

Python程序设计课程的教学实践

胡艺颖

西安翻译学院 陕西西安 710105

摘要: Python程序设计作为计算机编程语言之一,在大数据、人工智能、网站设计中应用广泛。随着软件编程技术的发展,Python程序设计的重要性越来越明显,各高校在推进Python程序设计教学实践过程中,存在着各种问题。本文主要对Python程序设计课程进行概述,分析了当前Python程序设计教学中存在的问题,结合教学实践情况,提出具体教学实践路径,从而培养学生对Python程序设计的实践水平,更好服务于我国计算机产业发展。

关键词: Python; 程序设计; 课程; 教学实践

前言:

Python程序设计是大学计算机基础教学的核心课程,也是作为大学生学习编程的入门基础课程。相比其他程序设计语言,Python程序设计简介高效,十分适合初学者。该课程的学习,能够让学生初步了解计算机程序设计逻辑与方法,培养学生运用程序设计解决实际问题的能力。在课程学习过程中,对于学生的逻辑思维能力、数学能力、创新能力有较高要求,存在一定的学习难度。在Python程序设计课程教学实践中,应当注重教学模式的创新,基于课程知识特点强化课程考核与教师教学水平的提高,从而使Python程序设计成为一门实践性与理论并重的应用型课程。

一、Python 程序设计课程概述

随着计算机技术的发展,程序设计成为计算机专业学生必须的一门课程,当前,在计算机程序设计中,除了我们熟知的C语言、Java外,还包括Python。相比其他程序语言,Python程序代码简单,适合初学者,简单易用,十分适合没有编程经验的大学生。在使用范围上,Python可以做日常任务,可以设计网站,做网络游戏的后台,可以说使用范围广泛。随着高校计算机专业对编程的重视,Python程序设计课程成为软件专业的必修内容,对于学生未来长远发展及软件开发能力有重要意义。

1.Python 程序设计人才培养方向

Python是软件行业基础语言之一,其应用范围主要在大数据技术与应用、云计算等。大学生在学习该程序设计课程时,主要学习该语言的基本知识和编程内容,掌握程序设计的基本步骤,学会运用该语言进行设计设计,从而解决实际问题。Python程序设计是一门基础性课程,其运用领域还涉及到数据采集与分析、人工智能等领域,为学生日后进入相关工作行业打好基础。对于

当代软件人才培养来说,Python程序设计对于学生的工作技能、职业素养、社会适应能力、交流沟通能力、团队协作能力、创新能力和自主学习能力有直接培养,从而使学生能够适应未来软件编程技术发展,成为一名合格的计算机专业人才,适应未来人工智能、大数据等技术发展对程序设计语言的要求。

2.Python 程序设计课程主要内容

根据Python程序设计人才培养方向,结合课程内容,打造具有现代教育特色的课程内容。在教学目标设定上,使学生能够理解Python的编程模式,熟练运用Python运算符、内置函数以及列表、元组、字典、集合等基本数据类型和相关列表推导式、切片等特性来解决问题,熟练掌握Python分支结构、循环结构、函数设计以及类的设计与使用,熟练使用字符串方法,适当了解正则表达式,熟练使用Python读写文本文件,适当了解二进制文件操作,了解Python程序的调试方法,了解Python面向对象程序设计模式,掌握使用Python操作SQLite数据库的方法,掌握Python+pandas进行数据处理的基本用法,掌握使用Python+matplotlib进行数据可视化的用法,同时还应培养学生的代码优化与安全编程意识。在教学模式上,按照现代软件开发节奏与流程,以任务为载体,将课程内容与实际应用结合起来,使学生以任务驱动为学习逻辑,完成Python程序设计。按照项目实施节点,完成对学生Python程序设计的节点考察,注重学生对知识的应用能力,体现现代教育理论联系实际的教育特色,着重培养学生的实践应用技能。

