
**APPLICATION OF ARTIFICIAL INTELLIGENCE IN TEACHING AND LEARNING
INTRODUCTORY PROGRAMMING AT HUNG VUONG UNIVERSITY**

THIEU THI TAI¹¹*Faculty of Engineering Technology, Hung Vuong University, Phu Tho Province, Vietnam**Email address: thieutai@hvu.edu.vn.***Corresponding author: Thieu Thi Tai**<http://doi.org/10.51453/3093-3706/2026/1477>*

ARTICLE INFO		ABSTRACT
Received:	15/3/2026	Artificial intelligence (AI) is significantly changing programming education in higher education institutions. This study investigates the application of AI in teaching Introductory Programming at Hung Vuong University and proposes an AI-assisted teaching model for first-year Information Technology students. The proposed approach integrates Flipped Classroom, Blended Learning, Project-Based Learning, and AI-assisted learning methods. Experimental results conducted with 200 students showed improvements in learning motivation, self-learning ability, programming performance, and debugging efficiency. However, the study also identifies challenges related to students' dependence on AI-generated code and assessment of actual competencies. The research suggests that integrating AI into programming education is an effective direction for improving Information Technology training quality in the digital transformation era.
Revised:	20/4/2026	
Published:	28/4/2026	
KEYWORDS		
<i>Artificial Intelligence; Introductory Programming; AI-assisted Learning; Programming Education; Digital Transformation.</i>		

1. INTRODUCTION

In recent years, artificial intelligence (AI) has become one of the core technologies driving the digital transformation process in higher education. The strong development of generative AI tools like ChatGPT, GitHub Copilot, Gemini, and Claude has significantly changed the learning methods of students as well as the teaching methods of lecturers in the field of Information Technology. For the Information Technology major, Basic Programming plays a role as a foundational module helping students form algorithmic thinking, problem-solving skills, and software development capabilities. However, reality shows that many first-year students encounter difficulties during the programming learning process due to limitations in logical thinking, problem analysis skills, and self-study abilities.

At Hung Vuong University, the Basic Programming module is taught to first-year Information Technology students to equip them with foundational knowledge about programming and develop algorithmic thinking capacity. However, traditional teaching methods mainly focus on theory and programming language syntax, and have not yet truly promoted the proactivity and creativity of learners. The emergence of AI programming support tools opens up many opportunities to innovate teaching and learning methods. AI has the ability to support students in explaining algorithms, detecting program errors, generating source code, and personalizing learning activities. However, the

use of AI also poses many challenges such as the risk of technology dependence, the decline of algorithmic thinking, and difficulties in assessing the true capacity of students.

Stemming from the above reality, this study is conducted to analyze the impact of AI on teaching the Basic Programming subject, and simultaneously propose an AI-integrated teaching model suitable for the training conditions at Hung Vuong University in the current digital transformation context.

2. LITERATURE REVIEW

In recent years, the strong development of Generative AI tools such as ChatGPT, GitHub Copilot, and Gemini has significantly changed teaching and learning methods in the Information Technology field, especially for basic programming modules. Many international studies show that AI is gradually becoming a popular learning support tool in programming education, helping personalize learning, support debugging, and improve students' programming practice efficiency.

In the study “Integrating Generative AI into Programming Education” published in 2025 in Education and Information Technologies, the author group A. Kumar, J. Lee, and M. Rodriguez argued that AI helps students improve their ability to explain source code, supports program debugging, and enhances creativity when learning programming [1]. However, the research team also pointed out that students often face difficulties in detecting logical errors or evaluating the optimality of AI-generated source code if they lack a solid algorithmic thinking foundation.

Another notable study by Hwang, Chen, and Chang in 2025 titled “The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance” published in the International Journal of STEM Education showed that combining AI in pair programming significantly improves learning motivation, reduces psychological pressure, and enhances the programming practice results of first-year IT students. Survey results show that the group of students using AI had a higher level of learning engagement and collaboration ability than the group learning with traditional methods [2].

