Advances in Education 教育进展, 2026, 16(11), 1270-1278
Published Online November 2026 in Hans. https://www.hanspub.org/journal/ae
https://doi.org/10.12677/ae.2026.16112500

Hans汉斯

生成式人工智能时代下计算机语言类课程教学改革与教师角色升级

王晓梅，李　刚

西安邮电大学计算机学院，陕西　西安

收稿日期：2026年10月9日；录用日期：2026年11月9日；发布日期：2026年11月17日

摘　要

随着生成式人工智能的快速发展，企业对人才能力提升的需求，以及高校教育个性化的需求愈发强烈，针对计算机语言类课程实践性强、逻辑思维和抽象思维要求高、技术更新快的特点，本文提出了“夯实基础知识 + 提升生成式人工智能复杂编程能力”的计算机语言类课程的改革思路，以基础题目为基础，生成式人工智能工具解决复杂题目为进阶，提高学生解决实际问题的能力。鼓励教师进行角色升级，从知识传授者转向生成式人工智能工具引导者，引导学生正确高效地使用AIGC工具。

关键词

生成式人工智能，计算机语言类课程，教学改革

Teaching Reform of Computer Language Courses and Teachers’ Role Upgrading in the AIGC Era

Xiaomei Wang, Gang Li

School of Computer Science & Technology, Xi’an University of Posts and Telecommunications, Xi’an Shaanxi

Received: October 9, 2026; accepted: November 9, 2026; published: November 17, 2026

Abstract

With the rapid development of AIGC, the demands of enterprises for talent competence improvement and the personalized needs of higher education in universities have become increasingly prominent. Targeting the characteristics of computer language courses, which feature strong practicality, high requirements for logical and abstract thinking, and rapid technology iteration, this paper proposes

文章引用: 王晓梅, 李刚. 生成式人工智能时代下计算机语言类课程教学改革与教师角色升级[J]. 教育进展, 2026, 16(11): 1270-1278. DOI: 10.12677/ae.2026.16112500

a reform idea for computer language courses: "consolidating basic knowledge + enhancing AIGC-enabled complex programming capabilities". Taking basic exercises as the foundation and solving complex problems with AIGC tools as the advanced stage, this reform aims to improve students' ability to solve practical problems. It also encourages teachers to upgrade their roles, transforming from knowledge transmitters to AIGC tool guides, so as to instruct students to use AIGC tools correctly and efficiently.

Keywords

AIGC, Computer Language Courses, Teaching Reform

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 2026 年 4 月，教育部等五部门联合印发了《"人工智能+教育"行动计划》的通知，通知中提出推动人工智能相关内容融入高校公共基础课，整合高水平研究型大学、科技领军企业、国家实验室等多方力量，探索人工智能拔尖创新人才培养新模式[1]。当前，国内很多高校已经在多个课程体系中开展"人工智能+"的课程改革实践，例如"人工智能+"课程改革在大学计算机基础课程中进行了尝试[2]，边小勇等提出面向 AI 能力培养"师－生－机"三元结构的计算机语言类课程教学体系[3]。同时针对 AIGC (生成式人工智能)技术的快速发展，很多教师尝试将AIGC技术与传统教学进行结合，比如数据挖掘课程[4]、计算机系统类课程[5]、计算机系统基础课程群[6]，都取得了较好的效果。

AIGC 技术的快速发展给计算机语言类课程带来了前所未有的优势，但每件事物的出现都具有两面性。当学生的基础知识和专业思维尚未建立时，AIGC 快速生成答案降低了学生进一步深入学习的兴趣；另外，AIGC 生成的内容不一定完全准确，更需要学生拥有扎实深厚的知识积累，才能够鉴别出生成内容的真伪。因此传统课程在 AIGC 技术的广泛应用下，需要对课程进行改革，改变学生和教师的传统思维，转换教师角色，使计算机语言类课程更快顺应时代发展。