二、Python 程序设计课程现状

1.教学过程中没有突出Python 程序设计特色

相比于其他程序设计语言,Python语言简单,十分适合初学者,具有简单易学的特点,但是在实际教学

环节, 由于教学理念及方法的问题, 导致Python程序设计并未体现其特色。由于现有教师普遍学习过C语言、JAVA语言, 因此在教学中习惯按照其他语言编程思路展开教学, 这就导致学生很难体会到Python语言的特色, 无法适应学习节奏。在教学环节, 教师习惯于在课堂按照教学大纲进行理论知识讲解, 然后再要求学生完成课后作业, 从而掌握Python编程。对于没有基础的学生来说, 初期很难进入快速学习的节奏, 且缺乏实际应用, 也会导致学生无法体验到Python程序设计的优点。目前Python语言教学课程注重理论与实践相结合的方式, 这种教学模式可以最大程度激发学生的学习热情, 并以实际应用带动学生尽快掌握Python程序设计的精髓。目前来看, 随着Python程序设计的发展, 传统教学模式越来越难以适应Python语言的发展, 不利于学生学习能力的培养, 会导致学生在后续学习过程中失去对Python语言的学习兴趣。

2. Python程序设计知识点多, 难度较大

计算机编程对于学生的数学能力、逻辑思维能力、计算能力均有较高要求, 具体到Python程序设计, 虽然相较其他程序设计语言, Python语言简洁的特性对于初学者有一定的优势, 但是对于没有经验的学生来说仍然存在一定的难度。其中的一些专业名词及运行逻辑, 考验学生的逻辑思维能力, 如果没有坚持学习的耐心, 很容易在前期放弃, 无法真正进入到Python语言的学习环境中。在教学环节, 教师为了让学生打好基础, 会讲述大量基础性知识, 其中既有Python程序设计发展史, 也有其程序设计理念、运行规则等, 而在后期会开始学习一些Python程序设计内容, 这时会对学生的实践操作能力有较高要求。一些学生由于缺乏经验, 无法适应教学节奏, 逐渐选择放弃学习, 导致在期末考试中无法及格。相比于其他课程, Python程序设计知识点多, 学习难度大, 需要教师与学生共同努力, 完成整个教学进度。

3. 课程考核评价方式较单一, 缺乏有效的长期评价机制。

有效的课程考核评价不仅有利于学生正确学习, 更为客观检查学生的学习效果, 同时也能够对教学模式与教学思想改革, 推动教学水平的不断提高。当前, Python程序设计以理论考试为主, 叠加学生平时的出勤成绩, 从而得出学生的课程成绩, 这种考核方式对于Python程序设计具有一定的局限性。一方面, Python程序设计更多体现在实践操作中, 考察学生的应用能力以解决实际

问题为出发点, 这样才能真正体现学生的学习效果, 如果单纯以理论考察的方式, 并不能从中选拔出有创造性与应用型的人才; 另一方面, 理论考察的方式容易导致学生在考前为了取得优异成绩, 采用死记硬背的方式, 虽然这样能够提高成绩, 但是并不能真正反映学生的学习水平, 在后续教学中, 教师也无法根据考核成绩针对性改进教学, 从而导致教学效果不佳。

4. Python程序设计课程缺乏专业性

Python程序设计具有极强的专业性, 因此在实践环节, 需要培养学生以实际需求为出发点开展程序设计。当前, Python程序设计中的实践环节主要以验证型实验为主, 设计型和综合型实验偏少, 且缺少与专业背景需求相结合的实验。导致学生在编程学习过程中机械、刻板地去背代码, 不能自主编程, 程序设计能力较差。对于学生来说, 已有的Python程序设计更多是参考作用, 而在独自设计中, 考察学生的逻辑思维能力与应用能力, 这样才能符合Python程序设计要求。如果单纯依靠背代码, 虽然在一定时间内能够提高Python程序设计水平, 但是长远来看, 作为专业程序设计语言, Python无法发挥自己的实际作用。

