Exploring the Teaching Reform Method of "AI+ Programming Courses": An Example is the "Fundamentals of Mapping Programming" Course

Lin Li*, Zhiwei Chen
Heilongjiang University of Technology, Jixi, Heilongjiang, China
**Corresponding Author.*

Abstract: With the development of artificial intelligence technology, AI tools are increasing, and the application of AI tools in the field of education is also increasing. By analyzing the current situation of program courses, this paper integrates the achievement of AI ability into the teaching goal. The AI+ program curriculum reform plan is proposed, and the curriculum implementation process is broken down by building personalized learning modules on online platforms and integrating AI tools into classroom teaching, so as to achieve the goal of program writing at different levels for students, better personalized teaching, and improve students' class initiative and learning interest. In this paper, the course of "Basic Mapping program Design" is taken as an example, the course is divided into several experimental units, and with the aid of AI, the program course is "1 question, 1 demonstration, 1 debugging, 1 simplification" to help learners achieve the learning goal. The improvement of this teaching method also has a certain reference value for the reform of the teaching mode of the course of computing programming.

Keywords: Artificial Intelligence; Programming Courses; Teaching Reform; AI Tools

1. Introduction

As the application of AI becomes more and more widespread, the relationship between AI and education is getting closer and closer, and the construction of AI+ courses also shows an explosion increase. As a strategic technology leading the future, AI is the core driving force of the new round of scientific and technological revolution and industrial change, and an important engine for the development of new quality productivity [1].

The rapid development of artificial intelligence has brought unprecedented opportunities and challenges to the field of higher education [2]. In the education of procedural courses, how to apply AI technology to build an innovative teaching mode and enhance students' practical ability, innovative thinking and competitiveness has become an important issue in front of teachers [3]. Applied undergraduate colleges and universities should cultivate students' ability to solve complex engineering problems in professional related fields [4,5]. Taking the course "Fundamentals of Surveying and Mapping Program Design" as an example, this paper will focus on solving the problems existing in the teaching of traditional program courses such as insufficient personalization, limited practical scenarios and lagging feedback, so as to provide a reform plan that can be popularized to improve the teaching efficiency of core courses of surveying and mapping engineering, cultivate innovative surveying and mapping talents in the intelligent era, and promote the digital transformation process of surveying and mapping engineering education.

2. Analysis of the Current Status and Problems of the Program Courses

As a professional elective course in the curriculum system of surveying and mapping engineering undergraduate, "Foundation of Surveying and Mapping Program Design" has some typical common problems in its teaching process.

2.1 Procedural Cases are Numerous but not Precise

Programming courses, no matter what programming language is used, have a lot of cases in the practical teaching section, but among the many cases, many teachers use their personal subjective opinion as a reference to select cases on the basis of their own merits, without being able to assess whether the cases are suitable for the current batch of students. Take "Fundamentals of Surveying and Mapping Programming" as an example, the chapter of formatting output is the key content of this course, and the training of the course includes cases such as calculating the area of a circle, converting Cartesian coordinates to Cartesian coordinates, and so on, and some of these cases cover a relatively lopsided coverage of the knowledge points, and some of them are rather difficult. students to learn independently, which lacks standardization.

2.2 Stronger Articulation of Knowledge Required by the Program Curriculum

Compared with other courses, programming courses, no matter which programming language is used, involve a large amount of syntax knowledge, mainly data types, expressions, statements, functions (methods), etc., which are very strong in normality and articulation, such as statements need to have a data type, expression basis, which tests the students' ability to memorize the knowledge, summarize the knowledge and other comprehensive abilities.

2.3 Inability to Achieve Complete Synchronization of Teaching and Learning in Program Courses

If learning portraits are created for students using their learning backgrounds, the differences between students will increase year by year as Internet technology continues to develop. Differences in the learning of new knowledge among students of the same grade are reflected in individual differences in comprehension and memory skills, especially in procedural courses, where more emphasis is placed on comprehension and memory, making it impossible to achieve uniformity in education, which is detrimental to the improvement of students according to the mass education approach.