2. AIGC 时代下计算机语言类课程改革的必要性

2.1. AIGC 技术发展促使计算机语言类课程改革

传统的计算机语言类课程专注于基本的语法规则和算法的逻辑框架，学生通过学习基础知识和上机实践操作，一步步巩固知识学习。然而，随着 AIGC 的崛起，学生甚至是普通人都可以通过自然语言描述快速获得程序代码，通过和 AIGC 大模型的进一步对话，还可以对程序进行优化。在学习初始阶段，学生辛辛苦苦编写调试出的程序，既没有大模型编写的速度快，也没有大模型编写的质量好，这会对学生编程学习的成就感造成较大冲击，如果不能够正确引导，学生将很快失去对编程的兴趣。如果计算机语言类课程还是坚持传统的教学方法，而忽视 AIGC 辅助下的高效程序开发方式，则学生很难适应未来行业的发展。因此，计算机语言类课程进行范式转型，淘汰过时内容，培养学生 AIGC 协同开发能力，是当前亟待解决的现实问题。

2.2. 能力需求的提升促使教学目标转型

随着 AIGC 的发展，企业对人才的能力要求越来越高，只拥有单一编码能力很难在众多的竞争者中

脱颖而出。现在在求职市场上，越来越要求求职者具有复合型技术能力。特别是如下能力更受企业青睐，首先是 AIGC 工具的使用能力，AIGC 可以使编码效率大幅提高，使用提示词生成符合需求的代码，并能修正其中的逻辑错误；其次，要有批判性思维，AIGC 生成的内容并不一定完全正确，也可能有错误或漏洞，需要具备识别出其中错误、并能优化代码的能力；最后，需要具有创新的设计能力，探索传统编程模式下开发效率较低的复杂算法或系统。因此，这些能力要求促使计算机语言类课程教学目标进行转型，从传统的知识灌输转为能力提升训练，通过引入 AIGC 辅助编程或 AIGC 辅助开发项目，帮助学生构建知识主导与 AIGC 赋能的开发范式，引导学生提高编程开发能力。

2.3. 教育个性化需求驱动教学模式革新

学生之间差距较大。有些学生在初中或者高中就已经接触到了编程，甚至参加过全国的编程大赛，而有些学生是第一次接触计算机语言的学习。不同高校可向学生提供的教学资源存在显著差异，头部高校凭借自身合作优势更易整合外部优质资源，比如企业级的开发环境或真实的项目案例，而大部分普通高校提供给学生的资源相对受限。而 AIGC 的发展为普通高校获取计算机语言类的资源提供了可能。例如，以 AIGC 为驱动的智能教学平台，可以模拟复杂的开发环境，针对不同学生编写的代码，快速进行错误语法反馈，为完整的程序代码提供性能优化的建议，甚至能够给出行业内最新实现技术的相关思路或参考方案。因此，AIGC 为计算机语言类课程的个性化培养提供了机会。针对不同学生定制个性化的学习路径，促使教学模式进行革新，教师从知识传授者转变为学习引导者，改变自己的角色定位。

3. AIGC 时代下计算机语言类课程改革的内容设计

3.1. 计算机语言类课程的特点

以 C 语言、Java、Python 为代表的程序设计类课程，是计算机专业的核心基础课，其中 C 语言课程更是理工类专业的公共基础课，是多数理工科学生的第一门编程语言课。其普遍具备以下几方面典型特征。

3.1.1. 实践性强，动手能力更重要

许多学生表示，虽然上课的内容听懂了，可是真正面对题目编写代码的时候就会频频出错。像指针、递归、面向对象这些概念，学生仅靠上课听讲，记忆语法规则远远不够，只有通过大量的上机编程练习才能够真正掌握其中的含义。所以，计算机语言类课程和传统的理论课程有很大的区别，其本质上更像一门技能型课程，要求学生除了掌握基础理论外，还需要大量的实践操作提升编程能力。相关教学研究表明要学好这类课程，课后投入的编程时间应是课堂时间的数倍，才能得到预期的学习成效。

