

基于 OBE 理念的 Python 语言程序设计 课程教学改革与实践

王瑞平 诸强 周雪忠 熊轲

北京交通大学计算机科学与技术学院, 北京 100044

摘要 Python 语言作为当前高校计算机基础教学的重要语言之一, 因其简洁性和广泛应用性, 已成为非计算机专业与计算机专业学生编程入门的首选。然而, 传统 Python 课程教学存在重语法、轻能力、实践不足、评价单一等问题, 难以满足新工科背景下对复合型人才的培养需求。本文以 OBE (Outcome-Based Education) 理念为指导, 重构课程目标与教学内容, 创新教学方法与考核机制, 构建“理论+实践+项目”三位一体的教学模式。经过两轮教学实践, 学生编程能力、项目完成质量与课程满意度显著提升, 为高校 Python 课程教学改革提供了可借鉴的经验。

关键字 Python, 程序设计, OBE, 教学改革, 项目驱动, 能力培养

Eaching Reform and Practice of the Python Programming Course Based on the OBE Concept

Wang Ruiping Zhu Qiang Zhou Xuezhong Xiong Ke

School of Computer Science & Technology, Beijing Jiaotong University
Beijing 100044, China

Abstract—Python has become the dominant first language in university computer-literacy courses because of its concise syntax and broad applicability for both majors and non-majors. Traditional offerings, however, over-emphasize grammar, neglect competency formation, provide insufficient practice, and rely on one-shot assessments, all of which fail to cultivate the interdisciplinary talent required by emerging-engineering education. Guided by Outcome-Based Education (OBE), we re-engineered the course outcomes and content, created innovative pedagogies and assessment schemes, and built an integrated “theory + practice + project” model. Two iterations of delivery show significant gains in students’ programming skills, project quality, and course satisfaction, offering a transferable framework for Python curriculum reform in higher education.

Keywords: Python; programming, OBE, teaching reform, project-based learning, competency cultivation

1 引言

随着人工智能、大数据、自动化等技术的快速发展, Python 语言因其语法简洁、库资源丰富、跨平台支持良好, 已成为高校计算机基础教学的重要语言之一。尤其在“新工科”建设背景下, Python 不仅是计算机专业的核心课程, 也逐渐成为非计算机专业(如金融、管理、生物、医学等)的编程入门语言。

然而, 当前高校 Python 课程教学普遍存在以下问题^[1,2]:

(1) 教学内容重语法、轻应用: 课程以语法讲解为主, 缺乏真实项目案例, 学生难以将知识应用于实际问题解决;

(2) 学生编程思维薄弱: 缺乏系统性训练, 导致学

生“听得懂、写不出”;

教学方法单一: 以教师讲授为主, 学生参与度低, 缺乏主动学习动力;

(3) 考核方式传统: 以笔试为主, 忽视过程性评价与实践能力考核, 难以全面反映学生真实水平。

为解决上述问题, 本文以 OBE 理念为指导, 开展 Python 语言程序设计课程的教学改革, 探索以学生能力培养为核心的教学新模式^[3,4,5]。

2 教学改革设计思路

“以学生为中心、以成果为导向”已成为新工科背景下课程建设的核心理念。然而, 理念落地需要具体的设计桥梁。在本轮 Python 语言程序设计课程改革中, 我们首先从“期望学生毕业时能做什么”反向倒推, 重新审视课程边界、教学策略与评价方式, 力求让每一节课、每一次作业、每一个项目都对标明确、

* **基金资助:** 本文得到北京交通大学本科教学建设与改革项目资助(项目编号: 356651535423)。

可测的能力成果。以下内容将系统阐述如何依托 OBE 框架完成课程目标的重构与教学内容的重塑,为后续教学方法与考核机制的创新奠定坚实基础^[5,6,7]。

2.1 OBE 理念指导下的课程目标重构

OBE (Outcome-Based Education) 是一种以学生学习成果为导向的教育理念,强调“以终为始”,即先明确学生应达到的能力目标,再反向设计教学内容与教学活动。该理念要求教师从“学生毕业时能做什么”出发,将宏观的人才培养目标逐级分解为课程目标、单元目标乃至课堂目标,确保每一教学环节都能对能力达成形成有效支撑。Python 语言作为一门兼具简洁性与实用性的编程入门课程,其教学目标不应局限于语法知识的传授,更应聚焦于编程思维、问题解决与团队协作等核心能力的养成。因此,结合 Python 课程实践性强、应用面广的特点,本文将课程目标细化为以下四个维度,以期实现知识、技能与素养的有机融合^[8,9,10]。

