

面向多元专业的Python程序设计通识课程教学创新研究

孔子潇^{1*}, 黄文俊¹, 孟 春²

¹国际关系学院网络空间安全学院, 北京

²北京理工大学计算机学院, 北京

收稿日期: 2026年1月8日; 录用日期: 2026年2月5日; 发布日期: 2026年2月14日

摘 要

针对国内高校面向人文社科与新兴学科等多元专业背景的本科生开设的Python程序设计通识选修课存在“教学目标单一化、教学内容普适化、教学案例脱离学科背景、考核方式传统化”等教学实际问题, 分析课程现状并进行问题诊断, 提出“分层分类、学科融合、实践驱动、多元评估”四位一体的教学创新方案, 以期对人文社科与工科类多元专业的计算机通识选修课程教学提供参考与借鉴。通过教学评价调查显示学生课程满意度94%, 说明学生对课程接受度良好。

关键词

Python程序设计, 通识教育, 教学创新, 学科融合

Research on Educational Innovation of Python Programming General Education Courses for Diverse Majors

Zixiao Kong^{1*}, Wenjun Huang¹, Chun Meng²

¹School of Cyber Science and Engineering, University of International Relations, Beijing

²School of Computer Science and Technology, Beijing Institute of Technology, Beijing

Received: January 8, 2026; accepted: February 5, 2026; published: February 14, 2026

Abstract

For general elective Python programming course offered to undergraduates from diverse majors

*通讯作者。

文章引用: 孔子潇, 黄文俊, 孟春. 面向多元专业的 Python 程序设计通识课程教学创新研究[J]. 创新教育研究, 2026, 14(2): 412-417. DOI: 10.12677/ces.2026.142141

such as emerging disciplines and humanities and social sciences in domestic universities, there are practical teaching issues including “single-focused teaching objectives, generalized teaching content, teaching cases detached from disciplinary context, and traditional assessment methods”. This study analyzes the current state of the course and diagnoses these problems, proposing a four-in-one teaching innovation plan of “tiered and categorized instruction, interdisciplinary integration, practice-driven approach, and diversified assessment”, aiming to provide reference and guidance for teaching computer literacy electives to students from diverse majors in the humanities, social sciences, and engineering. Teaching evaluation surveys show a course satisfaction rate of 94%, indicating that students have responded well to the reformed course.

Keywords

Python Programming, General Education, Educational Innovation, Interdisciplinary Integration

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在数字经济与数字化转型的背景下，程序设计能力已成为当代大学生知识结构的重要组成部分。教育部在《高等学校人工智能创新行动计划》等文件中明确提出“推动人工智能与多学科专业教育的交叉融合”[1]。在此背景下，国内众多高校面向非计算机专业，特别是人文社科与新兴交叉学科学生开设 Python 程序设计通识选修课[2]-[4]。

然而，在教学实践中暴露出若干问题：第一，学生专业背景差异巨大，从纯文科的行政管理、法学到新兴工科的数据科学、网络空间安全专业，其数理基础、逻辑思维、学习目标迥异，统一的教学内容难以满足多元化需求；第二，传统以语法为核心的教学模式，难以激发人文社科学生的学习兴趣；第三，教学案例往往脱离学生的专业背景，导致“学用分离”；第四，考核方式仍以编程题笔试为主，未能充分评估学生的计算思维与实践能力[5]-[11]。

针对上述问题，本研究以北京国际关系学院面向行政管理、国际政治、传播学、政治经济哲学(PPE)、网络空间安全、数据科学与大数据技术、法学七类专业开设的“Python 程序设计”通识选修课为实践平台，开展教学创新探索。

2. 课程现状与问题诊断

2.1. 学生专业背景与需求分析(以某学期选课学生为例)

以某学期选课学生为例，对学生专业背景与学习需求进行分析，见表 1。

2.2. 传统教学模式存在的具体问题

- 1) “一刀切”的教学目标：要求所有学生掌握相同的编程技能深度，忽视专业差异[4]。
- 2) “语法中心”的教学内容：按变量、分支、循环、函数、文件、库的线性结构推进，枯燥抽象。
- 3) “脱离情境”的教学案例：多使用数学计算、游戏开发等案例，与人文社科学生关心领域脱节[11]。
- 4) “单一标准”的考核方式：期末笔试占比过高，无法评估实际应用能力与思维过程。

Table 1. Analysis of students' academic backgrounds and needs
表 1. 学生专业背景与需求分析

专业类别	学生人数	平均数学基础	主要学习需求
行政管理/国际政治/ 法学(人文社科)	50%	较弱	自动化办公、数据分析辅助决策、政策文本分析
传播学	17%	一般	社交媒体数据抓取与分析、信息可视化
政治经济哲学(PPE)	6%	一般	经济数据分析、逻辑建模、哲学概念的形式化
数据科学与大数据技术 /网络空间安全(新工科)	27%	强	数据处理、编程能力提升、为进一步专业学习铺垫