3.1.2. 逻辑思维和抽象思维要求高

计算机语言类课程训练的是一种逻辑思维和抽象思维能力。以 C 语言为例，如图 1 所示，展示的是《C 语言程序设计》课程的经典知识体系，涵盖选择结构、循环结构、数组、函数、指针、结构体共用体、文件及编译系统等内容。程序设计的本质是用计算机理解的方式去描述并解决问题。这要求学生将现实问题抽象成数学模型，再将模型转化为算法，最终将算法落实为代码，每一步出现逻辑错误都会导致程序运行失败，现实问题无法解决。许多编程语言涉及高度抽象的概念，比如封装、继承、内存管理、并发控制等，学生需要在脑海中构建代码在计算机中如何一步步地执行过程，这种逻辑思维和抽象思维的培养是计算机语言类课程的典型特征。

3.1.3. 技术更新快，需具备自学能力

很多人都深切地感受到技术的更迭非常快速。编程语言在不断的演进，特别是一些框架或工具链使

用几年后就被新技术替换。这意味着学生在学校所学的具体技术可能在毕业进入工作岗位时就已经过时。因此，计算机语言类课程除了教授语言中具体的语法规则，更应该培养学生学习任何一门新语言的方法论和迁移能力，例如阅读官方文档、查阅技术社区、解决新技术中的错误，使学生在面对新技术时，能够快速适应并持续学习。

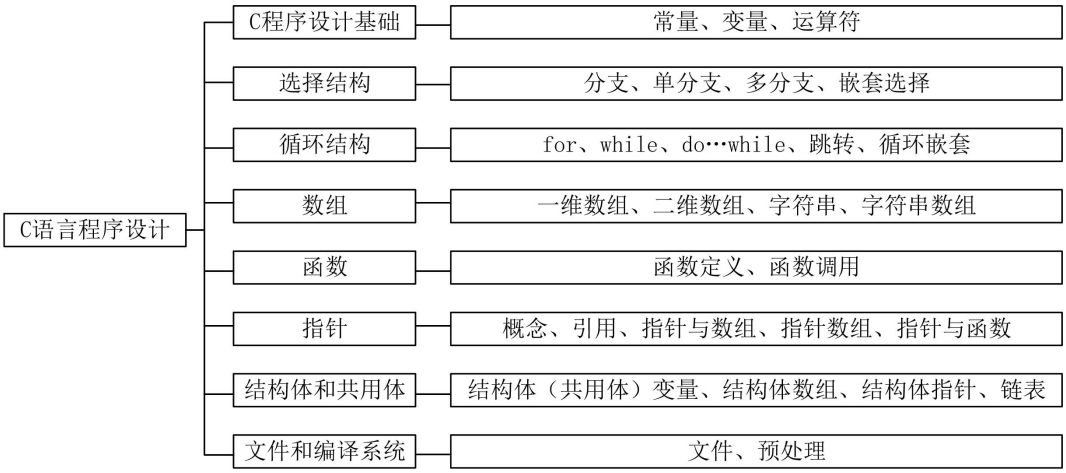


Figure 1. Typical knowledge structure of the C language programming course
图 1. C 语言程序设计课程典型知识结构

3.2. 教学改革的理论基础与 AIGC 的认知影响分析

课程改革不应该只讨论 AIGC 的工具具体怎么用，而应该探讨更深一层的问题。AIGC 进入编程课堂之后，学生的认知过程发生了哪些变化，课程的设计又应当依据什么来调整。本文以建构主义学习理论和认知负荷理论为学理基础，首先分析 AIGC 对学生认知过程的影响，再根据这些基础理论进行后面的分层设计与风险防控。

建构主义学习理论认为，知识并不是被动接收的结果，而是学习者在原有经验基础上主动建构起来的[7]。编程学习对这一点体现得尤其明显，学生盯着报错信息逐行排查、改完再编译、编译再报错，如此反复，才能真正把语法规则和算法思想内化为自己解决问题的经验。相比之下，直接从 AIGC 那里拿到能运行的代码虽然省事，却没有经历知识建构最关键的环节。因此，在基础阶段，AIGC 只能作为认知工具和学习支架辅助学生，不能替代学生完成意义建构。

