

“高级语言程序设计”课程的教学问题探讨

赵占芳¹, 刘坤起¹, 赵致琢^{2,3}

- (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;
3. 仰恩大学计算机与信息学院计算机科学系, 福建 泉州 362014)

【摘要】 本文针对“高级语言程序设计”课程的教学实际,分析了计算机专业“高级语言程序设计”课程的设置与语言的选择问题,给出了课程教学应该坚持的原则,并对教学中的几个关键环节和知识点的处理与教材建设进行了讨论,以期不断提高课程质量,达到教学目标。

【关键词】 程序设计语言; 程序设计; 课程内容; 教学方法; 实践环节

【中图分类号】 G642.3 **【文献标识码】** A **【文章编号】** 2095-5065 (2013) 12-0068-06

0 引言

学习并熟练掌握高级语言和程序设计技术是计算机专业学生的基本功之一。为了练就这一基本功,计算机专业的培养方案中均设置了“高级语言程序设计”类课程。不同的学校对这门课程的设置及大纲的制定,从高级语言的选择到程序设计方法、技术的深度和训练要求各不相同。为此,课程配置的讲授学时数和实验学时数也不尽

相同。然而,这些课程的定位及要求是否合理?如何更有效地教学?哪一个方案是科学、合理的?这些问题一直是从事“高级语言程序设计”课程教学的教师长期探索,而且一直没有很好解决的问题。由于“高级语言程序设计”课程的教学质量对后续专业(基础)课程的学习具有基础性的地位和作用,其教学质量决定了学生能否顺利入门计算机专业领域,并在后续课程中更好地完成学业。本文系统地回答了这些问题,为今后

收稿日期: 2013-12-5

作者简介: 赵占芳(1978—),女,河南焦作人,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,软件工学博士,教授,主要从事计算机模型与分布式算法、逻辑程序设计基础、计算机科学教育等方面的教学与研究工作。

基金项目: 本文的工作得到教育部计算机专业综合教学改革(石家庄经济学院)、河北省计算机专业综合教学改革(石家庄经济学院)、河北省高校综合改革试点学院(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和2008J04、福建省计算机专业人才培养创新实验区(厦门大学)、福建省计算机科学与技术特色专业(厦门大学,仰恩大学)、福建省本科重点学科专业(仰恩大学,计算机科学与技术)等项目的经费支持。

的发展和教学提供了一种参考。

1 计算机专业“高级语言程序设计”课程的设置与语言的选择问题

设立“高级语言程序设计”课程，离不开遴选高级程序设计语言，因为语言是学习程序设计的载体，同时又对学习后续课程和构建学科核心知识结构产生广泛而深远的影响。

1.1 宿主语言到底应该选哪一种

一般情况下，“高级语言程序设计”是学生进入计算机专业后学习的第一门专业基础课，其宿主语言的选择不是任意的，需要遵循一系列的原则。根据学科人才培养科学理论体系，有如下原则。

(1) 宿主语言应该是过程性语言。过程性语言与算法比较接近，其思维方式又贴近大多数刚完成12年基础教育的大学一、二年级学生，易于为初学者接受。课程所介绍的程序设计是建立在算法基础之上的实现技术，由此一脉相承。尽管开设这门课程时，“算法设计与分析”课程尚未开设，但这并不影响教学中通过将编程例题、习题限定在趣味程序设计和简单的计算问题的范围，在教学中引入算法的概念，将问题分析、计算方法、算法与程序设计分层次表达、传递给给学生，让学生顺利掌握高级语言及其程序设计的内容。而更为高级的程序设计方法和技术，应该与本课程实现合理的分工，将来由“面向对象程序设计”、“程序设计方法学”等课程去承担。

(2) 宿主语言应该具有过程性语言所共有的基本结构和成分，如类型、程序基本结构、概念、设施和控制机制等，承载着更多学科发展进程和程序设计发展历程中保留下来的思想、概念、原理、方法和技术信息，其成分结构(包含语法和语义描述)应该具有普遍性，设施比较完备，