2.4 Students' Competence Shortcomings are Evident

In the systematic observation and teaching feedback of programming practical training, it is revealed that there are significant shortcomings in students' ability, which are manifested in three dimensions: first, there are systematic defects in code debugging ability. Most students can only carry out superficial grammar check when the program reports errors, and lack the ability to use professional debugging tools such as breakpoint debugging and variable monitoring. When facing logic errors, they often fall into a blind trial-and-error state, resulting in the average debugging time exceeding 50% of the classroom design expectation and seriously slowing down the teaching rhythm. Secondly, the ability to solve complex problems is not structured enough. When multi-module collaborative surveying and mapping data processing tasks are involved, about 65% of the students are difficult to effectively disassemble the problem, exposing problems such as lack of algorithmic design thinking and weak awareness of modular programming. For example, in the coordinate transformation program design, some students cannot distinguish the difference between the data processing flow of geodetic coordinate system and engineering coordinate system. Finally, there are obvious faults in the cultivation of innovative thinking. About 78% of training jobs show obvious traces of code templating, and most solutions remain at the violent enumeration level, with code redundancy as high as 2-3 times the industry standard value.

2.5 Analysis of the Problem

As the core course group in the construction of new engineering, the traditional teaching mode of program course has exposed dual difficulties under the background of the current digital transformation of education. First of all, at the level of large-scale teaching, limited by the homogenized teaching paradigm of "teacher teaching + case demonstration", it is difficult to achieve accurate learning diagnosis in the teaching process in the face of the normal class type of 80-120 students. According to the cognitive load theory, there are significant individual differences in students' understanding of abstract concepts such as variable scope and recursive call. However, the existing teaching evaluation mostly relies on lagging feedback mechanisms such as after-school homework and final written test, which makes it impossible for teachers to acquire class knowledge points and master heat maps in real time, making it more difficult to implement hierarchical intervention for graded cognitive impairment. Secondly, in the dimension of technology iteration, the programming education ecology presents an exponential updating feature: according to Gartner technology maturity curve, the average life cycle of mainstream development frameworks has been shortened to 18 months, and the IDE tool chain iterates more than 5 major versions every year (such as VS Code added 12 core functions such as AI programming assistant in the past three years).This acceleration of technology makes teachers face the pressure of continuous technical debt. They not only need to maintain the stable teaching of basic syntax such as Python/JavaScript, but also need to keep track of emerging paradigms such as Web Assembly and low-code platforms, and also need to balance the teaching choices of competitive frameworks such as TensorFlow and PyTorch. This results in structural dislocation between course content update cycle (3-5 years on average) and industrial technology iteration speed (6-12 months on average), which ultimately affects the degree of matching between graduates' technology stack and job market demand.

3. Innovative Strategies for Updating and Optimizing Teaching Content Introduced by AI

The introduction of AI tools into programming courses is essential, firstly because AI technology is gradually maturing and is extremely adaptable to programming; secondly, the application of AI technology enables teachers and students to improve their learning efficiency, and is also very meaningful to students' future work. The introduction of AI will inevitably lead to the updating of the teaching content and objectives to ensure that students are better equipped with cutting-edge technology and skilled in the use of advanced tools. The introduction of AI will inevitably lead to a renewal of teaching content and objectives to ensure that students are better equipped with cutting-edge technologies and skilled in using advanced tools.

3.1 Renewal of Pedagogical Objectives for AI Integration

The teaching objectives of the course join the knowledge of AI tools, such as clarifying the achievement objectives of AI in the course, and the new knowledge and ability objectives achieved with the help of AI tools [6,7]. Through questionnaires, interviews, big data, etc. to understand the learning characteristics of the learners, including their educational background, learning style, memory level, etc., the learners are divided into three levels, the foundation consolidation layer, the ability to enhance the layer, the excellent development layer, As is shown in Figure 1,the three levels of learners to determine the learning objectives, respectively, the learning objectives should be in line with the AMART principle, that is, the Specific, Measurable, Attainable, Relevant and Time-bound, in order to achieve the customization of personalized training for different learners.