认知负荷理论指出，人的工作记忆容量是有限的，学习效果取决于内在认知负荷、外在认知负荷与相关认知负荷之间的合理配置。对刚接触编程的学生来说，查语法、配置开发环境、对着报错信息找原因，是占用工作记忆最多、也最消耗耐心的环节；AIGC 把这些外在负荷接过去以后，学生才能把有限的注意力集中到问题分解、算法设计这些真正有助于图式建构的任务上。因此，对于计算机语言类的课程改革应该采用分层设计：基础阶段由学生独立编写代码，AIGC 辅助学习完成调试排错、代码解释等，完成图式建构；进阶阶段再借助 AIGC 强大功能，把认知资源集中在复杂任务中的高阶认知活动中。

3.3. AIGC 时代下计算机语言类课程的改革思路

现在网络上有两种观点较为流行。一种是针对学校教育，不管是写报告还是编代码，老师都要求学生自己分析要求，独立完成题目；而工作之后，企业要求员工使用 AIGC 工具提高生产效率，不会使用工具的编程人员面临着淘汰。因此，学校教育经常被人诟病，认为直接学习 AIGC 工具，不需要再进行

基础知识和系统知识的学习。然而，实际的情况是，不学基础知识，AIGC 工具给你的答案你也看不懂；不学 AIGC 工具，你的效率达不到企业要求。因此，本文提出 AIGC 时代下计算机语言类课程的改革思路是“夯实基础知识 + 提升 AIGC 复杂编程能力”。

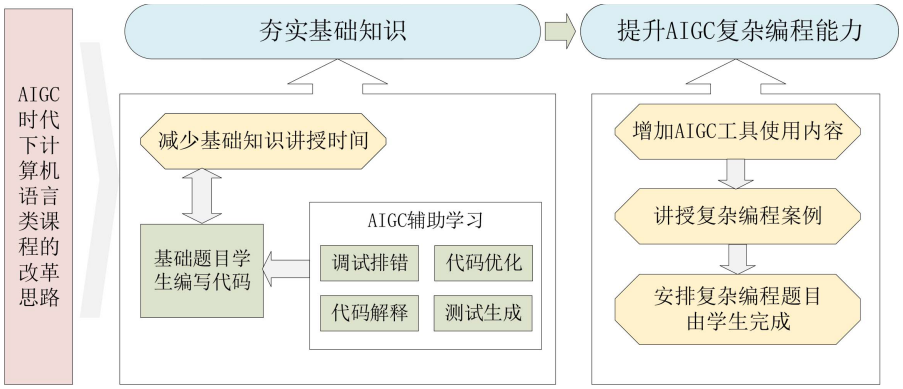


Figure 2. Module diagram of reform ideas for computer language courses in the AIGC era
图 2. AIGC 时代下计算机语言类课程的改革思路模块图

如图 2 所示，模块图包含两大部分，一部分是“夯实基础知识”内容，其中包括减少基础知识讲授时间，针对基础题目由学生独立编写代码，另外 AIGC 作为辅助工具，在学生编写代码时帮助其进行调试排错、代码解释、代码优化和测试生成，使学生快速准确地掌握各个基础知识点；另一部分是“提升 AIGC 复杂编程能力”内容，将减少的讲授基础知识的课时用于新内容，包括增加 AIGC 工具的使用、讲授复杂编程案例，并安排复杂编程题目，由学生依托 AIGC 工具辅助完成，提高学生解决复杂问题的能力。

3.4. 夯实基础知识，AIGC 工具进行辅助

3.4.1. 减少基础知识讲授时间

在传统的计算机语言类课程的教学中，基础知识的讲解一般占总学时一半以上，教师会针对语法规则、数据结构进行解释，根据典型题目讲解编程思路。这种方式的优点是学生理解题目比较透彻，能较快搭建编程思路。但是这种教学方法也有弊端，最明显的现象是有些学生不敢下手编写程序，对待编程有畏难情绪。计算机语言类课程是重实践的课程，只有将所学知识转化为亲自编写的代码，才能对基础的知识有更深刻的理解。因此，在课程的设计中，将基础知识讲解部分进行压缩，压缩至原来的三分之二，比如原先 40 学时的讲课学时压缩至 26 学时，余下的 14 学时留给第二部分内容的讲授。

