数据类型与基本运算	31
2.1 数据类型的概念	31

2.1.1	为什么程序设计语言中要引入“数据类型”这一概念?	31
2.1.2	数据类型的概念	32
2.1.3	数据类型的代数理论基础	32
2.1.4	Pascal 语言中数据类型的分类	32
2.2	基本数据类型	33
2.2.1	整数类型	34
2.2.2	实数类型	34
2.2.3	布尔类型	36
2.2.4	字符类型	36
2.3	常量与变量	37
2.3.1	常量	37
2.3.2	变量	39
2.4	标准函数	41
2.5	表达式	44
2.5.1	算术表达式	44
2.5.2	关系表达式	45
2.5.3	布尔表达式	45
2.6	数据类型的自动转换	46
2.7	数据类型的等同和相容	47
2.7.1	数据类型的等同性	47
2.7.2	数据类型的相容性	48
2.8*	计算机科学与技术学科中核心概念讨论之一——抽象概念	49
	本章小结	49
	习题	50
第三章	输入与输出	51
3.1	输入语句	51
3.1.1	read 语句	51
3.1.2	readln 语句	53
3.2	输出语句	55
3.2.1	write 语句	55
3.2.2	writeln 语句	56
3.3	程序设计举例	58
3.4*	关于输入输出的进一步讨论	60
	本章小结	60
	习题	61

第四章 语句与控制流程	62
4.1 语句概述	62
4.2 说明语句	63
4.2.1 标号说明语句	63
4.2.2 类型说明语句	64
4.2.3 几点说明	64
4.3 赋值语句	65
4.3.1 赋值语句的定义	65
4.3.2 有关赋值语句的讨论	66
4.3.3 程序设计举例	66
4.4 复合语句	69
4.5 条件语句	70
4.5.1 if 语句	70
4.5.2 case 语句	71
4.5.3 程序设计举例	72
4.6 循环语句	76
4.6.1 for 循环语句	76
4.6.2 while 循环语句	78
4.6.3 repeat 循环语句	79
4.6.4 多重循环	80
4.6.5 循环程序设计举例	81
4.7 转向语句	88
4.8* 关于语句的进一步讨论	91
4.9* 计算机科学与技术学科中核心概念讨论之二——绑定概念	92
本章小结	92
习题	93
第五章 程序的结构与类型	96
5.1 程序的基本结构	96
5.1.1 顺序结构	97
5.1.2 选择结构	97
5.1.3 循环结构	98
5.2* 程序的类型	99
5.3 程序设计技术	100
5.3.1 结构化程序设计技术	101
5.3.2 模块化程序设计技术	107

5.4*	关于程序结构的进一步讨论	108
5.5*	计算机科学与技术学科中核心概念讨论之三——分解概念	109
	本章小结	110
	习题	110
第六章	构造型数据类型	113
6.1	枚举类型	113
6.1.1	引言	113
6.1.2	枚举类型及其变量说明	113
6.1.3	枚举类型数据的运算	114
6.1.4	附注	114
6.1.5	程序设计举例	115
6.2	子界类型	118
6.2.1	引言	118
6.2.2	子界类型及其变量说明	119
6.2.3	子界类型的数据允许进行的运算	120
6.3	数组类型	120
6.3.1	数组的概念	120
6.3.2	数组类型及其变量说明	121
6.3.3	数组元素的访问方法及存储方式	122
6.3.4	数组类型允许进行的运算	123
6.3.5	数组的输入与输出	123
6.3.6	压缩数组	124
6.3.7	程序设计举例	126
6.4	集合类型	136
6.4.1	引言	136
6.4.2	集合类型及其变量说明	136
6.4.3	集合类型的数据允许进行的运算	138
6.4.4	集合类型的进一步说明	139
6.4.5	程序设计举例	139
6.5	记录类型	144
6.5.1	引言	144
6.5.2	记录类型及其变量说明	144
6.5.3	记录成分(域)的访问	146
6.5.4	记录类型的数据允许进行的运算	147
6.5.5	记录类型的数据的输入与输出	148
6.5.6	记录数组	148

6.5.7 变体记录	149
6.5.8 程序设计举例	152
6.6* 计算机科学与技术学科中核心概念讨论之四——聚集概念	157
本章小结	157
习题	158
第七章 过程、函数与分程序	161
7.1 过程	161
7.1.1 过程概述	161
7.1.2 过程说明	161
7.1.3 过程调用	163
7.2 函数	164
7.2.1 函数概述	164
7.2.2 函数说明	164
7.2.3 函数调用	166
7.2.4 函数与过程的比较	166
7.2.5 程序设计举例	167
7.3 标识符的作用域与生存期	170
7.3.1 全局量与局部量及其作用域与生存期规则	171
7.3.2 标识符的作用域的数理逻辑基础	173
7.3.3 作用域概念对程序设计语言及程序设计的意义	173
7.3.4 非局部变量及其副作用	174
7.4 信息传递	176
7.4.1 引言	176
7.4.2 信息传递的方法	176
7.5 过程与函数的嵌套	182
7.5.1 过程与函数的嵌套	182
7.5.2 过程(函数)的调用原则	183
7.6 递归	185
7.6.1 递归的概念	185
7.6.2 递归过程(函数)的执行	187
7.6.3 递归程序的特征	188
7.6.4 递归程序设计技术举例	188
7.6.5 递归与递推的关系	195
7.7 分程序	195
7.8* 计算机科学与技术学科中核心概念讨论之五——封装概念	196
7.9* 计算机科学与技术学科中核心概念讨论之六——递归概念	197

本章小结	197
习题	197
第八章 指针与动态数据类型	204
8.1 指针	204
8.1.1 指针的意义	204
8.1.2 指针数据类型	206
8.2 动态数据类型	209
8.2.1 静态数据类型与动态数据类型	209
8.2.2 动态变量的生成与废料的回收	209
8.2.3 动态变量的使用	211
8.2.4 指针与动态变量有关知识小结	212
8.3 指针的应用	213
本章小结	219
习题	220
第九章 文件	223
9.1 文件概述	223
9.1.1 文件的概念	223
9.1.2 文件的分类	224
9.1.3 标准 Pascal 文件的数学基础及特点	225
9.2 类型文件	226
9.2.1 文件类型的说明及其变量说明	226
9.2.2 文件缓冲区变量	227
9.2.3 对类型文件实施的基本操作	228
9.2.4 类型文件的应用	232
9.3 文本文件	240
9.3.1 文本文件及其操作	240
9.3.2 标准文件	246
9.3.3 文本文件的应用	246
9.3.4 文本文件与类型文件的比较	251
本章小结	252
习题	252
第十章 高级程序设计语言 C	254
10.1 C 语言与 Pascal 语言的符号、约定的比较	255
10.1.1 ANSI C 语言与标准 Pascal 语言的字符集合	255