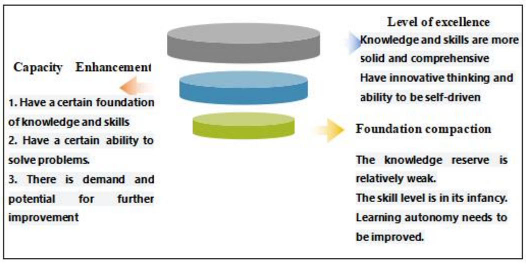

Figure 1. Individualized Learner Hierarchy

3.2 Intelligent Organization of the Teaching Process

Generative AI tools, online education platforms such as Wisdom Tree, and AI enhancements to integrated development environments bring unlimited possibilities for intelligent teaching of programming courses.

3.2.1 Build personalized learning modules using online platforms

Generative AI can create personalized learning paths and push teaching resources for learners based on their learning data and behaviors to meet the needs of different learners [8], and at the same time, the teacher's end can grasp the students' learning data in real time, so that the learning data can be quantified and the drawbacks of the original traditional empirical teaching can be broken. Teachers according to the mastery of learning data will learners' stratification, classification training, to provide personalized tutoring and real-time feedback, to help students learn programming knowledge more efficiently [9].

3.2.2 Use of AI tools for classroom teaching

One is the use of AI features in integrated development environments. For example, the new version of Visual Studio provides AI enhancements that can sense code based on context, provide complementary functions, chat assistance, debugging suggestions, and unit testing functions to help learners write and understand code faster.

Second, effective training with the help of AI tools. The use of AI tools can provide learners with a wealth of programming examples and practice cases to help students to carry out classroom focused training and self-study after class, consolidate theoretical knowledge and expand programming knowledge, so that students' programming ability can be trained.

Third, the use of AI+project-based teaching tools. The use of AI to customize complete programming projects, set the degree of difficulty of the project, allow students to experience the process of real projects, train students' innovative thinking, and cultivate students' comprehensive practical ability.

4. Teaching Reform Practices of AI+Programming Courses-Taking the Basics of Surveying and Mapping Programming as an Example

4.1 Overall Structure of the Curriculum Building

In view of the above mentioned problems in the teaching of "basic mapping programming" course, the advantages of AI technology should be brought into play to reshape the teaching mode of the course and effectively improve the quality and effect of the course teaching [10]. This paper proposes a reform programme for AI+programming course, and the overall framework is shown in Figure 2.

Different from the traditional teaching mode, this teaching mode is clear about the achievement of AI goals in the setting of teaching objectives, and in the process of classroom teaching, through generative AI tools such as ChatGPT, DeepSeek, Doubao, etc., such as program writing grammar questioning, program debugging, program simplification, etc., and through the networked learning platform to customize students' personalized learning paths, to help students' personalized cultivation. In the process of conducting course evaluation, it even introduces multiple evaluation systems such as learning process, programming results, and assessment results, in order to improve the teaching quality of the classroom.

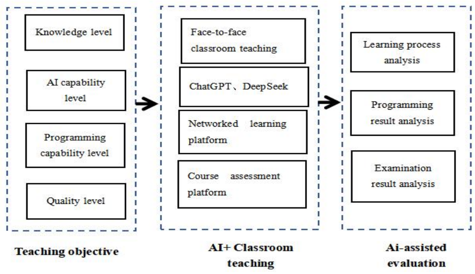

Figure 2. Curriculum Structure

4.2 AI-integrated Course Teaching and Learning Process

According to each experimental unit to build a guide case, complete the experimental project, and the use of AI prompts keywords, so that students can grasp the knowledge faster, through the process of "1 question, 1 demonstration, 1 debugging, 1 simplification" to better achieve the teaching objectives, as shown in Figure 3, the teaching implementation process is divided into beginner and advanced implementation, and the teaching objectives are different between the two. As shown in Figure 3, the teaching implementation process is divided into beginner and advanced implementation, and the teaching objectives of the two are different, the teaching objective of the beginner learners is to be able to use AI tools to write code, while the objective of the advanced learners is to be able to write code independently, and finally, the objective of the course is assessed through the course evaluation.

4.3 Validation of the Effectiveness of Teaching Reform

Teaching practice data show that the experimental class has achieved significant improvement in programming thinking ability and complex engineering problem solving ability. Through the dynamic diagnosis and personalized feedback of the multivariate assessment system, the programming thinking excellence rate of students in the experimental class increased by 23% compared with that of the control group, of which the algorithm design and code debugging abilities increased by 18% and 25% respectively. Students' task completion in complex mapping project development increased by 37%. These quantitative indexes verify the significant effect of AI-enabled teaching mode in improving higher-order thinking ability and engineering practice literacy.