The study “Does generative AI help in learning programming” by Becker et al. published in 2025 in Computers and Education: Artificial Intelligence showed that although AI helps students speed up the coding process and reduce debugging time, students who use AI frequently tend to achieve lower results in algorithmic thinking assessment tests if not guided to use AI correctly. The research team emphasized that AI supports the programming learning process but cannot replace the learner's thinking and problem-solving capacity [3].

In the large-scale study “Teaching introductory programming in top universities: A global study of languages, paradigms, assessment, and AI” published in 2026 in the Journal of Computer Languages, the author group Simon, Porter, and Zingaro surveyed 150 basic programming modules at 83 top universities worldwide. Results showed that many educational institutions have changed their assessment methods to adapt to the emergence of AI by reducing the weight of simple coding exercises and increasing the assessment of problem-solving thinking, direct testing, and algorithmic interviews [4].

In the 2025 study “Teaching Introduction to Programming in the times of AI: A case study of a course re-design”, author Juho Leinonen and colleagues proposed a model to redesign the basic programming module towards integrating AI as an official learning tool instead of banning its use. The research team suggested shifting the training focus from memorizing syntax to developing critical thinking, the ability to verify AI results, and algorithm explanation skills [5].

In addition, the study “Three Years with Classroom AI in Introductory Programming” by Leinonen, Hellas, and Ihantola published in 2026 tracked the process of using AI in Python programming classes during the 2023–2025 period. Results showed that students' perception of AI had clearly changed as AI gradually became a familiar learning tool. At the same time, the lecturer's role also shifted from transmitting knowledge to orienting thinking, monitoring, and supporting students in verifying the results generated by AI [6].

Another notable evidence is the study by Prather, Becker, and Denny presented at the ICSE 2026 conference analyzing 1,614 Python programming submissions by students during the 2022–2024 period. Results showed that the rate of copying assignments among students decreased by about **60,11%**, but the similarity level between students' work and AI-generated source code increased sharply. This reflects the trend of shifting from traditional copying to relying on AI in learning programming, thereby posing an urgent requirement to innovate testing and assessment methods in basic programming training today [7].

Besides academic studies, many lecturers and IT education experts also reflect the reality that students are increasingly dependent on AI during the programming learning process. Some opinions suggest that although students can quickly create software products with AI support, they encounter difficulties when having to self-analyze algorithms, debug, or explain the operating principles of programs without AI assistance.

In general, recent studies show that AI is opening up many innovation opportunities in teaching basic programming, such as personalized learning, enhanced practice, and support for developing programming skills. However, AI also poses many challenges related to algorithmic thinking, academic ethics, and assessing the true capacity of students. This is an important basis for training institutions to research and redesign teaching methods for basic programming subjects in the current artificial intelligence era.

3. ROLES AND CHALLENGES OF AI IN TEACHING AND LEARNING BASIC PROGRAMMING

3.1. Supporting Personalized Learning

One of the outstanding advantages of AI is its ability to support personalized learning. AI systems can analyze each student's learning capacity to suggest exercises suitable for the learner's level and absorption speed. Students can learn anytime, anywhere and receive near-instant feedback during programming practice. AI also helps students easily access new knowledge by explaining algorithms, describing the operating principles of programs, and offering multiple solutions for the same problem. This contributes to improving students' proactivity and self-study abilities.

3.2. Supporting Lecturers in the Teaching Process

For lecturers, AI provides significant support in building learning materials, creating exercises, and evaluating students' programs. AI tools can:

- Generate sample code;
- Create sets of programming questions;
- Support automatic grading;
- Analyze students' common errors.

Thanks to this, lecturers have more time to focus on guiding problem-solving thinking and organizing in-depth learning activities.

3.3. Enhancing Project-Based Learning

AI helps students develop software products faster by supporting code generation, suggesting program structures, and software testing. This creates conditions for strongly implementing the Project-Based Learning method, helping students get closer to the practical working environment of IT enterprises.