其定义方式具有代表性。

(3) 宿主语言应该是一种比较规范的语言，能体现良好的程序设计风格，语言结构简明，规模大小适中，便于初学者学习和掌握。Pascal语言比较符合这样的条件，也便于教学，但仅能支撑部分后续重要课程，且与当前企业对毕业生的专业要求也不尽一致。C语言可以适应市场大多数用人单位的要求，但对部分后续重要课程的支撑力度不够。更重要的是，C语言自身在设计和发展过程中，语言成分众多，规模比较大，内容和设施比较琐碎，规范性不足，初学者不易掌握，还会带来副作用。

(4) 宿主语言应尽可能遴选在整个学科发展中具有承上启下作用，在各行各业有广泛应用的流行语言，能为学生学习后续课程，如“高级程序设计语言理论”、“硬件描述语言与数字系统设计”、“编译原理”、“现代软件开发方法和技术”等打下良好的基础，也能为学生未来顺利就业提供支撑。目前，已知VHDL类语言和软件开发工具Delphi等与Pascal比较一致，但Pascal的流行性趋势已经减弱，取而代之的是以C语言为核心的各种版本的C语言及其派生语言，如ANSI C、C#。

(5) 宿主语言有较好的编译系统和调试工具的支持。基于上述分析和多年的教学经验，笔者认为，作为计算机专业学生学习的的第一门程序设计课程，宿主语言应该首选Pascal而不是其他语言。虽然，在今天的大学里，绝大多数高校的高级语言程序设计：课程已采用C语言、C++语言或Java语言，但这并非是一种恰当、正确的选择。因为，C、C++、Java这类语言因自身不够规范，或含有复杂结构等多种因素，并不适合初学者学习。针对计算机专业低年级的学生，初学高级语言程序设计时选用哪一种语言，应该将这样一种选择放在整个学科人才培养模式及其方案下，依据学科发展规律、人的认识规律和教学规律来综合判断，否则，极可能出现“主次不分”、“喧

宾夺主”等情况，不仅影响其他课程的教学，也影响教学计划和培养方案的平稳推进，而且，还不利于学生将来对新知识和更高级课程的学习。作为计算机专业的科班学生，在学习了精心规划和设计的“高级语言程序设计”课程之后，应该有能力依靠本专业系统的核心知识结构和专业技术能力，在需要时迅速地自主学习掌握一种新的高级语言及其程序设计，这一观点已经在实践中被反复检验，并被证明是正确的。

1.2 计算机专业学习语言越多越好吗

目前，国内计算机专业普遍开设了多门程序设计语言课程。而随着新语言的推出，还出现了可能增加的趋势。对此，国内一些专家开始思考下列问题。

计算机专业学习编程语言越多越好吗？计算机专业程序设计语言课程到底如何设置？

其实对这些问题，可根据计算机学科特点、语言学和教学规律得出正确答案。

在计算机专业四年学制不变的情况下，过多地开设程序设计语言类课程，势必压缩其他课程的学时，甚至压缩核心课程学时，这样下去“后患无穷”。由于整个学科的核心知识体系已相当庞大，其培养方案中公认的核心课程门数已远超其他大多数学科一般仅有6~8门核心课程的情况。因此，要保障人才培养质量，既不能用降低核心课程学时数和教学深度，转而增加通识类课程的方法，也不宜用降低核心课程学时数和教学深度，转而增加各种计算机选修课程，包括程序设计语言类课程的方法。这两种方法都将引起本科教学中广而不深的情况，许多课程的教学将无法达到既定的、应有的目标，许多重点内容因学时数不足、基础薄弱、训练不到位而难以实现专业能力的有效提升，更谈不上融会贯通，最终导致毕业生在就业市场的激烈竞争中被淘汰，或难以适应高层次、高端岗位的要求。

要解决这个问题，必须贯彻内涵发展优先的教学理念，采取相应的措施，即在“高级语言程

序设计”教学中，深入研究，厘清学科方法论意义下本质的内容，准确界定这门课程的核心内容和教学目标，贯彻内涵发展优先的教学理念，兼顾课程当前的教学目标、中长期对后续课程学习和构建更高层次核心知识体系的需要，兼顾相关课程的教学需要，精选课程教学内容，明确教学要求，引入先进的教学方法，以学科方法论统领整个课程的知识体系，真正按照内涵式的教学方式方法，扎实推进课程各知识点的教学与训练，循序渐进，努力促进学生在课程内部、课程与课程之间融会贯通，将各个重要的教学环节和阶段性目标落到实处。