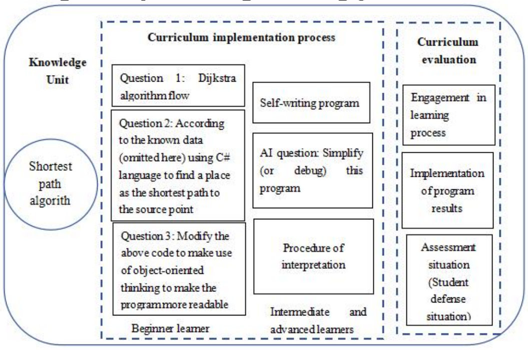

Figure 3. AI integration into Teaching and Learning Implementation

5. Conclusion

This study has systematically sorted out the status quo and pain points of the teaching of programming courses, and put forward the reform strategy of reconstructing the teaching goal system, iteratively updating the teaching content and optimizing the innovation of the teaching mode. Taking the "Fundamentals of Surveying and Mapping Programming" course as a practical carrier, the course content is deconstructed into modularized knowledge units, effectively connecting the ability cultivation needs of different learning stages. In the process of practice, the innovative "AI + case-driven" teaching mode is adopted, and the dynamic diagnosis of learning situation is realized through the multivariate assessment system. Teaching practice shows that the reform mode significantly improves students' programming thinking ability (the excellence rate of the experimental class increased by 23%) and complex engineering problem solving ability (the completion of project practice increased by 37%). This study not only verifies the feasibility of AI-enabled programming course reform, but also provides an operable implementation paradigm for programming course reform in the context of new engineering disciplines, which is an important reference value for the construction of similar courses.

Acknowledgement

This work was supported by the 2024 Heilongjiang Institute of Technology Educational Teaching Research and Reform Project-Research on the Construction of Learning Maps for ‘Artificial Intelligence+Programming Courses’ in the Context of Digital Transformation of Education.

References

[1] Zhang Yanfang, Gao Lu, Li Yan. Teaching mode innovation of programming course driven by AI technology project. Computer Education, 2025, (02): 159-163.
[2] Wang Fang, Zhao Zuo, Wang Yihang, et al. Exploration and Practice of Artificial Intelligence in Programming Teaching Application. Computer Education, 2023, (11): 45-50.
[3] Chen Hanshen. Artificial Intelligence Enabling the Teaching of Programming Courses—Taking JavaScript Programming Course as an Example. Computer Knowledge and Technology, 2024, 20(28): 137-139.
[4] Yu Bo, Peng Chen, Chen Yu, et al. Reform of Civic and Political Teaching in Programming Course of Artificial Intelligence in Applied Undergraduate Colleges. Computer Education, 2025, (02): 134-138+143.
[5] Zhang Sumin, Yin Ling, Huang Qiong, Chen XiangJi. Reform and practice of "Java language programming" course for non-computer undergraduate majors based on "Internet+" and AI. Journal of Lanzhou Institute of Vocational Technology. 2025. 2, 41(1): 41-46, 84.
[6] Lai Jinhui. Student-led teaching of AI+ Programming basic course "Exploration → Application → Creation → Debate". Computer Education, 2024, (09): 17-22.
[7] Mou Yuting, Long Huan, Jiang Hao. Teaching Practice of Computer Programming with AI Large Model. Journal of Electrical and Electronic Teaching, 2024, 46(04): 128-131.

[8] Zhang Hongzhuo, Zhou Xiaobao, Xu Yuhuan, et al. Generative Artificial Intelligence Enabling Teaching Innovation in Computer Programming Courses. Computer Education, 2024, (07): 44-48.
[9] Zhou Ta, Lu Ye. Teaching Reform and Practice of Computer Programming Curriculum Integrating AI Information Literacy. Science and Technology Information, 2019, 21(06): 146-149.
[10]Su Xiaohong, Miao Qiguang, Chen Wenyu. Individualized Teaching Model for Improving Programming Ability based on AI empowerment and Production-teaching integration. Chinese University Teaching, 2023, (06): 4-9.