

“程序设计基础”课程教学实施方案

何钦铭 颜 晖 苏小红 叶乃文

摘 要:“程序设计基础”是计算机科学与技术专业的核心课程之一。在教育部高等学校计算机科学与技术教学指导委员会统一组织下,作者针对程序设计基础课程的培养目标和教学实施中普遍存在的问题,提出了课程教学实施方案。本文介绍了该实施方案的总体框架,并围绕教学内容组织、重点与难点问题、实践环节等方面的内容分析了课程实施的要点,对在不同类型计算机人才培养中课程实施的不同侧重点进行了说明。

关键词:程序设计基础;计算机科学与技术专业;核心课程;课程教学;教学实施

“程序设计基础”是计算机科学与技术专业的核心课程之一,主要讲授程序设计语言的基本知识和程序设计方法,使学生了解高级程序设计语言的结构,掌握程序设计的思想和方法,以及基本的程序设计过程和技巧,具备初步的分析问题和利用计算机求解问题的能力。

程序设计基础是一门教学难度较大的课程。作为程序设计入门课程,本课程是学生深入接触计算机学科领域知识的第一门课程,因而,学生往往难以理解程序设计的思维方式,对教学内容似懂非懂,而掌握良好的程序设计方法则更加困难;在课程教学实施中还存在着学生基础差异大的问题,教师很难把握教学进度与教学内容深度;本课程的主要目标是让学生掌握程序设计的方法和技能,即培养学生程序设计的能力,而如何通过切实可行的编程训练来提高学生的程序设计能力,是许多从事这门课程教学工作的教师非常关心的问题。

近年来,许多高校的任课教师为了提高课程教学质量,在教学内容、实践体系、教学方法和教学手段等方面进行了积极的改革,取得了一些成效。特别是,一些学校在程序设计基础精品课程的建设方面积累了丰富的经验。为了很好地整理和总结这些经验,根据教育部高等学校计算机科学与技术教学指导委员会的统一部署,由浙江大学计算机科学与技术学院何钦铭教授、哈尔滨工业大学计算机科学与技术学院苏小红教授、北京工业大学计算机学院叶乃文副教授和浙江大学城市学院颜晖教授等高校的相关教师组成课题组,结合各校在本课程长期教学改革中所取得的实践经验,共同研究程序设计基础课程教学实施方案,供承担本课程教学任务的广大教师参考。

目前,程序设计基础课程实施方案已由高等教育出

版社出版^[1]。该实施方案以过程型程序设计语言为基准,分析程序设计入门课程的教学实施方法。其特点主要是:

(1)针对性——重点针对教学过程中普遍存在的问题和疑惑;(2)全面性——从教学内容设计、教学要点、实践体系建设等方面给出指导性意见;(3)分类性——充分考虑不同层次、不同类型学校培养人才的差异性。

本文介绍了该实施方案的总体框架,并围绕教学内容组织、重点和难点问题、实践环节等方面的内容分析课程实施的要点,最后说明在不同类型计算机人才培养中课程实施的不同侧重点。

一、课程定位与教学实施方案总体框架

1. 课程的定位

程序设计基础课程主要涉及学科方法论中的设计形态的内容,设计能力的培养是核心,加强实践是培养设计能力的重要途径。

在教学基本内容上,本课程包含了过程型高级程序设计语言的基本内容,以及程序设计方法学、数据结构与算法基础等方面的初步内容。经过本课程的学习,使学生理解计算机学科问题求解的基本特点、过程型高级程序设计语言的构成和特点,以及数据表达、流程控制、算法和程序设计等方面的知识,掌握程序设计和软件开发的基本思想、方法和技巧。

在课程能力培养目标上,从提出问题、设计算法、选定数据表示方式,到编写代码、测试和调试程序,以及分析结果的整个程序设计过程中,培养学生抽象问题、设计与选择解决方案的能力,以及用程序设计语言实现

何钦铭,浙江大学计算机科学与技术学院教授;颜晖,浙江大学城市学院教授;苏小红,哈尔滨工业大学计算机科学与技术学院教授。

方案并进行测试和评价的能力。

2. 教学实施方案的总体框架

根据本课程教学实施过程中存在的问题,以及课程自身的定位,课题组在设计教学实施方案时,重点考虑以下两方面的问题:

(1) 教学内容的组织与传授。如何合理地组织教学内容,使之较好地反映课程教学的重点和学生程序设计方法的认知规律;如何解决教学内容中的难点和重点问题,使学生能更好地理解相应知识点。

(2) 程序设计能力的培养。如何建立良好的编程能力实践训练体系,使大多数学生能得到基本的编程能力训练,同时基础较好的学生能得到更加深入的训练机会。

目前,已经完成的教学实施方案具体包括以下9个方面的内容^[1]:

