

“高级语言程序设计实验” 课程的教学问题探讨

赵占芳¹, 刘坤起¹, 赵致琢²

(1. 石家庄经济学院信息工程学院, 河北 石家庄 050031;
2. 厦门大学信息科学与技术学院计算机科学系, 福建 厦门 361005)

【摘要】 针对“高级语言程序设计实验”教学过程中普遍存在的问题, 本文提出了以程序调试技术为主线的实验课程教学思想和方法, 并加以实施; 要求学生严格遵循代码的编程规范, 帮助学生建立高质量程序设计的理念; 通过综合性课程设计全面提高学生的综合能力。此外, 还介绍了课程所用教材的建设情况。

【关键词】 高级程序设计语言; 程序设计; 实验教学; 编程规范; 程序调试

【中图分类号】 G642 **【文献标识码】** A **【文章编号】** 2095-5065 (2014) 12-0034-06

0 引言

“高级语言程序设计”课程是计算机科学与技术类专业的一门非常重要的专业基础课程。该

课程实践性强, 需要经过大量的上机实践才能真正掌握。在多年的教学实践中我们发现, 相当一部分学生在该课程实验部分的教学, 并没有真正掌握该门课程的基本实验技能, 没有为后续课程的学习打下坚实的基础, 致使一部分学生难以顺利学习后续课程, 有的甚至失去了对专业的兴趣, 放弃了本专业的学习, 教训十分惨痛!

那么, “高级语言程序设计实验”课程的教学究竟存在什么问题? 如何解决这些问题? 这都是大家普遍关心的问题。本文首先从分析课程在教学中存在的问题出发, 提出了解决这些问题的一系列方法和措施。教学实践表明, 课程教学中存在的问题可以得到很好的解决, 从而使实验课程的教学质量得到根本保证。

1 “高级语言程序设计实验”课程普遍存在的问题

“高级语言程序设计实验”课程普遍存在一

收稿日期: 2014-10-8

作者简介: 赵占芳 (1978—), 女, 河北正定县人, 副教授, 主要从事演化计算、程序设计语言、计算机科学教育等方向的教学与研究工作;

刘坤起 (1966—), 男, 河北无极县人, 工学博士, 教授, 主要从事演化计算、程序设计语言、计算机科学教育等方向的教学与研究工作;

赵致琢 (1957—), 男, 上海人, 博士, 教授, 科学办学理念及其实践的首倡者, 独立创建了经过实践检验的计算机科学一级学科人才培养理论体系, 系统性的教学成果曾获省级、国家级教学成果奖, 主要从事计算机模型与分布式基础算法、逻辑程序设计基础、计算机科学教育等方向的教学与研究工作。

基金项目: 本文的工作得到教育部计算机科学与技术专业综合改革试点 (石家庄经济学院)、河北省高等学校二级学院综合改革试点 (石家庄经济学院信息工程学院)、石家庄经济学院教改项目2012J01、福建省计算机科学与技术人才培养创新实验区 (厦门大学) 等项目的经费支持。

些问题,如教师对实验课程的教学目标定位有偏差、学生实验技能的掌握情况不尽如人意,作业和课程设计的实施不理想,等等。这些问题如果得不到解决,将严重影响实验课程的教学质量。

1.1 实验课程教学目的定位不准确

“高级语言程序设计”的理论部分和实验部分的教学目的存在明确分工。理论部分的教学重点是介绍宿主语言(高级程序设计语言)的基本概念、基本结构、设施、成分和机制等基本内容和基于宿主语言的基本程序设计方法和技术;而实验部分的教学重点是借助宿主语言的集成开发环境,介绍程序调试技术,使学生尽快掌握程序调试——计算正确性的验证以及其他性能的验证方法和技术,以加深对理论知识的理解。

一些教师对实验部分教学目标的定位不准确,常以教会学生如何解题为目的,这与理论部分的教学混为一谈,没有准确理解实验部分在“高级语言程序设计”课程中的地位和作用。实验部分的任课教师需要明确认识实验部分和理论部分的授课重点分别是什么。如何针对一个具体问题进行分析,并为其设计算法和程序,是基本程序设计技术理论部分教学的重点之一,而不是实验部分教学的重点。实验部分的教学目的,首先是验证程序的正确性,其次是验证程序的健壮性等其他性能。实验部分的教学重点是教授在程序运行出现问题时,如何借助集成开发环境,利用程序调试方法和技术去分析问题,找到问题的根源并解决问题。