三、Python程序设计课程的教学路径

1. 强化师资力量, 注重循序教学

Python程序设计是一门专业性极强的课程, 不仅要求教师熟练掌握Python语言, 还要具备一定的实际项目开发经验, 在教学中能够熟练使用各种教学方法, 将枯燥的理论知识转化为具有实际应用效果的程序设计内容。考虑到当前教师普遍缺乏实际项目开发经验, 因此可以考虑从校外引进优秀Python程序设计人员, 丰富现有教师的项目开发经验, 使更多教师能够在教学中将理论知识与软件项目开发结合起来, 因材施教, 做到学以致用。在教学环节, 教师要认识到学生对Python程序设计课程存在一定的陌生感, 因此要采用分步骤分阶段的教学策略, 由浅入深讲述Python程序设计知识。第一阶段先给学生讲解Python语言的含义和作用, 以及Python程序设计界面的基础操作内容, 让学生认识到Python语言的实用性, 熟悉Python程序设计界面和一些基本的操作。第二个阶段开始教学生学习Python程序设计中的基本语句、程序结构和一些主要的知识点。在学生能熟练掌握Python程序设计的基本语句和程序结构后开始第三阶段的教学。第三阶段的教学主要教学生Python程序设计的整体操作, Python语言的算法知识, 教会学生能自主编写完整的Python程序。

2. 建立过程性考核评价机制

针对当前Python程序设计考核机制单一、无法真实体现学生实践水平等问题,应当建立过程性评价考核机制,这样能够避免学生在考前突击复习,将Python程序设计死记硬背,大量刷题,从而获得优秀成绩的弊端。在建立过程性考核评价机制时,使用考核与学习效果相融合的方式,形成以教师引导、学生学习、小组完成的评价主体,通过平时学习记录、考评测试及课上实践验证考核的评价手段,对学生的课堂表现、平时学习效果、拓展探索及团队合作完成情况进行考核评价。首先是对基础理论知识的考察,包括课前预习复习测试、课上针对本节知识点的实践验证效果及课后章节测试成绩;其次是对知识体系的终结性考核,以选择、填空、程序改错、程序填空、程序设计等形式构成;最后是综合应用考核,通过设置四级课程设计任务,由学生小组选择课程设计题目,锁定成绩级别。

3. 改进优化教学模式

对于Python程序设计教学来说,成功的教学模式应当注重实践操作,在充分利用现代多媒体等教学设备基础上,灵活运用情境教学法、项目教学法、任务驱动法、分组讨论法、案例教学法等多种教学方法,引导学生积极思考、乐于实践,提高教学效果。例如为了增加学生对Python程序设计的应用认识,可以采用情境教学法,模拟实际软件开发环境,让学生学会运用Python语言解决实际遇到的问题,培养学生的创新精神。为了进一步增加学生对Python程序开发的认识,还可以采用项目教学法,以一个完整的Python程序开发为依托,教师将完整的项目开发引入到教学中,引导学生逐步学会程序开发工作,解决开发中遇到的各种问题,实现以项目带动教学,将理论与实践很好的结合在一起,加强对学生职

业能力的培养。此外,还可以采用任务驱动法教学模式,设置若干个学习情境,并将Python程序设计细分为不同的任务,学生按照各自的任务制定完成时间与任务目标,通过任务驱动的形式完成教学。

4. 拓展启发式教学, 引导学生学习

启发式教学是一种有效的教学干预模式,可以帮助学生养成计算思维。第一,教师在引导学生解决问题的过程中,要先让学生反思与回忆,也就是让学生用已经写好的代码来对应教师提出的问题,探讨这段代码是否可以解决这些问题,帮助学生将自己的思维转化成为目的导向和结果导向的思维。第二,在解决问题的过程中,教师要引导学生分步骤地进行自我反思,并对其他小组成员编写出来的代码进行实质性的审查,探讨自己的代码与其他同学的代码存在哪些异同,通过批判性的思考来探讨自己的不足。这样能让学生掌握循序渐进的学习方式。

四、总结

Python程序设计课程教学实践中存在着课程特色不突出、缺乏有效的考核、知识众多、难度大、专业性不足等问题。教师在教学过程中应当强化师资力量,注重循序教学,建立过程性评价机制,改进优化教学模式,激发学生学习兴趣,引导学生自我学习,从而使学生真正掌握Python程序设计,成为一名现代化计算机人才。

参考文献:

- [1]王思霖.新工科Python语言程序设计课程教学实践[J].集成电路应用, 2022(005): 98-99.
- [2]亢华爱.高职院校python程序设计课程实践教学探索[J].休闲, 2021(007): 1.
- [3]贺国平.基于计算思维的Python程序设计课程教学实践[J].集成电路应用, 2022(001): 194-105.