(1) 基本描述。课程名称、课程性质、预备知识要求和基本学时(含授课、实验和选学的课程设计)。

(2) 教学定位与要求。基础知识要求、能力培养要求、实践教学要求、作业要求和授课要求。

(3) 内容矩阵。两级知识点和相应的讲授、实验学时,其中必修部分10个一级知识点,40个二级知识点;选学部分1个一级知识点,8个二级知识点。

(4) 内容大纲。针对内容矩阵中的11个一级知识点,分别给出基本学时、教学要求、教学内容、难点和重点。

(5) 讲授中要注意的问题。教学内容的组织方式、教学策略、重点和难点的解决思路,以及教师应具备的背景知识。

(6) 教材建设中要注意的问题。知识体系及内容组织方式,示例、习题与上机实验题的要求等。

(7) 课程实验。基础实验和综合实验的内容、要求和实施建议,并针对科学型、工程型与应用型专业给出了2个综合实验的实例。

(8) 课程设计。实施建议和具体的应用示例。

(9) 课程考核的基本要求。笔试和机考的考核重点和题型,以及总评成绩的组成建议。

其中,(1)、(2)是对课程的基本描述和整体定位;(3)、(4)、(5)、(6)涉及课程教学内容的组织与传授方法;(7)、(8)、(9)涉及程序设计能力的培养方法。

二、教学内容的组织方式

程序设计基础教学内容的组织方式主要有两种:以语言知识为主线、以程序设计为主线。目前普遍采用的方式是前者,经过一些高校多年的探索实践,以程序设计为主线的组织方式也取得了成功的经验。

以语言知识为主线的方式主要根据语言的表达功能,

同时考虑程序设计的学习进程,以完整的语言表达模块为基础来组织教学内容。程序设计语言的核心是两大表达功能:数据表达和控制表达。为了实现用有限的语法表达客观世界中各式各样的数据类型,过程式程序设计语言在提供标准数据类型的同时,提供了构造数据类型的手段,比如数组、结构、指针等;在控制表达方面,则提供了语句级控制(如分支、循环)和模块级控制(如函数)。

所以,以语言知识为主线的教学内容组织方式往往以上述语言表达模块为单位,同时考虑程序设计的学习进程来组织教学内容。典型的教学内容组织路线是:概论—数据类型和表达式(标准类型)—分支—循环—数组—函数—指针—结构—文件。

以程序设计为主线的方式则重点围绕学生程序设计能力培养,以程序设计的学习进程为主,打破语言模块的完整性,循序渐进地组织相关教学内容。在教学中,以语言知识为主线的教学内容组织方式重视语法知识的完整性,侧重以语法的理解带动程序设计能力的培养;以程序设计为主线的方式则更加强调编程实践,侧重以编程带动对语法知识的深入理解。

因此,以程序设计为主线的教学组织方式往往按照程序设计的渐进学习进程,结合实际案例来组织和实施教学,将程序开发过程的相关内容贯穿和渗透到每个知识点的讲授中去。一种以程序设计为主线的教学内容组织路线是:概论—初识程序设计—语句级控制(分支与循环)—数组应用—模块级控制(函数)—指针、结构应用—程序设计进阶(递归、链表结构)—文件应用。

其中,“初识程序设计”通过若干简单、完整的程序使学生认识程序设计的基本概貌、了解程序设计的核心概念,并使学生能尽快进入编程实践环节。而有关数据类型和表达式方面的有关内容则通过相关的案例分散到有关章节中。此外,为了使学学生尽快理解程序设计的基本思维方式,将部分难点内容放到课程的后期中教学中,避免内容难点干扰学生对程序设计基本思维方法的理解。

在采用以程序设计为主线的实施方式时,可以将课程内容分为若干渐进式的程序设计学习阶段。在每个阶段的教学过程中,通过案例讲解程序设计的思想和方法,引出知识点,必要时再使用多种方法(如多媒体动画)进行深入的讲解与分析,并通过布置思考题和实验题,让学生通过学习思考和大量的上机练习培养编程能力,掌握知识。

由于本课程的主要教学目的是培养学生的程序设计能力,所以我们建议可以采用以程序设计为主线的教学内容组织方式。

三、重点和难点的解决思路

教师只有从学生的角度体会“学”之困惑,反思

“教”之缺陷，因学思教，才能由教助学，明了如何通过“教”帮助学生学习，体现现代教育以人为本的思想。程序设计基础课程对于许多学生来说，是第一次接触编程，因此，学生往往难以理解程序设计中分析问题、解决问题的方式，尤其对于其中的难点问题。