1.2 学生无法利用理性方法解决实验中遇到的问题

由于学生没有系统掌握程序调试方法和技术,在“高级语言程序设计实验”课程中,经常会出现如下两种情况。

(1)部分学生在调试程序的过程中,对调试技术一无所知,一旦程序出现问题,就会束手无策,下意识地寻求指导教师的帮助,缺乏自己主动探求问题的能力。而过度依赖指导教师的帮助,会导致一旦离开教师的指导,实验就做不下去,甚至寸步难行。

(2)部分学生对程序调试技术掌握得不透彻,同时缺乏理性调试方法的指导,盲目相信调试器,调试程序的过程就像无头苍蝇一样到处碰壁,工作效率低。

以上两种情况说明了学生调试程序的实验过程是盲目的,不是在理性方法指导下的高效的实验。即使程序调试正确了,很多学生依然不能把调试过程中用到的方法和技术进行系统、理性的总结和分析,用于指导后续实验,这造成实验课效率低下。不仅如此,由于盲目的调试,造成指导教师的工作量极大,甚至疲于应付。

1.3 课程必要的作业量得不到保证

“高级语言程序设计”课程高质量的教学特点之一,就是必须通过高强度的实验训练,才能真正理解宿主语言,进而理解高级语言及其程序设计的精髓。经过多年的教学实践发现,该课程基础实验的练习量至少应有2500~3000行的代码总量,即理论部分和实验部分的作业总量应不少于2500行的代码。如果该课程必要的作业量得不到保证,会使得教学效果大打折扣,即使教师反复讲解,很多学生依然对程序设计不得要领,教学效果事倍功半。

由于没有系统掌握程序调试方法和技术,使得学生在完成课后作业的过程中困难重重。也因为缺乏理性方法和经验的指导,作业拖延上交的情况严重,同时作业完成的质量也不尽如人意。注意到很多学生在实验的实施过程中是盲目的、非理性的,他们做实验的效率低下,如果不能及时加以解决,会造成很多学生完成课后作业的积极性较差,致使该课程必要的作业量得不到保证,最终影响该课程的教学质量。

1.4 课程设计的实施困难重重

多年的实践表明,对于计算机科学与技术类专业的技术类课程的学习,要想熟练掌握之,必须做一个有一定规模的综合性的大题目,即课程设计。否则,其中的技术难以真正熟练掌握,“高级语言程序设计”课程也不例外。

但是,“高级语言程序设计”课程的课程设计,其实施过程却举步维艰。许多学校由于培

养方案不合理，实验部分学时数太少，当课程结束时，很多学生只会编写几十行的小程序，一旦要求其独立写一个规模稍大一点的程序，如500~1500行，他们多数不能胜任，无从下手。这是因为该课程必要的作业量没有得到保证，学生对于基本的程序设计方法和技术、程序调试方法和技术没有完全掌握，导致他们对综合性课程设计的实施如同构建空中楼阁，困难重重，甚至不得不夭折。

如果综合性课程设计不能顺利地推进和有效实施，学生就不能利用所学的计算机知识去解决现实中的问题，其分析问题、解决问题的综合能力也得不到锻炼和提高，那么，此课程的教学目的也就没有真正达到。或者说，该课程的实施不是完全成功的，并且对后续课程的顺利开展也会造成一定的困难。

2 以程序调试技术为主线开展基本实验教学

授人以鱼不如授人以渔，由于高级语言程序调试方法和技术是这门课程最基本的实验技术，而对这一技术的熟练掌握是科班学生必须练就的基本功之一，因此，有必要将其作为这门实验课程教学的重点。

2.1 学习程序调试技术的意义

程序调试技术的用途相当广泛。首先，它最主要的用途是定位并排除程序中的错误（bug）；其次，调试技术在安全领域也有重要的应用，利用调试器可以用来查杀杀毒软件无法对付的病毒，内核调试器是检测系统入侵的重要工具；第三，程序调试技术一项非常重要的应用，就是让调试器成为强大的“学习机”，可以利用其独特的观察力和强大的控制力，来观察和学习计算机软硬件，探索和认识计算机世界。

作为高级语言程序设计的初学者，掌握基本的程序调试方法和技术，对课程的学习有着非常重要的意义。掌握程序调试技术，可以使程序员对程序的纠错从经验走向理性，从混沌走向清

