

高职教育中 Python 程序设计课程教学的探索与研究

李桂锦

(湖南交通职业技术学院, 湖南 长沙 410000)

摘要: 近年来, 人工智能领域发展飞速, 社会出现的大量与之相关的新兴产业都需要更多技术型人才支撑。为了顺应市场形势的变化、达到企业用人的要求, 许多高职院校顺应形势开设了人工智能领域相关课程。其中, Python 程序设计课程是最重要的基础课程, Python 语言是当下最流行的程序语言之一, 未来的应用前景十分广阔。为了提升高职院校 Python 程序设计课程的教学质量和效果, 提高学生的就业竞争力, 对该课程进行有效的教学方法的探索与研究是非常有必要的。

关键词: Python 课程; 教学探索; 高职教育

一、绪论

Python 语言最早是荷兰人吉多·范罗苏姆(Guido van Rossum)在1989年发明的, 1991年公开发布。Python 语言是一种面向对象、开源的解释型脚本语言。Python 语言的应用领域十分广泛: 数据分析、自然语言处理、云计算、深度学习、大数据分析、图像处理、科学运算等。在不同领域中, Python 语言不但在用, 而且用得越多, 重要性逐年提高。例如: 目前业内许多大中型企业都在使用 Python 完成各种任务, 如: Youtube、Google、百度等。2021年2月份发布的 TIOBE 语言排行榜显示: Python 语言仅次于 C 语言和 Java 语言排在了第三的位置。

Python 语言之所以会受到越来越多技术人员的青睐, 发展如此迅猛, 离不开本身的优势以及它优越的性能。Python 采用极简主义设计思想, 语法简单优雅, 不需要很复杂的代码和逻辑即可实现强大的功能。Python 语言学习简单、上手快, 同时它是开源免费的, 可以直接安装使用, 还可以对其源码进行修改。Python 还具有良好的跨平台和可移植性, 它可以被移植到大多数平台下面, 例如 Windows、MacOS、Linux、Android 和 IOS 等。除了使用 Python 语言本身编写外, 还可以混合使用 C 语言、Java 语言编写。此外, Python 还提供了丰富强大的库, 可以实现很多强大的功能, 如用于数据分析和科学计算的 NumPy 库和 pandas 库、用于数据可视化的 matplotlib 库等。

教育部高等学校大学计算机课程教学指导委员会在2016年发布了《大学计算机基础课程教学基本要求》, 首次将 Python 语言作为程序设计入门课程的教学语言。教育部考试中心于2017年10月11日发布了“关于全国计算机等级考试(NCRE)体系调整”的通知, 决定自2018年3月起, 在计算机二级考试加入了“Python 语言程序设计”科目。近几年, 越来越多的本科、高职甚至部分中职院校纷纷将 Python 程序设计这门课程纳入人工智能相关计算机类专业的必修课程中, 也有一些院校将 Python 程序设计这门课程作为公共基础课, 在校内开展教学。因此, 在此背景下, 探索如何提高 Python 程序设计课程的教学方法既有意义也有必要, 也

符合当今大数据时代发展的趋势。

二、Python 程序设计课程教学现状

在传统的教学模式中, 类似 Python 程序设计类课程的课堂教学主要包括理论知识教学与实践操作练习。为深入分析传统 Python 程序设计课程教学方法的改进之处, 在本校师生中开展了相关问卷调查, 结果反映出在教学中存在的一些问题, 具体表现在三个方面:

(一) 教学内容偏理论化, 难度和挑战性较大

在传统的 Python 程序设计课程教学中还是以教师讲授为中心的教学模式为主体, 师生互动、生生互动偏少。教师通常结合书本内容, 讲解课程中的理论知识, 理论化程度高、专业名词多, 学生无法深入理解和掌握。比如, 在讲授 while 循环结构相关理论知识点时, 主要侧重于教师在讲台上讲授 while 循环的特点、语法结构、执行过程等。在理论教学阶段, 学生更多的时间只能看着教师在教师机上演示相关知识点的代码示例, 对于学生而言, 印象较深的就是看到一个程序的正确运行结果, 但是学生自身不一定能理解为什么程序要这样写。对于基础较弱, 没有接触过程序语言学习的学生而言, 这种授课方式学习效果会大打折扣, 这种理论化的讲授方式难以培养学生的“计算机思维”, 随着学习的深入、专业化程度提高, 学生的学习难度和挑战性都会增强, 甚至有些学生出现畏难情绪, 学习兴趣和主动性降低。

(二) 实践操作多以学生个人为主, 小组合作学习较少

实践操作也是 Python 程序设计类课程的课堂教学中一个非常重要的组成环节。传统的实践操作主要采用“教师发布任务——学生自主实践——教师指导讲解”学生个体为主, 教师为辅的模式。学生首先根据任务要求, 思考解决问题的方法, 然后编写程序完成任务, 教师一般会提供讲解和参考程序。但这种模式仍旧有一定的局限性: 一方面, 部分编程思维较弱的同学而言, 对于问题的理解模糊, 程序逻辑关系理解不全面, 可能会出现直接复制粘贴代码的情况, 只知其然, 不知其所以然, 实践操作变成了一项“为了完成任务而完成的任务”。另一方面, 实践主要以学生个人为

