

新工科程序设计课程建设探讨

刘培刚, 杨劭辉, 张学辉

(中国石油大学(华东)计算机与通信工程学院, 山东 青岛 266580)

摘要: 新工科建设对高校程序设计课程提出了新要求。囿于课程规划、学生基础等因素, 当前高校非计算机专业程序设计课程普遍存在定位落差、偏语法弱应用、重计算轻实用等问题。面向新工科建设, 本文以提高学生实用程序设计能力为目标, 提出Python+C/C++的程序设计课程互补体系, Python搭建应用、C/C++构造基础(类库); 综合利用各类教学资源, 提出课外加强基础学习、课堂重点讲方法、在线强化练习的教学模式; 采用分阶段能力教学方案, 在课程阶段重算法逻辑训练, 在实训阶段“选择实际案例、引入Windows开发机制”强化应用性和实用性, 效果良好。

关键词: 新工科; 程序设计; 互补体系; 实训

中图分类号: TP311 **文献标识码:** A

Study on the Construction of Programming Courses in New Engineering

LIU Peigang, YANG Shaohui, ZHANG Xuehui

(College of Computer & Communication Engineering, China University of Petroleum, Qingdao 266580, China)

Abstract: The construction of new engineering has put forward new requirements for programming courses in colleges and universities. Subject to curriculum planning and student capacities, there are many problems in the programming design course for non-computer majors, for example, positioning discrepancy, prioritizing grammar instead of application, prioritizing computation instead of practicability. In view of the construction of new engineering, to improve students' ability of practical programming, this paper puts forward a complementary system of programming courses, Python plus C/C++, building the application in Python and constructing the foundation (Class Library) in C/C++. Taking advantage of all kinds of resources, the paper proposes the teaching model of strengthening basic learning outside class teaching, emphasizing methods in class and intensifying online practices. The grading teaching mode is adopted, that algorithms were emphasized in class and the applicability and practicality were strengthened by real cases and the Windows Development Mechanism in practical training scheme. And it has gained good effects.

Keywords: new engineering; programming design; a complementary system; practical training

1 引言(Introduction)

2017年以来提出的“新工科”建设强调学科交叉与应用, 尤其是计算机相关技术与传统工业的紧密结合, 需要大学生具备较强的“利用计算机解决问题”的能力, 对大学计算机教育提出了更高的要求。大学生计算机应用能力的核心是程序设计能力, 而程序设计是大学计算机教育中的重要一环, 引导学生“抽象实际问题(数学建模)、利用计算机解决问题”, 培养、训练学生计算思维, 是理工科学生从事科研、

工作必不可少的基本技能。一定程度上, 程序设计的教学效果直接影响理工科学生对所学专业的认识与学习。目前的程序设计课程普遍存在一些问题, 不利于程序设计教学的开展^[1-3]。

2 程序设计课程中存在的问题(Problems in programming course)

(1) 课程定位落差

绝大多数高校非计算机专业的教学计划将程序设计定位为通识课, 停留在学习使用一门语言的层次。许多非计算机

专业教师对于程序设计教学的态度具有矛盾的两面性,一方面认为编程很容易,自学即可,因此主张压缩学时;另一方面又对程序设计教学效果不满,认为学生没有具备实际编程能力。面对刚刚接触编程的大学生,在缺乏语言基础的前提下,程序设计课程的教学确实很容易演变为程序设计语言的教学。如何提高学生的程序设计能力始终是程序设计教学的核心与难点。

(2)学生基础参差

普通高校入学新生的计算机基础能力参差不齐。许多学生对于一些基本操作比较熟悉,对于娱乐软件的使用非常熟练,但仍有一部分学生很少接触计算机,而且大部分学生没有程序设计基础,对计算机程序和语言知之甚少。因此,非计算机专业的程序设计教学首先要让学生熟悉一门语言,之后才能进行“设计”,间接导致了程序设计教学变为语言教学的窘境。

幸而国家已经开展中小学编程教育实践,如山东省将Python列为小学试点内容、教育部规划在高中教材中增加Python等内容。将来大学新生将具备一定的编程基础,而程序设计教学也可脱离语言层面,向设计靠拢。

(3)语言影响教学内容

非计算机专业程序设计课程主要讲授程序设计的基础内容。教学内容大体分为两阶段,第一阶段以基本程序结构为主,包括顺序结构、分支结构和循环结构,兼顾基础语法和程序分解;第二阶段运用基本程序结构解决问题,兼顾常用算法和多维数据表达。

学习程序设计语言是进行程序设计的基础,目前国内大部分高校理工科选择C语言。C语言功能强大、效率较高,自其诞生以来,始终是底层程序设计的不二之选,在其基础上发展起来的C++更是当前高效程序的主流。但是对于程序设计教学,C语言具有先天劣势^[4]——语法复杂、可见性差,其高级结构如指针、链表,对于初学者来说如坠云雾,内存异常造成的崩溃更是雪上加霜。许多学生反映程序设计难学,其本质是C语言难学。此外,C语言的初级实例以数字、符号类为主,使得C语言的学习过程非常枯燥。因为细节问题,部分学生始终不能理解、接受C语言的语法和格式,导致教学效果受到很大影响。