晰，极大地提高学习的效率；调试技术是学习其他计算机软硬件技术的极好的工具，通过调试器的强大观察能力和程序控制能力，可以快速了解程序的模块、架构和工作原理，可以更深入地理解计算机语言的语法机制和各种设施。而且，核心的调试方法和技术大多源于CPU的设计思想、结构和操作系统的直接支持，具有非常好的健壮性和稳定性。相对于其他的软件技术，程序调试技术不会在短时间内被淘汰，是一门一旦掌握便终身受用的技术。

2.2 将程序调试技术分布到不同实验单元，并通过典型实例加以介绍

程序调试是一门实践性强、涉及面广的综合技能，具体的调试技术与CPU架构、操作系统和编译器密切相关。要想全面系统地掌握程序调试方法和技术，需要一个长期艰苦的学习和实践过程。对于程序设计的初学者，调试技术的学习也是一个由浅入深，逐步深入的过程。因此，在“高级语言程序设计实验”课程的教学过程中，按照知识点的展开顺序，将程序调试技术循序渐进地分散到每个实验单元中，通过典型实例分别介绍，让学生由易到难，逐步加深并掌握程序调试的基本方法和技术，提升专业技能。

在“高级语言程序设计实验”课程的授课过程中，要善于利用典型例题，介绍程序调试技术的排错和学习这两大功能。对于高级语言程序设计中的几个教学重点和难点：多重循环、参数传递和函数调用、递归程序、指针及链表等，首先借助于集成调试器强大的程序运行控制能力，观察程序的执行流程、子程序的调用、模块与框架等；在掌控程序执行的过程中，要充分利用集成调试器的强大观察能力，观察变量的值、变量的地址、递归调用栈及内存空间的使用情况等。在此过程中，更多地是教给学生如何利用调试器的控制能力和观察能力来学习程序设计语言，从而更深刻地理解程序设计语言的内涵。

2.3 加强程序基本调试技术的训练，采用对比方法学习其他调试器的使用

程序调试技术是一种基于CPU和操作系统支

持的技术,因此,发展相对稳定。对于不同的调试工具,其调试程序的功能在工作方式和工作原理上基本一致,但在特定平台和开发环境中其实现细节有可能存在略微的差别。因此,初学者首先应在某个特定的开发环境中加强基本调试方法和技术的训练,进而采用对比方法学习其他环境中调试器的使用。

石家庄经济学院教学改革试点班“高级语言程序设计”课程的教学,宿主语言采用Pascal语言,要求学生自学C语言。在实验部分的教学,采取了首先加强基本调试方法和技术的学习和训练,在此基础上,引导学生对比不同开发环境下,不同的程序调试工具的差异。这使得学生后来在运用语言比较学方法,以自主学习为主学习C语言程序设计部分时,能够自发、主动地拓展学习该语言开发环境下的程序调试方法和技术,从而借助程序调试技术,提高学习的效率。实践证明,这种方式不仅可行,而且有效。

3 综合性课程设计的实施

课程设计是培养学生综合运用所学知识,发现、提出、分析和解决实际问题,锻炼实践能力的重要的教学环节。通过课程设计的实施,充分锻炼学生综合运用所学知识分析问题、解决问题的能力,使学生程序设计的综合素养得到较大的提升。

3.1 课程设计的实现考虑开放性原则,鼓励创新

课程设计的实施依照开放式作业的原则,仅给出最核心的软件功能要求,鼓励学生自主对软件功能进行拓展。课程设计的实施采用开放式的原则,可以让学生对课程设计的实现有更多的选择,也照顾到了不同层次学生的不同需求。基础稍弱的学生完成基本的软件功能实现即可,基础较好的学生可以进行更深层次的探索和研究。

课程设计的实施采用开放式原则,充分锻炼了学生的综合能力。为了对软件功能进行拓展,首先,学生会查阅和搜集与课程设计内容相关的

知识背景、数据资料;在综合分析相关信息资料的基础上,进行软件功能的设计;在拓展功能的实现上,还需要在实现方法和实现技术上进行学习和探索。这个过程使学生搜集信息、提出问题、分析问题、解决问题的综合实践能力得到很好的锻炼,对知识的理解能力也有了不同程度的提高。