表 1 OBE 视角下 Python 课程能力目标与毕业要求对应关系

能力维度	具体目标
知识理解	掌握 Python 基本语法、控制结构、函数、面向对象编程等核心知识
编程能力	能独立编写中等复杂度的 Python 程序,具备调试与优化能力
问题解决	能运用 Python 解决实际问题,如数据处理、文件操作、可视化等
团队协作	能在小组项目中承担角色任务,具备代码规范与协作开发意识

2.2 教学内容重构:从“语法导向”到“任务导向”

传统教学往往以语法知识点为章节主线,按照“变量→条件语句→循环→函数→列表→字典→面向对象”的线性顺序展开。这种“知识罗列式”结构虽然逻辑清晰,却与学生认知规律和实际应用需求脱节:课堂案例碎片化,缺乏情境支撑;学生被动接受语法规则,难以形成解决真实问题的能力;章节之间缺少横向关联,导致“学得快、忘得更快”的现象屡见不鲜。为破解这一困境,改革后的课程以“任务驱动+项目贯穿”重构内容体系:每一模块都围绕一个真实可感的编程任务展开,任务内部隐含目标知识点,学生在“完成—反思—拓展”的循环中自然习得语法;五大模块按照认知负荷由浅入深、技能递进层层叠加,

最终汇聚成学期末的综合项目,实现知识、技能与素养的螺旋上升。表 2 给出了五大模块的任务主题、核心知识点及其与课程能力目标的映射关系。

表 2 任务驱动模块与知识能力对应表

模块名称	核心任务	涉及知识点
模块一: Python 基础	编写计算器程序	变量、输入输出、运算符、条件语句
模块二: 流程控制	实现猜数字游戏	循环、随机数、异常处理
模块三: 数据处理	分析 CSV 成绩文件	列表、字典、文件读写、pandas
模块四: 可视化	绘制学生成绩柱状图	matplotlib、数据清洗
模块五: 综合项目	开发“学生信息管理系统”	面向对象、模块封装、GUI 界面

3 教学方法创新

“教法”不变,“学法”难变。为了把 OBE “以学定教”的理念真正落到课堂,我们摒弃“教师讲—学生听”的单一范式,构建“课前自主建构—课内高阶操练—课后综合拓展”的三级梯度。围绕这一思路,课程重点打造两种互补式教法:一是翻转课堂,用短小精悍的微视频把知识传递前置,释放面授时间用于实战与思维碰撞;二是项目驱动(PBL),以学期大任务为牵引,将零散知识点编织成解决真实问题的完整能力链。二者首尾衔接、层层递进,共同支撑起“理论—实践—项目”一体化的教学新格局^[11,12,13]。

3.1 翻转课堂:课前学习+课内实战

传统课堂中,教师往往占据宝贵的面授时间进行知识点串讲,学生被动接收、当场理解有限,课后却又陷入“无人可问”的困境。翻转课堂通过“时空对调”破解这一瓶颈:知识传递前移至课前微视频与在线自测,课堂则变身成为高阶思维碰撞与编程实战的“演练场”。下面从课前奠基、课内跃迁、课后延伸三个环节,具体阐述本课程如何以碎片化学习撬动深度能力培养。

(1) 课前“微”学习

① 视频源:依托中国大学 MOOC 平台,遴选并自建 10 分钟以内微课,学生碎片化完成观看,每单元聚焦一个语法点或易错陷阱;