3. 教学创新方案设计

3.1. 核心理念：从“教编程”到“培养计算思维”

将课程核心目标从“掌握 Python 语法”转变为“运用计算思维解决本领域问题的能力”。计算思维在本课程中被具体化为：

- 问题分解：将复杂的专业问题拆解为可编程实现的子问题。
- 模式识别：在数据、文本、流程中发现规律与模式。
- 抽象建模：用变量、数据结构、算法描述专业问题。
- 算法设计：设计步骤清晰的问题解决方案。

3.2. “分层分类”的教学目标体系

针对不同专业背景[12] [13]，设定三层目标：

- 1) 基础层(所有学生)：理解计算思维核心概念，能用 Python 解决简单自动化任务。
- 2) 专业应用层(按学科方向)：
 - a) 人文社科组：掌握数据获取(爬虫)、可视化(词云展示)及基本分析；文本处理(正则表达式、词频统计)；自动轨迹绘制。
 - b) 新兴交叉学科组：进一步学习常用算法(排序、查找)、数据结构(列表、字典)、简单机器学习库应用。
- 3) 创新拓展层(学有余力者)：完成跨学科综合项目，解决本专业实际研究中的子问题。

3.3. “学科融合”的教学内容重构

打破传统语法线性结构，按“需”组织教学内容，为不同专业学生配备学科特色案例[14]-[16]，见表 2。

Table 2. Cases of disciplinary feature
表 2. 学科特色案例

教学模块	核心内容	人文社科案例	新工科案例
模块 1：数据获取	基础语法、文件操作	政府公文词频统计	csv 数据计数器
模块 2：文本与数据处理	字符串处理、正则表达式、列表字典	政策文件关键词提取	数据清洗与格式化
模块 3：数据分析与可视化	数据可视化、第三方库使用	词云图生成器	自动轨迹绘制
模块 4：算法思维与应用	函数、简单算法	汉诺塔问题	素数验证与哥德巴赫猜想

3.4. “实践驱动”的教学过程实施

- 采用“课内微型项目 + 期中上机测试 + 期末综合项目”三级项目体系[16]:
- 课内微型项目(每 2 学时): 如“政府工作报告词云”(人文社科)、“程序进度条”(新工科)。
 - 期中上机测试(9 道题): 与基础知识相关的编程题。
 - 期末综合项目(替代期末考试): 3~4 人跨专业组队, 解决一个真实问题, 如“图书借阅系统设计与实现”(需不同专业学生合作)。

3.5. “多元多维”的考核评价体系

- 取消期末笔试, 建立过程性评价体系:
- 1) 平时表现(20%): 课堂参与、微型项目完成情况。
 - 2) 期中机试(20%): 按时正确提交。
 - 3) 期末综合项目(60%): 项目的代码质量、文档规范性、专业贴合度、技术实现、团队协作、展示答辩。
 - 4) 设立“计算思维进步奖”: 特别鼓励零基础但进步显著的学生。

3.6. 教学支持环境建设

- 1) 开发学科案例库: 收集、开发 50+个学科相关案例, 形成开放资源。
- 2) 搭建低门槛开发环境: 统一使用 Python IDLE 和 Visual Studio Code, 避免环境配置困扰。
- 3) 创建“编程诊所”: 每周固定时间提供指导, 针对性解决问题。

4. 效果评估

本研究以某学期选课学生为基础, 收集多源数据, 进行统计分析、效果评估(见表 3), 共涵盖 6 个教学班, 学生总数 18 人(小班教学)。

Table 3. Outcomes of student learning
表 3. 学生学习成效

评估指标	学生人数
课程满意度(85 分以上)	94%
代码作业完成率	60%
期末项目优秀率(85 分以上)	60%
零基础学生坚持率(不退课)	100%
认为“课程有用”	94%

5. 讨论与反思

在 Python 语言程序设计课程中, 学生的培养目标是在熟练掌握 Python 语法规则的基础上, 提升他们在工程实践中发现问题、分析问题和解决问题的能力。为了达到这个目标, 本文采用了“先用后学”的教学方法, 设计了学生感兴趣的情景驱动教学任务, 并设置了多阶段、不同层次的教学实践。比如, Python 编程中有许多第三方库, 无法一一为学生讲解, 通过词云图生成的实例教导学生应用为王, 遇到问题能够解决才是王道, 先去使用再去进一步深入学习, 同学们都觉得受益匪浅。在实例中, 我先为同学们演示如何使用 wordcloud 库将政府工作报告生成词云图, 之后让同学们改写代码实现将《人工智