3.2 课程设计随着教学内容的深入而不断进行版本升级

课程设计随着教学内容的深入而不断进行版本升级,课程设计的内容和要求也随着版本的升级不断增加难度。这样的实施过程,使得课程设计的难度由易到难逐步递增,各版本之间难度跨度不大,避免了学生突然面对难度较大的设计时,无从下手的局面。

“高级语言程序设计”课程设计要求依次开发基于数组、基于链表与文件的学生成绩管理系统,开发语言分别采用Pascal和C语言。在使用Pascal语言开发低版本的学生成绩管理系统时,要带领学生从需求分析做起,按照概要设计、详细设计、编码实现等流程展开,并对每一个阶段进行详细的介绍和说明。后续基于Pascal语言高版本的开发和基于C语言各版本的开发,将由学生按照此流程进行设计和实现,不再作较为详细的介绍和说明。

3.3 严格的程序验收机制

传统的课程设计,验收不严格,重作业布置,轻作业评价,“走过场”现象比较常见。只有实行严格的程序验收机制,才能保证课程设计的质量。对每一个学生的每个版本的课程设计作业都要进行一对一、面对面的程序验收,程序验收要详细检查代码的正确性、规范性和必要的健壮性。在程序验收过程中,要对程序核心功能的代码实现或算法设计进行提问,要求学生当场给出解答。如果学生对自己编写的程序不能给出合理和正确的解释,则程序验收不合格。对于程序设计中不完善的地方或错误的地方,要给出建议和评注,要求学生改进或修正后,重新验收。最

终根据程序代码书写的规范性、功能实现的正确性、健壮性、人机交互性、界面设计的美观性及功能拓展情况给出课程设计的总成绩。

在全部的课程设计作业验收完毕后，要进行必要的总结和评价，这一工作的意义相当重要。要挑选出设计新颖、方法独特或功能实现较为完善的课程设计作业，通过班级讨论的形式，对其进行点评，鼓励通过同学之间的讨论对某个作业给出评价和建议。这个过程，可以激发学生的积极性和创造性，通过讨论和探讨相互借鉴，取长补短，对自己开发的程序进行必要的改进和完善。

3.4 课程设计报告的规范化

课程设计报告要按照需求分析、概要设计、详细设计、编码实现、总结与体会、参考文献等内容展开。同时还需要提交源代码和相关文档资料，如程序运行所需的数据文件、简单的程序使用说明文档等。

课程设计报告的需求分析部分，要明确数据需求、功能需求、性能需求等；概要设计要给出软件功能模块图、各模块的调用关系图和模块的接口设计；详细设计要给出核心功能的算法描述和重要的数据结构描述；编码实现部分要给出详细的测试用例以及程序测试运行的截图；总结与体会应该说说明程序开发过程中的难点分析，程序调试技术的使用情况，开发过程的心得体会等；最后按照国家相关规范给出参考文献目录。

4 建立编写高质量程序的理念

软件维护的成本远远高于软件开发的成本，因此编写高质量的程序是至关重要的。对于程序设计的初学者来说，在代码编写过程中，一定要建立编写高质量程序代码的理念，遵循程序设计语言的基本编程规范，培养良好的程序设计风格和习惯。

4.1 初学者要严格遵循编程规范

程序设计的主要功能除了能够解决具体的问

题之外，还有一个重要功能，就是共享和交流。一个只有自己才能读懂的程序不是一个好的程序，至少不是一个成熟的高质量的程序。因此，初学者在编写第一个程序时，就要建立编写高质量程序的理念，严格遵循代码编程规范。遵循代码编程规范对于代码的交流与共享，以及程序本身的调试和错误的排查都是极为重要的。

代码的编程规范包括命名规范、注释规范、代码排版格式规范、布局规范、模块化规范等。初学者在起步阶段要掌握这些规范，养成良好的编程习惯，将促进所编写程序的可读性、可维护性及健壮性等性能。

4.2 必须提供必要规范化的文档

初学者需要了解软件文档的重要性。软件文档是一种重要的软件工程技术资料，如系统分析文档、设计文档、版本说明文档等。软件文档的规范编制，在软件开发工作中占有突出的地位和相当大的工作量，对于软件产品的设计开发、发行使用、变更维护、二次开发等都有着重要的意义。

课程要求学生能够对复杂的程序尝试提供简单的文档说明资料，例如提供用户使用说明文档、程序运行的数据文档等。但是如何编写完善的软件文档资料不是实验课的重点，重点是让学生了解软件文档的重要性及意义，同时认识软件文档规范的重要性。尝试训练学生编写简单的文档，从而为学生后续课程的学习和工作需要打好基础。