3.4.2. 基础题目学生编写代码，AIGC 工具进行辅助学习

学生对于基础题目的训练是必不可少的，编程不仅仅是堆砌语法，更重要的是构建逻辑思维，形成解决实际问题的能力。学生编写代码的过程，本质上是将抽象思维转换为计算机具体操作指令的过程。当代码运行报错时，学生需要订正语法错误和逻辑漏洞，正是这种试错到修正的过程，成为将编程能力从会写上升为写好的关键步骤。如果直接依赖 AIGC 生成代码，学生则会失去深度思考的机会，当面对复杂题目时，则更无从下手了。

虽然基础题目要求学生自己编写代码，但 AIGC 工具的价值也不能全部否定，它可以作为学习伙伴辅助学生日常学习，使学生尽快得到反馈，加快学习进度，增强学习过程中的互动性。常用的辅助学习方法包括调试排错、代码解释、代码优化和测试生成等。

调试排错是编程学习中较为耗时的环节，学生面对问题时，一般需要逐行调试检查代码，或者是根据错误信息上网寻求解决方案，效率较低。而 AIGC 工具可以快速识别错误类型并给出分步调试指南，缩短了查找时间。

代码解释可以对抽象的代码进行语义化解析，使学生快速梳理代码逻辑，搭建抽象概念和具体实现之间的对应关系。

代码优化是提升编码技能的重要部分。学生在初学阶段编写的代码大部分都是以能够运行为目标，要做到更加专业和优秀，就需要对代码进行优化，AIGC 工具可以在性能优化、可读性提升、安全性增强等方面给学生提供建议。

测试生成是确保编程代码完整的关键环节。学生编写代码时往往觉得程序能运行就可以了，很少关心测试这个部分。然而，测试是确保代码正确性的关键部分，AIGC 工具可以生成测试用例，涵盖边界条件、异常情况等，这些测试用例帮助学生快速验证代码质量，降低出现差错的风险。

3.5. 增加 AIGC 工具使用内容，提升 AIGC 复杂编程能力

3.5.1. 课程增加 AIGC 工具使用部分

基础知识讲授学时压缩后，多出来的学时用于教授 AIGC 工具如何使用的内容。AIGC 工具的使用能力已经被很多企业作为招聘员工的重要指标。学生仅仅掌握编程语言很难提高编程效率，只有能够熟练运用 AIGC 工具，才能适应真实工作场景。

这部分内容的学习，主要包括如何写出高效提示词，而对于有些计算机语言编程课程，如针对 Python、Java 语言，还可以添加更多教学内容，例如如何在集成开发环境中添加插件、插件如何使用等。

在提示词教学方面，应使用结构化的编写方法，如指定特定身份以约束输出方向，将复杂需求拆分为多轮对话，提供示例让 AIGC 模仿输出格式，每种方法都结合案例进行讲解，让学生逐步掌握高效提示词的用法。

3.5.2. 教授复杂编程案例

在教授 AIGC 工具使用之后，需要在课程中进行复杂编程案例的讲解，使学生能够更直观地理解复杂编程的过程。考虑到 C 语言、Python、Java 等不同编程语言在知识体系与产业应用场景上的差异，本文分别针对三类课程设计了差异化的复杂编程案例。具体设计如下：

(1) C 语言课程案例：图书管理系统。C 语言的案例选用图书管理系统。系统要求的图书属性包括书号、书名、作者、ISBN 号、类型、价格和库存，先实现增加、删除、修改三项基本功能；第二步增加统计模块，按类型统计藏书，并把库存为 0 的图书单独列出来；第三步把图书信息写入文件 book.txt，程序下次运行时直接读取；最后加上登录功能，登录信息保存在文件 login.txt 中。学生拿到 AIGC 生成的分步代码后，需要自行完成编译调试，直到功能符合预期。案例的训练点集中在结构体、链表、文件操作与模块化设计，正好承接基础阶段的 C 语言内容，也能训练学生借助 AIGC 做需求拆解与分步实现，作为复杂编程案例的入门项目比较合适。

