

AI 赋能下 Python 程序设计课程的教学改革与实践研究

林亮 伍琴兰 何璠

江西工程学院，江西南昌，338000；

摘要：随着人工智能技术发展，尤其是生成式 AI 普及，传统编程教育面临挑战与机遇，Python 教学改革迫切。本文先分析高校《Python 程序设计》课程教学现存问题，接着探讨 AI 技术对教学理念、模式、评价体系的改革与重构方法。在此基础上，提出“AI 赋能”教学改革框架，包含教学目标重塑、内容重构、模式创新、评价体系改革四个维度，并辅以教学实践案例说明。最后，辩证分析 AI 赋能教学可能带来的学术诚信、技术依赖等挑战并提出应对策略。本研究旨在为编程类课程教学创新提供理论参考与实践路径。

关键词：AI 赋能；Python 程序设计；教学改革

DOI：10.64216/3104-9702.25.02.030

引言

Python 语言因语法简洁、生态库强大，且在前沿领域处于核心地位，成为国内外高校广泛开设的程序设计入门语言。传统《Python 程序设计》课程教学采用“教师讲授 - 学生模仿 - 上机练习”模式，虽体系完整，但有难以因材施教、学生实践创新能力培养不足等弊端。同时，AI 编程助手正改变程序编写范式，开发者从亲手编写代码转变为提出意图、审查优化 AI 生成代码。这迫使编程教育反思重构，如思考教学目标、教师角色转变、如何利用 AI 工具等问题。本研究旨在探索将 AI 技术从“挑战者”变为“赋能者”的教学改革之路，通过将 AI 融入 Python 课程教学全流程，重塑教学价值，创新教学方法，构建以培养学生计算思维和创新能力的新型教学模式。

1 传统 Python 课程教学现状与困境分析

教学目标滞后，与时代需求脱节：传统教学多将重点放在 Python 语法、数据结构、流程控制等基础知识的传授上，考核目标往往是学生能否写出“正确”的代码。然而，在 AI 时代，企业对人才的需求已从“编码实现”转向“问题定义”“算法选择”“系统架构”和“AI 工具协同”等高阶能力。教学模式单一，学生主动性差：“满堂灌”的理论教学配合验证性的实验，难以激发学生的学习兴趣。学生处于被动接受状态，缺乏主动探索和解决复杂问题的机会，批判性思维和创新能力的培养不足。实践环节薄弱，与现实应用断层：课程实验项目多为设计好的、孤立的数学题或小游戏，缺乏与真实世界场景（如数据分析、Web 开发、AI 模型调用）的连接。学生“知其然不知其所以然”，不知道所学知识有何用处，学习动力不足。评价体系片面，难以衡量真

实能力：传统评价多以期末笔试+机考为主，侧重于考察语法细节和短代码片的编写。这种“一考定乾坤”的方式无法有效评估学生的过程努力、代码质量、调试能力、项目协作能力和创新思维。师资负担过重，个性化指导缺失：一名教师面对数十甚至上百名学生，很难对每个人的学习进度、遇到的困难进行个性化关注和指导。学生的问题无法得到及时解答，容易积累挫败感，导致“从入门到放弃”。

2 AI 赋能 Python 教学的核心价值与赋能路径

AI 技术并非要取代教师，而是作为强大的“辅助工具”（Copilot）和“能力增强工具”（Toolset），从以下几个方面为教学赋能：赋能学生：从“被动接收”到“主动构建”永不疲倦的个性化导师：AI 编程助手可以 7x24 小时响应学生问题，无论是语法错误调试、代码逻辑解释，还是编程思路启发，都能提供即时、耐心的反馈，极大降低了学习门槛和焦虑感。项目探索的“加速器”：学生可以借助 AI 快速搭建项目框架、生成基础代码模块，从而将节省下来的时间精力专注于更核心的算法设计、性能优化和功能创新上，敢于尝试更复杂、更有趣的项目。培养代码审查与批判性思维：学生需要学会如何向 AI 提问（Prompt Engineering），如何审阅、测试和优化 AI 生成的代码。这个过程本身就是对逻辑思维、批判性思维和工程质量意识的极好锻炼。赋能教师：从“知识传授者”到“学习设计者”解放重复性劳动：AI 可以自动生成练习题、作业答案，甚至试卷，协助批改格式固定的作业，使教师从繁重的重复劳动中解脱出来。实现精准教学：通过分析 AI 互动平台产生的学习数据（如常见错误、提问频率、知识点停留时间），教师可以精准把握班级整体的学习难点和个体差异，从

而进行有针对性的课堂讲解和个性化干预。设计高阶学习任务：教师可以将教学重点从“怎么写”转向“写什么”和“为什么这么写”，设计更多开放式、探究式的项目任务，引导学生利用 AI 工具解决复杂的、跨学科的实际问题。