4.3 了解程序的版本管理规范化

课程设计的实施，是随着难度的递增而不断进行版本升级的。此处课程设计的版本升级与现实中应用程序的版本升级是有区别的。课程设计中每个新版本功能增强了，但数据结构发生了改变，代码均是完全重写的，每个版本都是完全独立的程序软件。

现实中应用程序的版本升级（version go up）是指对操作系统或软件以前版本的漏洞进行完善，或者对软件添加新的应用功能的更新，或者把原先软件中存在的bug等错误信息进行修改等，

使得软件功能更为完善，一般是一种增量式的升级更新。

在实验课的教学中，应该向学生介绍软件版本升级的概念、常用的软件版本管理软件，以及软件版本相关文档的管理等，理解课程设计中版本升级的概念与现实中应用程序版本升级的区别。

5 实验教材建设

自2010年石家庄经济学院启动计算机科学与技术专业综合教学改革以来，为了适应教学改革的需要，提升“高级语言程序设计”课程的教学质量，先后编写了两本教材。理论课教材《高级语言程序设计》，由刘坤起、赵致琢编写，第2版将由电子工业出版社出版发行；实验课程教材《高级语言程序设计实验教程》，由赵占芳、刘坤起编写，已经由电子工业出版社出版发行。

《高级语言程序设计实验教程》秉承计算机科学与技术专业综合教学改革的实验教材建设一体化理念，是一本与理论课程相配套的实验教材。该教材具有如下几个特点。

(1) 将基本的调试方法和技巧由易到难分散到各个实验单元中，以典型实例的形式展示调试工具的使用及调试方法的应用；

(2) 每个实验单元中选择典型例题展示“分析问题、算法设计、源代码编写”的程序设计学科流程，帮助学生重温和掌握学科方法论的思想；

(3) 将程序设计过程中常见的错误进行总结归纳，并分析错误原因，避免程序设计的初学者盲目调试程序。

《高级语言程序设计实验教程》的优势是以程序调试技术的介绍为主线贯穿实验教学全过程，把学生掌握程序调试方法和技术作为教学重点，使得他们在学习程序设计的过程中掌握程序调试技能，从而使用理性的方法和策略对待程序调试问题。经过几轮的试验发现，在程序设计语言的学习过程中，学生能够自主地学习和使用程

序调试方法和技术进行语言的学习，达到了预期的目标。

6 结语

本文针对“高级语言程序设计实验”课程普遍存在的问题，根据计算机专业科学办学理念下综合教学改革项目的要求，在实验课程教学中引入新的教学方式方法，教学改革围绕课程设计及其实施、程序设计规范、教材创作等方面进行建设，收到了良好的教学效果。试点班学生经过一段时间的训练，都能够熟练地运用工具和环境，开展课程设计，掌握程序调试方法和技术，独立开发有一定规模的综合性程序系统，课程教学质量上了新台阶。但是，教学改革是一个复杂的系统工程，任重而道远，需要做的工作还有很多，教育工作者仍需继续努力，进一步提高教学质量。

【参考文献】

- [1] 刘坤起, 赵致琢. 高级语言程序设计[M]. 北京: 电子工业出版社, 2014.
- [2] 赵占芳, 刘坤起. 高级语言程序设计实验教程[M]. 北京: 电子工业出版社, 2014.
- [3] 赵占芳, 刘坤起, 赵致琢. “高级语言程序设计”课程的教学问题探讨[J]. 工业和信息化教育, 2013(12): 68-73.
- [4] 赵致琢. 普通高等教育科学办学的理论探索与改革实践——以“计算机科学与技术”学科为例[J]. 工业和信息化教育, 2013(12): 1-11.
- [5] 赵致琢. 计算机专业教学改革与科学办学的探索与实践(上)——问题、理念、方案、特色与效果[J]. 工业和信息化教育, 2013(12): 12-20.
- [6] 赵致琢. 计算机专业教学改革与科学办学的探索与实践(下)——经验、教训与再认识[J]. 工业和信息化教育, 2013(12): 21-43.
- [7] 刘坤起, 李慧琪, 孙长圣, 赵致琢. “编译程序设计原理”课程教学之思考[J]. 工业和信息化教育, 2013(12): 64-67.