(4)能力培养困境

程序设计课程对学生能力的培养,着重于学生利用计算机程序解决实际问题的能力,初期主要是解决计算类问题的能力。在具体问题解决过程中,潜移默化地渗透计算思维。程序设计能力培养的根本在于实践,通过解决各种类型问题的大量练习,逐步提高。

练习过程需要有的放矢,不能机械重复,要求教师筛选练习题目,既能保证数量,还要覆盖类型。练习需要学生投入时间,计划课时远远不足,需要大量课外时间。为激发学生的积极性,练习题目结果是否正确、思路是否正确、程序是否完备等应及时反馈给学生,帮助学生改错、改善。许多时候学生面临的仅是一个小问题,一点即通,同时由于教师精力有限,不能实时解答,因此,练习过程中需要建立及时的反馈环境,还需要即时的交流环境和氛围。

3 程序设计课程的目标与体系(Target and system of programming course)

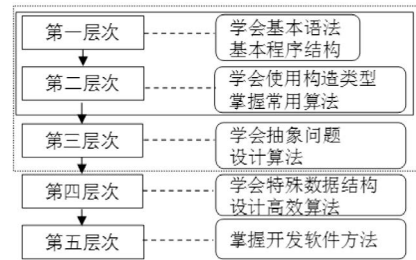


图1 程序设计课程的五个层次

Fig.1 Five levels of programming course

程序设计课程的目标分为五个层次(图1):(1)学会一种程序设计语言的基本语法,如C语言,能够使用基本程序结构解决简单计算问题;(2)学会这种程序设计语言中构造类型的使用,如数组、结构体等,能够解决大数据量交互的相关问题;(3)学会将实际问题抽象为数学问题、逻辑问题,能够正确表述解决过程,即算法设计,并掌握常用算法;(4)学会特殊数据结构的使用,如队列、树、图等,能够根据实际问题选择合适的结构、设计合适的算法,解决问题;(5)学会构建软件,即需求分析、架构设计、系统实现。五层目标分多个阶段实现:第一阶段包括第一二层次,会涉及第三层次,但不深入,一般在第一门语言课中解决;第三四五层次分别为一个阶段,在计算机专业中分属不同课程,其中第三四层次会有交叉。非计算机专业一般达到前三个层次的目标即可。

就目前国内高校计算机教育状况,计算机课程体系通常分为计算机基础、程序设计基础、高级应用三个方面。计算机基础和程序设计基础各设一门必修课,因专业不同稍有出入;高级应用作为选修开设多门课程,一般以应用为主。

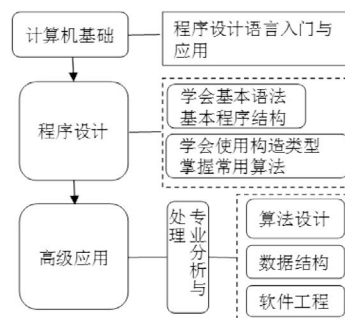


图2 新工科程序设计课程体系

Fig.2 Programming course system of new engineering

基于加强程序设计课程体系的考虑,建议在计算机基础部分增加程序设计入门和在行业中的应用作为引导,在程序设计部分重点实现程序设计课程的前两个层次,在高级应用部分增设相关专业分析与处理的多门课程,如算法设计、数据结构、软件工程等(图2)。

4 Python与C/C++的互补体系(Complementary system between Python and C/C++)

(1)Python的兴起

Python语言语法简洁灵活、弱化类型、易于理解^[5],对

于初学者难度较低,特别是利用第三方函数库,Python可以非常容易地构建解决实际问题的实例,在易入手的同时兼具强大的实践应用性,同时,引入第三方库后,Python语言教学可从图形化应用实例入手,枯燥的数据处理也可通过图形化方式展现,好用又可见,对学生的吸引力远大于其他语言。

非计算机专业学生的主要精力应该放在其专业领域,而不是重复开发计算工具。Python生态圈中各种类库已经超过11万。使用Python,学生要做的是寻找合适的库,然后使用Python语言实现数据传递、库调用等基本操作,真正的复杂处理由库承担。采用Python库,学生能够更容易地搭建系统应用。实际上,Python适合于所有专业学生学习。

(2) C/C++不可抛弃

虽然相对于Python, C/C++语言更难掌握,但是C/C++语言的强大是毋庸置疑的。Python生态圈中的库大部分采用C/C++编写。如果没有现成的功能库可用时,需要自己动手编写实现特定功能库,并丰富Python生态圈。