② 交互点：视频内嵌弹题（2~3道），答对方可继续；平台自动记录观看轨迹与正确率，作为平时成绩组成部分；

③ 预热任务：学生需在讨论区提交“一行代码疑问”或“生活场景应用设想”，教师课前归类，锁定课堂靶点。

（2）课中“深”实战

① 任务导向编程：教师发放“半成品”（如猜数字）代码，现场示范补全、调试与优化，学生同步跟做，即时提问即时解决。

② 随堂巩固实战：学生两人一组完成相似小游戏（如猜大小），下课前提交运行截图与源码，自动计入平时成绩。

（3）课后“延”拓展

① 课后即时巩固：在“希冀信息类专业课一体化平台（course.educg.net）”布置与课堂同难度作业，平台自动评测并生成完成度反馈，学生可即时查看得分与性能报告。

② 微视频反哺：教师依据课堂薄弱点每周录制3分钟“精华回顾”短视频上传群聊，支持2倍速播放，方便学生碎片化复习。

3.2 项目驱动教学（PBL）

学期初即布置小组项目任务：“基于Python的校园数据分析系统”。项目贯穿整个学期，分三阶段推进：

表 3 项目驱动三阶段任务与教学要点对应表

阶段	任务	指导重点
第1阶段（第3-5周）	数据采集与清洗	文件读写、异常处理、pandas基础
第2阶段（第6-10周）	数据分析与可视化	matplotlib、seaborn、数据分组
第3阶段（第11-15周）	系统封装与展示	面向对象、GUI设计、项目文档

3.3 混合式教学与AI辅助

当AI编码已成时代“标配”，课堂的使命就不再是“隔绝”，而是“嫁接”。让学生借助智能工具跑得更快，同时保持批判与创新的头脑。下列做法把CodeT5、CodeGeeX、腾讯云AI Code、阿里云Cosy等引入教学全流程，既发挥AI的生成优势，又通过对比、调试、反思确保学习深度，最终形成“AI助跑、

人工加速”的良性循环。引入AI编程助手（如CodeT5、CodeGeeX、腾讯云AI Code、阿里云Cosy）作为“编程伙伴”，鼓励学生：

（1）角色定位：将CodeT5、CodeGeeX、腾讯云AI Code、阿里云Cosy等工具定位为“1对1助教”，助教负责模板生成、语法补全与思路启发，而非直接给出最终答案。教师明确“三用三不用”原则：可用AI生成框架、补全库函数、提示异常原因；不可用AI替代核心算法设计、项目架构、业务逻辑分析。

（2）课堂嵌入流程

① 任务发布后立即“AI 5分钟”：学生用自然语言描述需求，CodeGeeX生成第一版代码；

② 人工调试15分钟：学生逐行阅读AI代码，加入断点、单步跟踪，记录“AI犯错点”（如越界、类型错误）；

③ 对比重构10分钟：保留AI优秀片段，重写冗余或低效部分，要求运行时间缩短20%以上；

④ 现场展示3分钟：投屏“AI版→人工版”side-by-side，用红色标注修改处，教师即时点评代码可读性、复杂度。

（3）AI使用报告（模板）

① 需求描述 & 提示词（Prompt）原文；

② AI生成代码首次通过率、错误列表与根因分析；

③ 人工改动行数占比、性能/可读性提升数据；

④ 学习反思：AI带给自己的最大帮助与最大陷阱；

⑤ 后续计划：哪些场景继续用AI，哪些场景坚持手写。报告字数300-500字，计入平时成绩10%。

（4）成效与反馈

两轮实践显示：

① 项目初版完成时间平均缩短28%，学生把更多精力投向架构设计与算法优化；

② 代码规范率提升35%，AI的PEP8自动提示弱化了“缩进、命名”等低级错误；

③ 85%学生认为“AI使用报告”促使自己真正读懂代码，而非复制粘贴。

4 考核机制改革

传统“期末笔试+平时作业”方式难以全面评价学生能力。改革后，采用“过程性评价为主、终结性评

价为辅”的多元考核体系:

表 4 多元考核体系与成绩构成比例

评价维度	权重	评价方式
平时实验	20%	每周实验报告+代码提交
项目过程	30%	阶段汇报+代码审查+组内互评
项目答辩	20%	现场演示+提问答辩
期末考试	20%	上机编程
学习反思	10%	期末提交《学习成长报告》 (含 AI 使用反思)