能+》的报告生成新的词云图(代码片段和项目成果如图 1, 图 2 所示)。这种解决真实问题的学习方法赋予了学习过程意义感, 激发了学习动机, 降低学生的焦虑, 课程考核采取的多元评价方式也鼓励了不同类型的学生。

```
#GovRptWordCloudv3.py
import jieba
import wordcloud
import imageio
#mask = imageio.imread("fivestar.png")
#excludes = { }
f = open("人工智能+.txt", "r", encoding="utf-8")
t = f.read()
f.close()
ls = jieba.lcut(t)
txt = " ".join(ls)
''' w = wordcloud.WordCloud(\
    width = 1000, height = 700,\
    background_color = "white",\
    font_path = "msyh.ttc", mask = mask\
)'''
w = wordcloud.WordCloud(\
    width = 1000, height = 700,\
    background_color = "white",\
    font_path = "msyh.ttc", max_words = 15\
)
w.generate(txt)
w.to_file("grwordcloudai.png")
```

Figure 1. Diagram of code snippet

图 1. 代码片段图



Figure 2. Diagram of wordcloud report

图 2. 报告词云图

6. 结语

综上所述, 面向人文社科与新兴学科等多元专业背景的本科生开设的 Python 程序设计通识选修课, 必须打破计算机专业课程的教学定势, 以“培养计算思维”为根本目标, 以“分层分类, 学科融合”为核心策略, 以“实践驱动”为实施路径, 以“多元评价”为保障机制。本课程通过设计情景驱动教学任务, 并设置多阶段、不同层次的教学实践, 举一反三, 可以有效提升学生发现问题、分析问题、解决问题的能力, 通过编程的过程, 学生逐步培养清晰的逻辑思维, 并给出相应的代码实现。这样的学习过程提高了学生的动手能力, 学生也获得成就感, 保持学习热情。通过这样的教学方式, 使程序设计教育真正服务于各专业学生的数字化能力培养, 为培养跨学科的复合型人才提供有效支撑。

未来可积极争取学院支持, 将跨学科项目纳入实践教学体系, 针对不同学科背景学生规划差异化计算思维发展路径, 以及探索人工智能辅助的个性化教学在通识类课程中的应用。

基金项目

国际关系学院教育改革创新项目(面向应用能力培养的 Python 程序设计通识课程改革、面向新文科专业创新发展的网数智通识教育课程体系建设调研与改革)。

参考文献

- [1] 教育部. 高等学校人工智能创新行动计划[EB/OL]. 2018-04-02.
https://www.gov.cn/zhengce/zhengceku/2018-12/31/content_5443346.htm, 2025-12-25.
- [2] 嵩天, 礼欣, 黄天羽. Python 语言程序设计基础[M]. 第2版. 北京: 高等教育出版社, 2017: 74-78.
- [3] 嵩天, 黄天羽. Python 语言程序设计教学案例新思维[J]. 计算机教育, 2017(12): 11-14, 19.
- [4] 肖敏, 陈永强. 面向人文社科专业的 Python 程序设计课程改革[J]. 计算机教育, 2021(5): 45-48.
- [5] 李静, 胡国华. Python 语言程序设计课程思政资源挖掘及教学路径探索[J]. 计算机教育, 2020(1): 24-28.
- [6] Brennan, K. and Resnick, M. (2012) New Frameworks for Studying and Assessing the Development of Computational Thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, 13-17 April 2012, 1-25.
- [7] 郑南宁, 王芳, 孙怡铭. 人工智能本科专业建设与课程改革探索与实践[J]. 计算机教育, 2025(12): 1-2.
- [8] 陈小强, 黄志鹏, 胡翰. 生成式 AI 赋能 Python 程序设计课程教学设计与实践[J]. 计算机教育, 2025(12): 109-113.
- [9] 俞文昌, 马小琴, 赵盼. 工程认证背景下 Python 程序设计课程改革[J]. 计算机教育, 2025(11): 154-159.
- [10] 李知菲, 彭浩, 王晖. 基于知识图谱的 Python 程序设计课程个性化 AI 助学模式探索[J]. 计算机教育, 2025(8): 200-205.
- [11] 周丽蓉, 全华凤, 李宜汀, 等. 财经高校 Python 程序设计课程改革实践路径[J]. 计算机教育, 2025(9): 202-207.
- [12] 胡艳华, 崔亚楠, 韦灵. 基于“OBE + 思政”理念的 Python 程序设计课程改革与实践[J]. 计算机教育, 2025(2): 144-149.
- [13] 孙洪春, 刘靖国. 基于大学生创新能力培养的高校课堂教学手段与方法改革研究[J]. 教育进展, 2023, 13(6): 3291-3296.
- [14] 刘竞泽, 于海涛. 新文科背景下“Python 程序设计”课程教学改革研究[J]. 创新教育研究, 2024, 12(2): 656-663.
- [15] 肖哲, 刘国亮, 刘强, 王鑫. 课程思政在《Python 程序设计》教学中的应用实践[J]. 创新教育研究, 2021, 9(4): 823-827.
- [16] 冀全朋, 严海升. Python 程序设计课程教学改革与实践——基于项目教学模式[J]. 西南师范大学学报(自然科学版), 2021, 46(11): 90-95.