因此,对于程序设计要求不高、类库支持丰富的专业仅学习Python课程即可;对程序设计要求较高、需要锻炼计算思维的专业,或者Python类库支持不足的专业,则还需要学习C/C++语言。

5 能力为主的培养模式(Ability-oriented training mode)

程序设计课程培养学生的程序设计能力,最直观的体现是,针对某一特定问题,学生能够写出解决问题的程序。这需要学生能够抓住问题本质,将其转化为计算或者逻辑问题,找到或者创造解决问题的方法,编写代码实现。抓住问题本质取决于学生的自然语言理解能力,转化为计算或者逻辑问题取决于学生的数学基础能力,找到解决方法取决于学生的算法设计能力,编写程序代码取决于学生的程序语言使用能力,功能复杂程序的实现取决于学生的程序分治能力。程序设计能力的核心是算法设计能力和程序分治能力,程序分治针对复杂程序,不作为程序设计课程的重点,因此非计算机专业的程序设计课程,重点培养学生的算法设计能力。

程序设计课程是一门实践课程。虽然相关程序设计的理论有很多,但是对于学生来讲,最重要的是学会应用,而不是理论。无论程序设计能力培养还是计算思维培养,必须通过大量的动手实践才能实现。课程学时限制与学生编程语言基础的矛盾,很容易导致程序设计课程的授课内容偏向语言使用而弱化算法设计。以下四个方面解决这一问题:(1)发挥学生的自主学习能力;(2)提供充足的练习机会;(3)提供及时的实践辅导、答疑(提供交流环境);(4)培养自主解决问题的能力 and 习惯。

(1) 发挥学生的自主学习能力

直接为学生提供的学习资料包括PPT课件、教材、参考书、常见错误解析、IDE使用简介等,通过网络可获得的资料主要以MOOC课程为主,辅以百度检索。

在课堂教学中,主要介绍常用的基本语法和典型实例;引导学生正确使用学习资料:从PPT课件中抓主要内容,了解教材的组织、熟悉每一部分内容的位置;在常见错误解析中寻找解决错误的方法,熟悉常见错误;使用IDE学会程序调试

方法;通过MOOC课程复习知识点;使用百度检索陌生问题的解决方法等。

(2) 提供充足的练习机会

程序设计能力、计算思维能力的培养重在训练。我们已经建成在线练习、评测系统,提供了大量各种类型的练习题目,并且对题目按照难易程度划分为简单、中等、提高三个层次,简单题目练习基本语法,中等题目练习复合应用,提高题目锻炼技巧和思维。要求部分接受差的学生至少完成简单题目,大部分学生完成中等题目,部分接受强的学生全部完成。

(3) 提供及时的实践辅导、答疑

大量练习必然产生大量问题,学生通过在线方式在任何时间都能练习,及时的解答和交流非常重要。我们建立了专门的程序设计交流群,学生遇到问题随时提出,既可以在群里提出,也可以和教师单独交流,群里的甚至不需要教师回应,已由其他同学进行解答。通过交流,一能及时答疑并了解学生提出问题、解决问题的情况,二能促进学生交流和互相学习,三能反映学生个体间的接受差异。

(4) 培养自主解决问题的能力 and 习惯

初次接触程序设计的学生,出现最多的问题是语法问题。在这一阶段,需要注意引导学生克服常见错误,如变量定义问题、关键字使用问题、语句匹配问题、括号匹配问题、中英文符号问题等。编译环境会给出错误说明并指出错误所在行,有些错误说明比较直接,如变量未定义(C语言);有些错误说明则非常晦涩、不能理解,出现这样的错误说明时,错误行往往是指出的错误所在行的前一行。只要学生能够记住这些特点,语法错误将比较容易处理。此外,许多学生畏惧语法错误,其根本原因在于错误说明是英文。

运行错误是学生最头疼的问题,可以归为三类:(1)因语法导致的运行错误,典型错误如相等判断写成赋值、循环结构错误导致循环体不执行等;(2)因逻辑不严谨导致的运行错误,如边界条件未判断、特殊情况未处理等;(3)逻辑错误导致的运行错误,问题的处理逻辑不正确。由于思维惯性,很难检查出运行错误,需要引入查错方法,如分段法、插桩法和Debug方法。

提问是一个很好的学习习惯,但不加思考的发问是大学生必须克服的不良习惯。遇到问题首先应独立思考、寻找办法解决,可以查资料、做试验,而不仅仅是寻求他人的答案,实在无法理解再寻求老师同学的帮助,因此,首先传授给学生解决问题的几种实用方法尤为重要。同时,我们提倡学生在网络上检索问题的解决方法,既能加深学生对问题的认识又能提高学生自主解决问题的能力;提倡学生之间交流来解决问题,对于求助的学生可以解决问题,对于被求助的学生可以加深理解。

6 实训综合提高(Improvement through practical training)

程序设计课程的内容以语言使用和基本算法为主,没有整体的认识,与实际接触的软件差别甚大,学生不知道怎样

(下转第40页)