5 教学实践与成效分析

改革在北京交通大学纳米专业连续实施两轮: 2024 级 (56 人) 与 2025 级 (60 人) 构成实验组 ($n=116$), 2023 级未改革班级作为对照组 ($n=60$)。两轮教学方案、课时、任课教师保持一致, 以下所有数据均源自教务系统与课程问卷, 用于直观呈现改革前后差异。

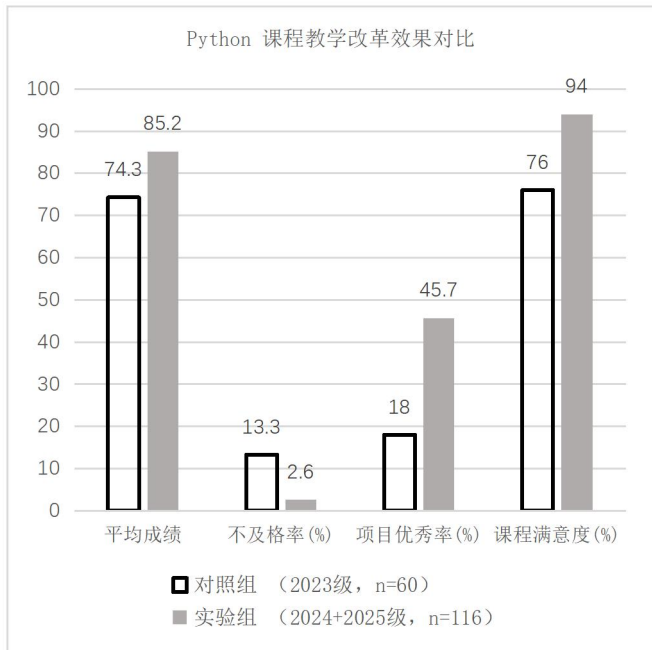


图 1 Python 课程教学改革效果对比

图 1 采用并列柱形方式直观呈现改革成效。图中每个指标均包含两根柱子: 左侧空心柱代表“对照组”(2023 级未改革班级, $n=60$), 右侧实心柱代表“实验组”(2024 + 2025 级改革班级, $n=116$)。从柱高差异可以看出:

(1) 平均成绩: 实心柱显著高于空心柱, 说明改革后学生整体分数提升约 11 分;

(2) 不及格率: 实心柱几乎贴近坐标轴, 表明改革将不及格人群压缩到不足原来的 1/5;

(3) 项目优秀率: 实心柱高度翻倍有余, 显示采用项目驱动与 AI 辅助后, 高质量作品比例大幅跃升;

(4): 课程满意度: 实心柱达到 94%, 对比空心柱的 76%, 反映出学生对翻转课堂、AI 编程伙伴等创新环节的高度认可。

总体而言, 实心柱在正向指标 (成绩、优秀率、满意度) 全面“压制”空心柱, 而在负向指标 (不及格率) 则明显“缩短”空心柱, 从视觉与数据双重角度验证了本次基于 OBE 理念的 Python 课程教学改革成效显著。

6 问题与反思

本课程是物工学院纳米材料与技术专业的定制版 Python 课, 该专业为北京交通大学与加拿大滑铁卢大学的中外合作办学项目。滑铁卢大学在联合培养方案中明确要求学生“具备用编程解决科学实验与工程数据的实践能力”, 因此课程必须突出动手与项目交付。然而, 纳米专业学生并非计算机类生源, 大一仅修过 2 学分“大学计算机基础”, 编程经验接近于零, 且每届学生来源地差异大, 导致同一课堂出现“零基础”与“有过竞赛经历”并存的两极现象。教学团队在两轮改革实践中暴露出以下突出问题:

(1) 基础差异放大, 项目难度难以“一刀切”: 同样一份“校园数据分析系统”, 基础好的小组 1 周即可进入可视化阶段, 而薄弱组 3 周仍卡在文件编码异常。虽然设置了分层任务 (必做/选做), 但学生自我评估不准, 常出现“选做”变“空做”, 影响整体交付节奏。