(2) Python 课程案例：数据 API 调用与可视化分析。这个案例对应数据分析岗位的日常工作，拿到一个公开数据源，经过请求、解析、存储，最后画成图。本题目要求学生根据天气生成可视化数据，首先通过接口获取天气数据，把返回的 JSON 解析后存成 CSV 文件；用 matplotlib 把气温随时间的变化画成折线图，补上图例和坐标轴标注；最后加上参数化查询与异常处理，应对网络超时、数据缺失等情况。整个案例把网络请求、JSON 解析、文件读写、数据可视化与异常处理串成一条线，学生在调试与扩展中完成数据获取、清洗、呈现的完整流程。

(3) Java 课程案例：服务模块开发。Java 案例的着眼点是企业级开发的工程化流程。提示词按开发顺序给出：先用 Spring Boot 搭建订单管理服务模块，提供增删改查的 RESTful 接口；再用 Spring Data JPA 完成与 MySQL 的持久化映射并配置数据源；随后补充统一异常处理、参数校验与日志记录；最后编写单元测试，启动服务后用 Postman 验证接口。学生在 AIGC 辅助下完成模块搭建、配置排错与功能验证，经历从接口设计到测试交付的完整过程，对服务架构下的模块化协作和工程化开发会有更具体的认识。

三个案例对 AIGC 的依赖程度是不同的。C 语言阶段学生最需要理解语法、指针和内存管理等基础内容，AIGC 只用于代码生成、调试排错与代码解释；到了 Python 课程，学生已有一定基础，可以让 AIGC 解释 API 用法，自己把精力放在理解数据流与处理异常上；Java 案例则将模块生成、配置排错与测试辅助都交给 AIGC 处理，因为工程化开发讨论的本来就是规范、框架与协作。三种案例的设计对照汇总见表 1。

Table 1. Comparison of AIGC teaching case design for different programming languages
表 1. 不同编程语言 AIGC 教学案例设计对照

编程语言	代表案例	核心知识单元	AIGC 介入环节
C 语言	图书管理系统	结构体、链表、文件操作与模块化设计	调试排错、代码解释
Python	数据 API 调用与数据可视化	网络请求、JSON 解析、数据可视化、异常处理	API 用法解释
Java	服务模块开发	Spring Boot、RESTful 接口、对象关系映射、单元测试	模块生成、配置排错、测试辅助

3.5.3. 安排复杂题目，由学生完成

下面将安排复杂题目，由学生利用综合所学的知识和 AIGC 工具辅助完成。这里安排的题目，应当贴近实际业务场景、从单一功能到多功能且模块协作逐层推进，学生能够完整走完需求分析、代码组织和调试排错的一整套流程。

同时，要求学生可以使用 AIGC 工具，但不能过于依赖工具。学生可以依靠工具辅助生成代码、排查错误、优化逻辑，但是最终提交的代码必须是自己理解并掌握的。因此，教师在进行题目验收时，应当让学生指出哪些内容是 AIGC 生成的，哪些内容是由自己修改并验证通过的，确保学生能够识别 AIGC 生成代码中可能存在的逻辑缺陷、安全隐患或性能瓶颈，并能够修正这些问题。

4. AIGC 在计算机语言类课程应用的潜在风险与教学干预措施

4.1. AIGC 在计算机语言类课程应用的潜在风险

AIGC 给计算机语言类课程教学带来的效率提升是明显的，随之而来的风险也不小。结合计算机语言类课程的实际情况，本文把风险归纳为三类。

(1) 基础能力弱化的风险。AIGC 能快速生成可直接运行的完整代码，学生一旦习惯直接拿答案，就很容易跳过语法练习、调试纠错和算法设计这些关键环节。基础语法掌握不牢，程序性知识内化不足，时间一长就出现离开 AIGC 就不会编程的情况。用认知负荷理论来解释，外部的过度替代让学习者缺失了构建编程图式所必需的精细加工过程。