所以，教师不但要研究教学内容，更要研究教学对象——学生，根据学生的学习特点，研究教学内容传授中重点和难点的解决办法。

实施方案总结了循环、函数、递归、指针和文件等五大方面的重点和难点的问题，并给出了解决思路。

循环：首先使学生理解循环，掌握循环的三种常见的控制方式（计数控制的循环、条件控制的循环、标记控制的循环）；学会使用循环不变式（表示一种在循环过程进行时不变的性质，不依赖于前面所执行过程的重复次数的断言）来设计循环，以及正确理解和设计嵌套循环。在教学中，可以通过实际案例将这些内容有机地组织在一起，做到由浅入深、循序渐进。同时穿插程序调试和排错方法的介绍，说明如何通过单步运行、设置断点、观察变量中间运行结果等方法来对程序进行排错。

函数：首先使学生了解模块化程序设计的概念，了解在函数的功能、规模和接口设计中应遵循的原则，通过案例讲解和上机实践使学生掌握函数的设计方法。

递归：首先用学生熟悉的例子引入递归的概念，介绍如何用递归函数解决用递归方法描述的问题，结合实例分析和程序运行观察，使学生理解递归函数的执行过程和变量栈的变化过程，以及递归和迭代的区别。

指针：结合应用实例，首先使学生了解指针的作用和重要性，理解指针变量与基本类型变量相比的特殊性。在讲解指针变量作为函数参数的相关内容时，如数组名作为函数参数、结构指针作为函数参数、函数指针作为函数参数等，可以将这些内容分到不同的章节中去介绍，以分散指针的难点，做到逐步深入。结合实际应用背景和实际案例，介绍如何用动态内存分配机制实现动态数组（数组元素个数为变量而非常量的数组）、链表等较为复杂的动态数据结构，以及使用数组和链表解决实际问题时存在的区别。

文件：首先使学生理解计算机中的流（一般称为数据流，也有叫做字节流、比特流的）的概念，结合实际应用中数据长久存储的需要，讲解文件的概念，并使学生明白文件中不同的数据格式由应用程序来负责解释，进而讲解文本文件和二进制文件的区别；结合案例介绍文件操作函数，以及文件读写的基本程序框架，让学生在掌握文本文件操作的基础上，对比了解二进制文件的操作方式。在教学安排时要注意给学生充分的时间上机练习文件的操作。

四、实践教学环节与考核

本课程的主要教学目的是培养学生的程序设计能力，而实践教学环节则是培养程序设计能力的最重要环节。考虑到学生的基础不同，应尽量设计有层次和梯度的实践训练内容，使不同水平的学生都能得到最大可能的提升。

本课程完整的实践环节包括课程实验和课程设计两个部分，其中课程实验由基础实验和综合实验组成。应按照循序渐进的方式设计实践教学内容，鼓励学生积极参加课外实践活动（如程序设计竞赛等）。

课程实验与课堂讲授内容同步，形成配套的实验训练单元，鼓励学生从第一周开始练习编程，并贯穿始终。其中，基础实验的主要目的是使学生通过训练，加深对某个知识点的理解；综合实验主要针对多个知识点进行综合训练，加深学生对课程内容的整体认识。

课程设在课程主要内容讲授完毕后进行，可以在本课程的后半期进行，或者在后续学期开设一门独立的课程，其主要目标是培养学生研究性学习及综合应用能力。通过大型程序的编程训练，进一步培养程序设计能力和进行初步的团队合作的训练，深入理解和真正掌握结构化程序设计的思想和方法，以及使用简单数据结构和算法的基本程序设计能力。

在实践内容设计时，要考虑有足够的编程训练量，使学生的程序设计能力得到确实的提高。一般来说，学生在本课程学习期间应完成 2 000 行以上的累计代码量。基础实验中，每个问题的规模可以比较小（约 40 行以内）；综合实验中，每个问题的规模中等（约 100 行或以上）。

为使实践训练能落到实处，并能给出比较客观、公正的实践评价，建议采用具有自动评价学生程序功能的程序设计上机练习及考试系统。同时，在课程考核方面，将课程考试分为笔试和机考两部分，期末总评成绩建议按照平时 20%、笔试 50%、机考 30%的比例计算总评成绩。

笔试：采用闭卷形式，重点考查学生对基本概念和基本方法的掌握程度。建议题目类型包含基本概念选择/填空（考查基本概念的掌握程度）、阅读程序写出运行结果（考查跟踪程序运行的能力）、程序填空（考查阅读程序和设计程序的能力）、程序改错（考查分析程序错误的能力）和编写程序（考查设计程序的能力）。

机考：在规定时间内，根据给定的题目设计程序，并最终调试运行。建议题目类型包含程序填空、程序改错和编写程序。

五、科学型与工程型、应用型分类教学的不同侧重点

教师在具体实施时，需要根据学生的特点，参考实

施方案的建议,确定课程的具体教学内容、实践体系、实施方式等。总体来说,这门课程是很基础的专业课程,在教学内容方面,不同类型、不同层次学校,其教学内容差别并不会很大,但在实践环节和能力培养侧重点上还是存在差别。