3 AI 赋能下 Python 课程教学改革的具体实践路径

3.1 教学目标重塑：从“编码员”到“架构师与质检员”

新目标应聚焦于以下维度：概念化能力：将复杂问题分解为计算机可执行步骤的能力。

AI 工具协同能力：精通 PromptEngineering，能高效利用 AI 助手生成、解释、调试和优化代码。

批判性思维与审查能力：具备对 AI 生成代码的测试、验证、评估和重构的能力。

系统集成与创新能力：能将多个 AI 生成的模块或外部 API 集成到一个完整、可运行的系统中，并实现创新应用。

3.2 教学内容重构：分层与融合

将课程内容分为三层：核心基础层（必须精通）：编程核心概念（变量、循环、函数、类）、基本数据结构、算法思维、程序调试与错误处理。此部分需精讲多练，即使有 AI，也必须亲手掌握。

AI 协同层（重点教授）：如何高效地向 AI 提问（Prompt 工程学）、代码审查技术与规范、单元测试、版本控制（Git）。将此部分作为重要教学模块嵌入课程。

项目实践层（主题化）：设计以问题为导向的项目模块，如：“网络爬虫与数据分析”、“机器学习模型调用与部署”、“智能聊天机器人开发”、“自动化办公应用”等。学生需在项目中强制使用 AI 工具并报告使用过程。

3.3 教学模式创新：“双师”课堂与翻转模式

构建“教师+AI”的“双师”课堂教学模式。课前：学生通过观看微课视频学习基础概念，并利用 AI 完成简单的预习任务（如：让 AI 生成一个函数，然后自己尝试理解和修改）。

课中：教师不再讲授基础语法，而是展示如何用 AI 解决一个复杂问题，演示 Prompt 技巧和代码审查流程。课堂时间主要用于项目研讨、小组协作和深度答疑。

课后：学生以小组形式完成项目，要求记录使用 AI 的对话日志，并阐述 AI 在项目中的作用、自身贡献以及对 AI 生成代码的改进过程。教师在此过程中提

供方向性指导和关键节点评审。

3.4 评价体系改革：过程化、透明化、多元化

在考核方面，本课程建立“无标准答案”的综合性评价体系。取消传统笔试：代之以多次迭代的项目大作业。过程评价（50%）：AI 使用日志：评估其 Prompt 质量、与 AI 交互的深度、问题解决的逻辑。代码审查报告：对自身或他人代码（含 AI 生成）的优缺点分析、优化建议。同伴互评：在项目中的贡献度与协作能力。成果评价（50%）：项目创新性与完成度。期末项目答辩：演示项目并回答提问，重点考察其对代码的理解而非记忆。诚信协议：要求学生签署学术诚信协议，明确合理使用与抄袭的界限，要求对所有 AI 生成的代码进行注释说明。

4 教学实践案例：“基于 AI 助手的智能天气预报分析系统”项目

项目目标：小组合作开发一个程序，能够获取指定城市的天气预报数据，并进行深入的数据分析，例如绘制最高温度趋势图、统计各种天气现象的频次等，最终生成一份详尽且直观的可视化报告。通过这一过程，学生不仅能够掌握编程技能，还能提升数据分析和可视化能力。

AI 赋能点：

项目启动：在项目初期，利用 AI 工具（例如 ChatGPT）进行头脑风暴，生成详细的项目计划和技术选型建议。具体包括选择合适的编程库（如 Requests 库用于 API 请求、Pandas 库用于数据处理、Matplotlib 库用于绘图等），以及如何设计高效且模块化的函数结构。

代码生成：在编码阶段，借助 AI 工具（如 Copilot）自动生成关键代码片段。例如，生成用于请求天气预报 API 的代码、使用 Pandas 进行数据清洗和处理的代码，以及使用 Matplotlib 绘制各类图表的代码。这样可以大幅提高编码效率，减少手动编写代码的时间。

调试优化：在调试过程中，遇到难以解决的 Bug 时，可以将错误信息粘贴给 AI，获取针对性的调试建议和解决方案。此外，还可以请求 AI 对现有代码进行注释，以提高代码的可读性，或对代码进行优化，提升程序的性能和运行效率。

文档撰写：在项目文档撰写阶段，利用 AI 协助生成项目文档的初稿，包括项目背景、目标、技术选型、实现过程、结果展示等内容。AI 生成的初稿可以为后续的文档完善提供良好的基础。

教师角色：在整个项目过程中，教师扮演多重角色。首先，提供 API 密钥申请的指导，帮助学生顺利获取所

需的天气预报数据。其次,讲解数据结构和 API 响应格式,确保学生能够正确理解 and 处理数据。此外,评审各小组的 Prompt 质量和代码质量,确保项目的顺利进行。最后,主持项目答辩,评估学生的项目成果和综合能力。