综合上述认识，笔者认为，“高级语言程序设计”课程的宿主语言应选择Pascal并兼顾C语言，用不同的方法同时介绍两种语言及其程序设计。引进语言学理论的认识观点，在采用传统方法讲深、讲透宿主语言Pascal及其程序设计的基础上，仅用少量学时数，把C语言作为一种扩展语言对待，用语言比较学方法，针对之前已经介绍过的实例，在对比两种语言表达方式、描述方法的过程中，向学生介绍C语言及其程序设计，使学生在教师的引导下，以自主学习为主，教师引导为辅，加强课外练习，迅速地学习、掌握C语言及其程序设计。以后，在学生学习了“数据结构”课程，掌握了抽象数据类型的概念和要领之后，再设置“面向对象的程序设计”，以讲座类课程（12~18学时）的形式开展教学，同时将一个精选的课程设计贯穿在“高级语言程序设计”、“数据结构”、“面向对象程序设计”系列课程中，通过加强实践环节的练习和训练，可取得提升学生自主更新知识、独立思考、分析问题和解决问题的能力，仰恩大学、石家庄经济学院的改革，用实证的方法证明了这种方法的可行性和有效性，对提高专业人才培养质量有重要意义。

1.3 课程安排在哪个学期比较合适

目前，绝大多数学校将这门课程安排在第一学年，不少学校甚至安排在第一学年的第一学

期。实际上,这样的安排是不恰当的,也是不科学的。首先,计算机科学与技术是在数学和电子科学基础上发展起来的学科,数学是这个学科最重要的基础。本科教育作为高等教育中的基础教育,数理基础课程、电子技术基础课程的教学任务占有相当的比例,而这些基础课程的学习,需要学生静下心来,尽可能排除各种干扰,潜心攻读,方有可能取得较好的成效。而同期开设“高级语言程序设计”课程,对数理基础课程的学习产生的不利影响不可低估。从多年的教学经验和实例来看,许多对程序设计有兴趣,甚至在高中阶段就参与了信息科学省市级竞赛并取得优异成绩的学生,尽管编程能力很强,有些后来经过算法设计进一步的训练,在国家级、国际竞赛中取得了较好的成绩,但由于前期忽视了数理基础课程的学习和训练,后续一系列重要的(专业)基础课程的学习表现不佳。以后,在参与更高层次的研究生入学申请、著名企业的入职竞争中,遇到困难。毕竟,算法设计和编程能力不代表计算机科学与技术专业的全部。其次,学习“高级语言程序设计”课程,尤其是宿主语言选择成分比较琐碎、语法设计不够规范的C语言,对刚刚从中学应试教育体系下走出来,进入大学的一年级学生来说,由于其数学基础、智力深度、思维方式、学习方法、自主学习习惯、自律能力等诸多方面还存在不足,要同时承担数理基础课程、计算机程序设计基础课程的学习是比较困难的。再加上各高校课程体系和培养方案设计、实施不科学,常见的情况是,两方面都没有学好,没有达到应有的高度。

2 高级语言程序设计教学应该坚持的原则

2.1 讲授程序设计和讲授语言并重的原则

“高级语言程序设计”课程在我国计算机学

科教育中已有几十年的历史,课程教学从1985年前以讲授程序语言为中心,已逐步转变到以讲授程序设计为中心。这个中心的转移使得学生的程序设计能力有了显著增强,但也产生了一个副作用,即学生轻视对语言本身(语言的特点和风格)的学习,使得学生对程序设计语言的基本概念理解不清,学习新语言的能力不强。为了克服这个不足,课程教学的中心应该坚持讲授程序设计和讲授语言并重的原则,将两者提到同等重要的地位,使两者在教学中有机的结合起来。这样做有利于学生学习与高级语言密切相关的后续课程,也有利于学生将来更新计算机科学知识,快速学习、掌握新语言及其程序设计。