根据教育部高等学校计算机科学与技术教学指导委员会发布的《高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)》^[2],计算机专业人才培养分为科学型、工程型和应用型三类。根据不同类别的人才培养目标,本课程可有不同的程序设计能力培养侧重点。

针对科学型教学,主要应从“培养基础、综合设计、研究创新”三个层次设计实践教学内容,鼓励学生积极参加课外实践活动。教师在课堂讲授中,注重与后续课程“数据结构与算法”的联系,侧重于算法的设计、分析和优化,侧重于学生提出问题、解决问题能力的培养,以及研究和创新能力的培养。给学生布置的作业应该具有一定的开放性,甚至可以让其自主选择或者设计一些有挑战性的 ACM/ICPC 竞赛题目作为作业。学生的学习方式也不仅仅是听课和完成作业,还应该有自主学习的内容。教师在考核学生完成作业的质量时,不仅仅是看学生是否达到了预期的功能,还要看学生编写代码的质量,关注使用的方法是不是最优的方法。

而针对工程型、应用型教学,主要应从“培养基础、

综合设计”两个层次设计实践教学内容。教师在课堂讲授中,注重与后续课程“软件工程”的联系,在算法方面更侧重于算法的设计和应用,以及初步的工程方法的培养。相对于提出问题的能力而言,更侧重于学生解决问题能力的培养;相对于最优算法的选择,更侧重于所选择的算法是否能够解决问题;相对于研究和创新能力而言,更侧重于编程实践能力的培养。要求学生完成编程作业的代码量要达到一定的规模,不仅注重程序的可用性,还要注重程序的易用性、可读性,以及程序文档的写作能力。

计算机科学与技术不断发展,计算机教学的研究和改革也从未停顿。程序设计基础课程教学实施方案中要解决的问题涉及面广、难度大,许多内容还需要在实践中不断丰富和提高。

参考文献:

[1] 教育部高等学校计算机科学与技术教学指导委员会. 高等学校计算机科学与技术专业核心课程教学实施方案[M]. 北京: 高等教育出版社, 2009.

[2] 教育部高等学校计算机科学与技术教学指导委员会. 高等学校计算机科学与技术专业发展战略研究报告暨专业规范(试行)[M]. 北京: 高等教育出版社, 2006.

[责任编辑: 余大品]

(上接第 87 页)科学技术知识、兄弟学院的经验,学习现代管理知识和管理经验,要结合教学与管理工作中存在的实际问题来学,将学习到的理论知识转化为实际能力的建设。近几年,学院派出 6 名教师到英国皇家农业大学攻读 MBA 硕士学位,派出领导班子成员、中层干部和骨干教师 200 多人次到国内外高职院校考察学习,参加省内外高职理论培训研究。先后组织开展了四次全院教职员工的职业教育观念大讨论,使教职员工进一步解放思想、更新观念,为推进教育教学改革、促进学院发展扫清了思想认识上的障碍,为学院高职教育的顺利发展并取得突出成绩奠定了思想基础。

3. 示范性高职院校文化建设要加强现代职业教育理论的引领作用,提高教育力。学院通过宣传、诠释、研讨、活动参与等方式使广大教职工掌握现代职业教育理论,运用理论指导教育实践,形成培养高素质技能型人才的教育教学能力、工学结合课程的开发能力、工学结合课程的实施能力、职业教学实训基地的建设能力、知识的创新能力,学院各系部也逐步形成专业开发能力、专业建设能力、技术创新能力。

4. 示范性高职院校文化建设要加强现代企业管理理论的运用与实践,提高管理力。学院依据高职教育的职业性、开放性、实践性办学要求,提高管理能力,能够按市场法则整合资源,利用校内校外两种资源,面对就业市场、生源市场、服务市场办学。在当前职业教育教学条件同质化、生源同质化、专业设置同质化的情况下,学院加强人才培养模式、教学内容、教学方法的改革,培养有特色的人才。学院在办学上注重投入和产出,注重效率,注重数量、质量、结构、效益协调发展,坚持有所为、有所不为的方针,充分发挥自身优势和地域经济发展优势,办出特色,办出水平。

参考文献:

[1] 苏荟, 胡宜挺. 高校核心竞争力构成要素及作用机理研究[J]. 辽宁教育研究, 2008(1): 39-42.

[2] 丁宇, 王婧, 蒋国璋. 高校核心竞争力的构成要素及相互关系探析[J]. 中国冶金教育, 2008(5), 11-12.

[责任编辑: 余大品]