考核重点:项目的考核重点并不在于功能的复杂性和多样性,而是关注小组如何高效利用 AI 工具推进项目进展,对 AI 生成代码的理解程度和改造深度,以及最终报告中所体现的数据洞察力和分析能力。通过这种方式,培养学生的创新思维 and 实际解决问题的能力。

5 面临的挑战与对策

学术诚信挑战:在当前教育环境中,学生可能会直接利用 AI 工具来完成作业,从而放弃自主思考和深入探究的过程。这种做法不仅违背了学术诚信的基本原则,还可能导致学生缺乏必要的思维训练和知识内化。对策:为了应对这一挑战,教育机构需要从根本上改变现有的评价体系,例如,可以采用更加注重过程记录和答辩考核的方式,确保学生真正参与到学习过程中。此外,教师应设计一些无法直接由 AI 工具解决的开源性问题,或者需要结合真实数据和个人洞察力才能完成的项目,从而迫使学生进行独立思考和创造性工作。同时,加强相关的教育引导,帮助学生树立正确的 AI 使用观念,将 AI 视为辅助学习的“实习生”而非替代思考的“枪手”。

技术依赖风险:过度依赖 AI 工具可能导致学生在基础知识掌握上不够扎实,一旦脱离 AI 辅助,便无法独立进行编程和解决问题。对策:为了防止这种现象的发生,教育机构应坚持要求学生在“核心基础层”必须手写代码,并通过闭卷笔试的方式考核核心概念的理解和应用。此外,在课程初期可以设置“无 AI 周”,强制学生在此期间独立完成编程任务,从而锻炼他们的自主编程能力。

教师数字鸿沟:面对新兴技术的快速发展,部分教师可能因缺乏相关知识和技能而产生畏难情绪,难以有效利用 AI 工具进行教学。对策:针对这一问题,学校可以组织校级层面的培训和工作坊,邀请专家分享成功案例和实用技巧,帮助教师提升技术应用能力。同时,建立教学共同体,鼓励教师之间相互合作、共同探索 AI 在教学中的应用,形成良好的互助氛围。此外,调整教学激励政策,对在教学创新方面取得显著成果的教师给予认可和奖励,进一步激发他们的积极性和创造力。

工具与成本问题:大型 AI 工具的使用往往需要支付高昂的费用,这对于一些学生和学校来说可能存在经

济负担和访问障碍。对策:为了解决这一问题,学校可以考虑进行集体采购,以降低单个用户的使用成本。同时,优先选择和推荐优秀的开源 AI 模型,或者提供校内的算力支持,确保学生能够便捷地获取和使用 AI 工具。此外,引导学生了解和掌握多种不同的 AI 工具,培养他们的工具选择和使用能力,以便在不同情境下灵活应对。

6 结论与展望

AI 的浪潮不可逆转,编程教育与其被动防御,不如主动拥抱。将 AI 深度赋能到《Python 程序设计》课程中,是一场必然的教学范式革命。它不仅仅是在现有体系中增加一个新工具,更是对教学目标、内容、方法和评价的全方位重构。改革的核心,是将教学重心从低层次的“语法记忆和代码实现”提升至高层次的“问题解决、创新思维 and 与智能工具协同的能力”。教师角色从知识的权威传授者转变为学习过程的引导者、设计者和激励者。学生则从被动的接收者转变为积极自主的探索者,通过与 AI 的持续互动,更早地体验到软件开发的真实工作流。随着多模态 AI 的发展,编程教学甚至可以进一步与自然语言理解、图像识别等技术结合,创造出更沉浸、更直观的学习体验(例如,用草图生成代码)。本研究提出的框架仅是一个起点,更多的实践细节、效果评估和模式优化,仍需广大教育工作者在持续的实践中共同探索与完善,最终培养出能驾驭 AI 而非被 AI 取代的新一代创新人才。

参考文献

- [1] 赵叶纯,王森洪,李万益,等.人工智能赋能 Python 程序设计课程教学改革研究[J].信息与电脑,2024,36(21):1-6.
- [2] 金枝,詹丹丹,王萍,等.ChatGPT 赋能下"Python 程序设计"课程改革探索[J].电气电子教学学报,2025(3).DOI:10.3969/j.issn.1008-0686.2025.03.004.
- [3] 刘姝.Python 数据分析通识课程教学研究与实践[J].2021.

作者简介:林亮(1996-),男,江西省新余市人,本科毕业,江西工程学院人工智能与数据科学学院教师,南昌大学软件工程专业学士,研究方向为计算机视觉。

基金项目:江西工程学院科学技术研究课题:基于骨架动作识别的中心差分转换器图卷积(2024-JGKJ-24).