2.2 将“计算机学科方法论”融入课程教学中的原则

科学哲学界公认,任何一个学科,学科范型的建立标志着这个学科的前学科时代结束,未来从事这个学科的科学研究和应用,有基本的“套路”可循。随着人们对计算机学科认识的深入和学科范型的建立^{[2][3]},一些人开始意识到学科方法论对认知计算机学科、发展计算机学科有着重要意义。随着学科的发展,新知识的不断累积,在学科体系日益庞大的情况下,如何在教学、科研和学科发展中牢牢把握、突出重点,已成为能否实现内涵式教学模式、精简课程体系的关键。按照学科范型理论,在该课程的教学过程中,应该将计算机科学与技术的学科特点、学科知识组织结构、学科形态、基本工作流程方式、核心概念、典型方法与典型实例融入教材中,贯穿在教学过程的始终。

2.3 与后续课程相互衔接的原则

“高级语言程序设计”课程是“数据结构”、“算法设计与分析”、“计算机组成原理与CPU设计”、“编译原理”、“UNIX程序设计”、“操作系统原理”、“数据库原理”等一系列后续课程的先修课程,它们之间存在着密切

的联系。在该课程的教学中的教学中要注意与它们的衔接,分清楚哪些内容应在哪门课程中讲授,哪些地方需要留下伏笔,分清主次,形成呼应,同时避免内容的重复。这样,就能够很好地实现循序渐进,促进融会贯通,产生水到渠成的效果。

“高级语言程序设计”课程尤其要与“算法设计与分析”、“程序设计方法学(选修)”课程的内容进行恰当的分工。“高级语言程序设计”课程的教学立足点应该是针对各种简单的计算问题、趣味性问题的,在已知问题求解方法和算法的基础上,利用宿主语言和基本的程序设计方法、技术,介绍如何进行编程,解决这个问题,而非深入介绍算法设计方法,也非深入介绍程序设计技术。因此,该课程既不能“喧宾夺主”,也不能“越俎代庖”。

2.4 坚持重视实践性教学环节的原则

“高级语言程序设计”课程具有实践性强的特点。其实践性教学环节包括课外做一些习题和上机编程、练习两部分。上机编程部分又分为基础实验和课程设计两个环节。教学中,应注意下列几点。

(1) 及时答疑、批改作业,精讲多练。做习题是学好任何一门课程都需要的重要的实践性环节。要想熟练掌握一门程序设计语言及其基本的程序设计方法、技术,必须编写和调试一定数量、质量的程序。实践证明,没有足够的源代码编程量,高级语言及其程序设计的许多概念、技术是很难深入体会和扎实掌握的。经验告诉我们,本课程的源代码编程量之总和,不应低于2000行,最好能达3000行以上,尽可能涉及多种类型的趣味程序设计练习题。

及时答疑、批改作业是保证教学质量的重要手段。课程答疑主要分为面对面现场答疑和网上答疑两种方式。现场答疑的好处是,对于共性问题可以统一讲解,但现场答疑受时间和空间因素的限制较多。而网上答疑则突破了时间和空间的限制,借助于QQ群这个网络平台,在教师的引导

下,学生可以在群中广泛探讨、分析问题。学生的留言、教师的答复都可以在网络平台中让学生共享。因此,网上答疑具有较灵活的时间自由度和较广泛的信息共享度。要保证网上答疑的效果,就要求教师每天分出一定的时间,在网络上与学生互动、交流、答疑解惑。

批改作业是常规教学的重要环节,既是课堂教学的补充和延伸,也是教学信息反馈的重要途径。然而,“高级语言程序设计”课程教师批改作业工作量大,要消耗教师极大的时间和精力。借助网络软件在线批改作业是一种经济可行的方法。但是,在线批改作业使得教师脱离了学生,不能真正了解学生对知识点的掌握程度。根据经验,保证手动批改至少三分之一的作业,才能及时了解学生的学习情况。

(2) 重视上机实践环节。本课程的实践内容主要是设计并调试程序。在实验教学中,教师一定要将程序调试技术作为教学的重点之一,让学生全面掌握它。否则,将给后续课程的大型程序调试环节带来后患,这一点应引起足够的重视。