(2) 课程容量与指导人力不足: 课程要求期末项目必须提供可执行文件+技术文档+3 分钟演示视频, 指导工作较普通班级增加约 40%。目前助教仅 1 名, 导致代码审查深度不足, 部分学生把 AI 生成代码“贴过来、能跑就行”, 未能真正消化。

(3) AI 工具滥用与诚信风险: CodeGeeX 类工具生成的代码相似度高, 课堂虽已加入“AI 使用报告”与现场答辩, 但仍需更细颗粒度的代码溯源机制。

针对上述问题, 课程组将在下一轮实施中:

① 建立“分层项目库”, 把任务拆成基础、标准、进阶三档, 学生按摸底测试自动匹配;

② 引入代码相似度检测工具 Moss+AI, 结合“随

机代码走查”制度，降低抄袭风险；

③ 继续保留 AI 辅助环节，但增设“无 AI 手撕代码”期末现场考，保证学生离开工具仍能完成核心算法，满足“真实实践能力”的要求。

7 结束语

Python 语言程序设计课程作为高校计算机基础教学的重要组成部分，其教学质量直接影响学生后续专业课程学习与创新能力发展。本文基于 OBE 理念，系统推进课程内容、教学方法、考核机制的改革，构建了以能力培养为核心的教学新模式。实践表明，该模式显著提升了学生的编程能力、项目实践能力与学习满意度，具有良好的推广价值。未来，我们将进一步优化教学资源建设，推动 Python 课程与专业课程的深度联动，为新工科人才培养奠定坚实基础。

参考文献

- [1] 教育部高等学校计算机类专业教学指导委员会. 普通高等学校本科专业类教学质量国家标准(计算机类)[S]. 北京: 高等教育出版社, 2018.
- [2] 杨陟卓. 基于OBE的程序设计教学及计算思维能力培养. 计算机技术与教育学报, 2025年 V13 N2 PP169-173.
- [3] 关玉欣, 刘广文, 张斌斌, 邢红梅. 基于OBE的软件工程专业实验教学过程考核与评价. 计算机技术与教育学报, 2024年 V12 N3 PP71-75.
- [4] 朱雄泳, 张菲菲, 陈忆群, 等. 新工科《Python程序设计》课程改革探索. 电脑知识与技术, 2020年 V16 N24 PP131-133.
- [5] 韩万江, 张笑燕, 陈珑峥, 韩卓言, 韩韧. 基于软件工全流程的项目驱动式课程实验教学. 计算机技术与教育学报, 2023年 V11 N5 PP61-68.
- [6] 戴晓东, 韦小平, 李跃飞, 彭芳芳, 陈沅涛. 数智赋能下程序设计基础课程项目驱动智慧教学创新实践. 计算机技术与教育学报, 2025年 V13 N3 PP152-156.
- [7] 刘明剑, 张思佳, 李奇蔚, 葛泰亨. 依托“Python程序设计”课程的学生创新能力培养模式研究与实践. 计算机技术与教育学报, 2024年 V12 N4 PP40-44.
- [8] 赵叶纯, 王森, 洪李万益, 邱爽. 人工智能赋能Python程序设计课程教学改革研究. 信息与电脑, 2024年 N21. 中国期刊网 <https://kns.cnki.net>
- [9] 李晶, 李雯, 赵燕飞, 郭红建. 智能审计背景下“Python程序设计”课程教学改革探索研究. 工业和信息化教育, 2025年 N5 中国期刊网 <https://kns.cnki.net>
- [10] 单纯, 张润霖, 余渝, 王静. OBE框架下高校新工科“三维”信创新质人才培养模式. 计算机技术与教育学报, 2024年 V12 N5 PP28-36.
- [11] 郭曦, 王建勇. 生成式人工智能在Python教学中的作用与思考. 计算机技术与教育学报, 2024年 V12 N2 PP31-36.
- [12] 李志刚, 杨吉斌, 张睿, 王彩玲, 陈卫卫. 基于ChatGPT的程序设计翻转课堂教学方法实践. 计算机技术与教育学报, 2023年 V11 N2 PP125-129.
- [13] 黄厚财, 郑伟俊. 基于SPOC的翻转课堂教学设计与实践. 计算机技术与教育学报, 2021年 V9 N2 PP65-68