(2) 学术诚信与学习动机的风险。部分学生可能直接提交 AIGC 生成的作业或项目代码，作业评价也就测不出真实的学习效果。同时，答案来得太容易会削弱学生持续钻研的兴趣，这与课程解决实际问题

能力的培养目标正好相反。

(3) 内容质量与安全风险。AIGC 的“幻觉”问题是很难回避的，它给出的代码可能语法正确但是逻辑有缺陷，例如数组越界、未释放内存、SQL 注入这类安全隐患，也可能引用并不存在的库或已经过时的接口。学生如果没有鉴别能力就直接采用，引入的不仅是错误甚至不安全的代码，还会对正确的编程范式形成错误的认知。

4.2. 针对潜在风险的教学干预措施

针对以上风险，本文在教学内容和考核评价中安排了如下干预措施，与“夯实基础知识 + 提升 AIGC 复杂编程能力”的改革思路配套实施。

(1) 检查学生解释代码。针对的是学生跑得通但说不清程序的情况。学生需要解释 AIGC 生成的关键代码块，讲清楚算法思想、时间与空间复杂度、边界条件和异常处理的原因。教师再通过追问检验学生是否真懂，防止学习停留在知其然不知其所以然的层面。

(2) 过程留痕与人机分工说明。提交复杂编程题目时，学生要标明哪些代码是 AIGC 生成的、哪些由自己修改并验证通过，并说明修改理由。教师把这些留痕情况计入平时成绩，评价更公平，学生也会更认真对待自己的学习过程。

(3) 分阶段进行能力考核。基础知识阶段安排限时、不借助 AIGC 的独立编程测试，检验学生真实的编写与调试能力；进阶阶段再考查人机协同解决复杂问题的能力。两类成绩分开评定，AIGC 的使用就不会掩盖学生真实的能力。

(4) 课堂使用规范与诚信教育。课程一开始就向学生说明 AIGC 工具允许使用的范围(基础练习须独立完成、复杂题目可协同完成)、引用与标注规则以及违规的后果，把风险防控放在教学起点，而不是出了问题再补救。

上述干预措施的共同点，是把 AIGC 的使用和学生认知发展和能力评价的全过程相结合，效率上的好处进行保留，能力培养平时靠解释、留痕、考核、规范这几个环节来监督，技术赋能与育人目标两者得以兼顾。

5. AIGC 时代下的教师角色升级

5.1. 在教学中引导学生批判性使用 AIGC 工具，避免技术滥用风险

正是由于 AIGC 工具能快速生成代码或者整体方案，因此在学生第一次接触 AIGC 工具时，就应引导学生正确使用工具，建立起批判思维，从源头上规避技术滥用。在课程开始之初，教师应引导学生了解 AIGC 产生的幻觉现象，使其了解 AIGC 工具产生的代码并不一定完全正确，只有拥有扎实的基础功底，才能分辨并纠正代码中的错误。另外，要让学生理解直接提交 AIGC 生成的完整作业或论文，是对学习目标的实质性回避，早期为学生提示风险，比在事后进行补救效果更好。

5.2. 培养学生与 AIGC 协同工作的能力

这部分是教师角色升级的关键部分。要想学生既能掌握基础代码编写能力，又能擅长 AIGC 工具使用，教师在这里起到举足轻重的作用。AIGC 工具的使用已经从一个可选项，转变为一项必须被系统训练的技能。作为教师，要指导学生用精确的自然语言或结构化的提示词清晰描述题目；并且对于 AIGC 生成的代码，学生能够判断出其是否正确、完备和安全；最重要的一点，当 AIGC 工具输出的内容不满足需求时，学生能够定位问题并有能力独自编写代码实现需求。这些优秀能力的培养和传统的教学方法不同，教师必须重新设计课程内容。