主体,小组合作学习、团队项目操作开展较少,课程教学没有以学生为中心,讨论式、项目式教学活动匮乏。

(三) 课程考核方式单一

与其他课程类似,传统的 Python 程序设计课程仍旧采用“考试成绩+平时成绩”的考核方式。考试成绩主要依据是学生期末考试的卷面成绩,平时成绩主要包括考勤、作业、上课表现等主观性较强的评价方式。但是,对于程序设计类课程而言,传统的考核方式显然不能全面衡量一个学生的学习效果,课程考核方式单一、注重期末考核而忽略过程性评价,会导致课程教学目标与考核标准不一致的情况。

三、Python 程序设计课程的教学新思路

(一) “混合式”理论教学模式

为了提高学生对 Python 程序设计课程中理论知识的理解及应用能力,教师在教学过程中要注重培养学生的计算机思维和编程素质,提高学生学习兴趣。课堂教学要从“以教师为主体”模式转变为“以学生为主体”的模式,教学方式应该多元化,除了传统教师讲授理论知识,还可以开展线上线下混合、翻转课堂、知识抢答等教学活动,有利于充分调动学生学习的积极性。同时,课堂教学不仅仅局限于课堂中的 45 分钟,课前导学及课后辅导巩固也是不可忽视的重要环节。“混合式”理论教学模式的探究可以从以下环节入手:

1. 课前导学环节。教师可以在课前将课程主要讲授的内容大纲先发布在班级课程群聊、学习通平台等线上平台中。学生可以通过平台自主学习了解本次课程的主要内容,对于内容有个大体的把握与印象,这样在听课时对于一些专业名词等理论性较强的内容不会感到很陌生。学生通过课前自主学习,了解重难点知识,教师可以将学生有疑问的知识点汇集起来,等上课时可以一起讨论学习,便于教师了解学习学生的需求点,提高教学质量。

2. 课中教学环节。一方面,对于理论知识的讲授,教师可以采用“类比教学”的方式做好课堂内容的教学规划。例如:在讲授 Python 序列数据内容的时候,可以根据不同的序列类型的特点进行分类教学。根据序列数据的不可变性,可以将字符串、元组划分为一类,将列表、字典、集合划分成一类;也可以根据序列数据的有序性划分,其中,字符串、元组、列表是有序的,字典与集合是无序的。教师可以通过分别举例子进行说明比较,结合不同序列数据的特点,展开具体知识的教学,对于有共性的序列数据列表、字典、集合而言,它们能进行的增加、修改、删除操作都是类似的,教师可以举一反三演示讲解,提高学生学习的类比学习能力和理解能力。

另一方面,在课堂教学环节中,要采用多元化的线上线下相结合的教学方式。教师可以通过线上平台设置相关知识点的抢答、讨论、随堂练习等互动环节,提高课堂中学生的参与度,让

更多的学生主动的沉浸在课堂中。教师在课堂教学时,也可以将一些相对简单的知识点讲解分配给学生完成,让学生成为“小老师”,其他学生可以提问,讨论,这种翻转课堂的教学模式也是非常适用程序设计类课程的。此外,教师在演示程序代码时,可以将需要注意的知识点或者易出错的语法人为地设置为“代码中的 bug”,程序运行报错时,让同学们主动思考找代码中的错误,这样学生的印象更加深刻,而不只是看到老师演示了一个正确的运行结果而已,同时,也可以让学生认识到学习就是一个不断发现错误、尝试解决错误的过程,学习时应该要克服畏难情绪。

(二) 多元化的团队合作实践操作模式

教师在组织学生进行实践操作时,可以采用小组学习、团队合作实践的方式进行。同时,在设计实践操作案例时,要尽可能地贴近实际、课堂教学理论知识点在实践案例中能大量运用,在每个章节结束时,也可以适当设计综合性强且难度相对高一些的案例。近年来,针对高职学生的职业技能大赛、“互联网+”创新创业大赛、中国大学生计算机设计大赛等各种涉及 Python 语言编程的竞赛日益增多。教师可以将各类竞赛的历年的竞赛题目作为一个实践项目,按照实际比赛的要求进行模拟分组,学生在自主分工、查阅资料、相互帮助、一起讨论、完成实践项目的过程中逐步提高自身的编程能力和职业竞争力。一方面可以让学生更好地使用 Python 语言这个工具为以后参加此类比赛打下坚实的基础,另一方面,也可以更好地满足与社会企业对于计算机人才的要求。

四、结语

本文以高职院校中 Python 程序设计课程教学方法的探究为主题展开,从课堂理论教学、实践操作及课程考核方式三个方面分析了传统教学方式中所存在的问题和局限,提出了“混合式”理论教学模式、多元化的团队合作实践操作模式、采用量化的过程考核方式的 Python 程序设计课程教学方法的新思路。本文提出的课程教学思路可以有效地提高学生的学习兴趣,激发学习内驱力,同时培养学生的编程能力和计算机思维能力。在以后的教学过程中,将进一步强化该课程与后续专业课程之间的紧密联系,更好的发挥 Python 基础课程的作用。

参考文献:

- [1] 范海燕. 信息技术在 Python 教学应用中的思考——以 Python 环境搭建为例 [J]. 湖北农机化, 2020 (01): 93.
- [2] 许洪云, 陈朝焰. 新工科背景下 Python 教学探讨与实践 [J]. 计算机时代, 2021 (02): 96-98.
- [3] 嵩天, 彭远红, 宋文婷. 理解和运用计算生态推动 Python 语言教学改革 [J]. 计算机教育, 2017 (08): 6.
- [4] 黄淑芹, 张海, 王淮生. 兴趣驱动 Python 语言课程教学改革研究 [J]. 电脑知识与技术, 2020 (16): .