此外,设置开放式的课程设计是进一步提高课程教学质量的有效途径。这种方式深受学生欢迎。在教学中,随着难度的递增,笔者要求学生依次完成管理信息系统的升级式开发,依次开发“基于数组”、“基于链表”、“基于文件”的“学生成绩管理系统”,并允许学生充分地扩展软件的各种功能。在软件的升级式开发过程中,学生充分体验了结构化程序设计的思想,以及不同数据类型(数据结构)在程序设计和软件开发中的作用。这一开放式的课程设计,使学生获得了前所未有的成就感,调动了学生学习的积极性,提高了学生对专业知识的学习兴趣。

3 高级语言教学中的几个关键环节和知识点的处理

任何课程的教学,初看知识点很多,但引

入学科方法论的思想方法后,最关键的环节和知识点一下子就缩小范围,往往就归结为那么几个了。缘此,在教学中必须抓住和处理好这几个关键环节和知识点,否则教学质量难以保证。

在这门课程的教学中,类型定义和声明、循环控制结构、参数传递、递归程序、指针及链表是最为关键的。对这几个关键知识点的处理,在很大程度上决定了课程教学质量的优劣。

例如,循环结构是学生遇到的一个关键点,处理方法就是加大程序设计的练习量,使学生在大量练习中掌握循环结构的灵活应用。

又如,参数传递是子程序设计中的一个重点和难点,这部分的教学要充分借助于调试工具,通过在编译环境中监视变量的生存期、变量的值、变量的地址等相关信息,理解参数传递的几种方式。

再如,递归不仅是课程的难点,也是计算机学科的一个核心概念和典型方法。递归程序如何设计?递归程序如何执行?这是学生理解递归概念的两个重要的关键点。在递归程序执行的讲解中,要充分借助调试工具,让学生理解栈的概念及工作原理。

4 关于教材建设

2010年以来,笔者按照新的理念和指导思想,创作了国家级规划教材《高级语言程序设计》^[1]。该教材基于计算机科学与技术一级学科人才培养科学理论体系,依据计算机科学系列教材一体化设计的纲要,选用Pascal作为宿主语言,全面介绍高级语言及其程序设计的基本内容,包括基本概念、类型说明、基本结构、设施、成分和控制机制以及程序设计的基本方法和技术,同时,用语言比较学的方法,介绍高级语言C及其程序设计。教材的创作,充分体现出本文及课程教学的指导思想和观点。该教材有两个鲜明的特点,一是将计算机学科方法论的内容融入其中,

设法建立起各知识点之间的联系和结构,这对学生网络知识结构,促进融会贯通,具有十分重要的意义;二是例题与习题的选取十分讲究,以典型例题为载体,对具有普遍意义的科学研究方法、过程和规范作了深入浅出的介绍。

为了配合课程的教学,笔者专门为实验课编写了内部讲义——《高级语言程序设计实践教程》。该讲义将Pascal和C的Debugger调试技术分散在各个章节中,与知识点相结合,贯穿于整个课程的教学。仰恩大学计算机教学改革试点班2008级、2009级、2012级分别试用了课程主教材(2012级试点班同时正在试用实验教材),石家庄经济学院计算机教学改革试点班2010级、2011级、2012级分别试用了主教材和实验教材,效果很好,学生编程能力大幅度提高。仰恩大学试点班的毕业生就业情况很好,受到用人单位的好评。该讲义不久将由电子工业出版社公开出版发行。

5 结语

提高“高级语言程序设计”课程的教学质量,需要教师做有心人,在实践中勤于思考,善于总结,长于积累,肯下硬工夫。只有这样,才能不断把课程的教学质量提高到新的高度。

【参考文献】

- [1] 赵致琢,刘坤起.高级语言程序设计[M].北京:国防工业出版社,2010.
- [2] 赵致琢.高等学校计算机科学与技术学科专业教育(修订版)[M].北京:科学出版社,2000.
- [3] 赵致琢.计算科学导论(第3版)[M].北京:科学出版社,2006.
- [4] 刘坤起.高级语言程序设计教学的若干问题[J].程序设计语言研究与教学,1998.
- [5] 徐宝文.关于程序设计语言教学的几个问题[N].计算机世界报,1994-9-14.