5.3. 教师从知识传授者转向 AIGC 工具引导者，培养教师 AIGC 使用能力

AIGC 工具的使用使得师生关系发生了重构[8]，教师的角色发生改变，从传统的知识传授者逐渐转变为 AIGC 工具使用的引导者，这对教师来说具有很大的挑战，因为教师作为知识主要来源的状态发生了变化，学生可以随时从 AIGC 工具获取答案。教师的核心价值已经从掌握知识转变为帮助学生解决问题。要实现这种转变，教师能力的培养是必不可少的，需要系统性的师资培训作为支撑。

教师能力提升主要有以下几个方面。首先，要掌握各种 AIGC 工具的特点。不同 AIGC 工具侧重点不同，针对不同任务的表现差异巨大，教师应当掌握各类 AIGC 工具的优缺点，为学生推荐合适的工具，而不是让学生自己艰难摸索。其次，重点训练教师将传统知识和 AIGC 工具相结合的能力，能够给学生演示如何将 AIGC 生成的内容和传统知识高效结合，让学生看到真实的案例。最后，要求教师自身能够对 AIGC 的能力边界有清晰的认知。教师能够把控边界，防止学生过度依赖 AIGC 工具，跳过构建完整基础知识体系的核心训练阶段。

因此，在教师的培训上，可以通过工作坊的形式帮助教师体验主流 AIGC 工具，鼓励教师参加 AIGC 工具厂商开展的培训交流活动，收集整理 AIGC 教学案例，及时更新工具库，定期举行教研活动，确保教师紧跟 AIGC 时代发展的步伐。

6. 结语

随着 AIGC 技术的快速发展，企业对求职者能力需求的提升，以及教育个性化发展的需求，AIGC 技术为计算机语言类课程的教学改革注入了新的驱动力。学生掌握了基础知识，才能够快速判断 AIGC 生成内容的正确性，同时，熟练掌握 AIGC 工具，才能提高编程效率。教师应转换角色，正确引导学生使用 AIGC 工具，提高学生在未来工作中的竞争力。与此同时，课程改革也不能回避 AIGC 带来的风险。将检查学生解释代码、分阶段进行能力考核等措施落实到日常教学和考核评价中去，技术带来的效率与能力培养的要求才能同时兼顾。AIGC 技术本身也在快速变化，课程内容与教师能力要求都需要随之持续更新。

基金项目

全国高等院校计算机基础教育研究会计算机基础教育教学研究项目“面向大模型编程的计算机语言类课程教学模式探索”(2025-AFCEC-687)。

参考文献

[1] 教育部等五部门关于印发《“人工智能+教育”行动计划》的通知[EB/OL]. 2026-04-10. http://www.moe.gov.cn/srcsite/A16/s3342/202604/t20260410_1433240.html, 2026-09-02.

[2] 李培, 刘擎, 白琳. 《面向学生的人工智能能力框架》指导下的“人工智能+”大学计算机基础课程改革[J]. 计算机教育, 2025(11): 63-67.

[3] 边小勇, 盛玉霞, 林晓丽, 等. 面向 AI 能力培养的计算机语言类课程培养体系构建[J]. 计算机教育, 2025(11): 170-175.

[4] 王兆静, 胡新荣, 李丽, 等. 大语言模型辅助下数据挖掘类课程教学内容与模式改革研究[J]. 计算机教育, 2025(11): 74-79.

[5] 张凯龙, 郑江滨, 马春燕, 等. AIGC 赋能的计算机系统类课程教学探索[J]. 计算机教育, 2025(11): 57-62.

[6] 卢立, 冯静雯, 王小航, 等. 生成式人工智能背景下计算机系统基础课程群面临的挑战与机遇[J]. 高教学刊, 2026, 12(16): 1-5.

[7] 何克抗. 建构主义——革新传统教学的理论基础(上) [J]. 电化教育研究, 1997(3): 3-9.

[8] 马晓榕, 李明, 陈午阳. 生成式人工智能时代教师发展的平衡之道[J]. 高等建筑教育, 2026, 35(3): 9-16.