3.4. Limitations and Challenges of AI in Learning Programming

Although bringing many benefits, the use of AI in learning programming also poses many challenges. First of all, many students tend to rely too much on AI during the learning process. Instead of self-analyzing problems and building algorithms, students often directly copy the source code generated by AI without clearly understanding the operating principles. This leads to a decline in logical thinking abilities and problem-solving skills – the core competencies of programmers.

In addition, AI sometimes provides unoptimized code snippets or contains logical errors and security flaws. If students lack foundational knowledge, they are very likely to accept results from AI without the ability to verify them. Another issue is the difficulty in assessing students' true capacity. When AI can solve most basic programming exercises, traditional testing formats like homework or small projects no longer accurately reflect the learner's level. This requires training institutions to change their testing and assessment methods to suit the new context.

4. INNOVATING TEACHING METHODS FOR BASIC PROGRAMMING IN THE DIGITAL ERA AT HUNG VUONG UNIVERSITY

Hung Vuong University is a multi-disciplinary training institution under the Phu Tho Provincial People's Committee. The university has the mission of training high-quality human resources with an application orientation, and at the same time is a center for scientific research and technology transfer serving the socio-economic development of Phu Tho province and the Northern midland and mountainous region. Currently, Hung Vuong University offers training in many fields such as Information Technology, Engineering - Technology, Economics, Agriculture - Forestry -

Fishery, Foreign Languages, Tourism, Social Sciences, and Pedagogy with levels ranging from undergraduate to postgraduate.

The university focuses on innovating teaching methods, developing digital learning materials, deploying online learning systems, and increasing technology application in training to meet the requirements of digital transformation and international integration. With the motto of development oriented towards application, linking training with practical needs, the university promotes corporate cooperation, scientific research, and creative startup activities for students.

In recent years, Hung Vuong University has paid special attention to developing majors in the field of Information Technology and digital transformation to meet the demand for high-quality human resources in the Industry 4.0 era. In the context of digital transformation and the strong development of artificial intelligence, the requirement to innovate teaching methods in Information Technology majors is becoming increasingly urgent. Especially, for the Basic Programming module - a foundational subject for Information Technology students, innovating teaching methods not only aims to improve training quality but also helps students develop algorithmic thinking, self-study capacity, and problem-solving skills in the modern digital environment.

Stemming from the orientation of training linked to reality and applying new technologies, Hung Vuong University is step-by-step researching and deploying teaching models integrating digital technology and artificial intelligence to improve the effectiveness of teaching Basic Programming in the current AI era. Specifically as follows:

4.1. Shifting Focus from Teaching Syntax to Developing Thinking

In the context where AI can automatically generate source code, programming training objectives should not just stop at “being able to write programs” but need to focus on developing:

- Algorithmic thinking;
- Problem analysis skills;
- Critical thinking;
- Program testing and optimization skills.

Lecturers need to increase activities that require students to explain algorithms, analyze complexity, and evaluate source code quality.

4.2. Integrating AI as an Official Learning Tool

Instead of banning the use of AI, training institutions need to develop reasonable guidelines for using AI in learning. Students need to be trained in the skills of:

- Asking effective questions to AI;
- Verifying results generated by AI;
- Analyzing and critiquing source code;

-
- Identifying errors and security risks.

AI should be considered a learning support tool, not a tool to replace programming thinking.

4.3. Innovating Assessment Methods

Assessment formats are shifted from testing the ability to memorize syntax to assessing problem-solving capacity. Some suitable solutions currently being applied for regular assessment and end-of-module assessment:

- Direct in-class programming exams;
- Algorithmic explanation interviews;
- Assessing the learning process;
- Requiring students to analyze AI-generated source code;
- Organizing group projects linked to practical problems.

4.4. Combining Modern Learning Models

Currently, the university is applying a learning model that combines traditional teaching methods and online learning methods on the LMS system. Especially for Basic Programming and other programming subjects, the university also applies the project-based learning model, helping students gain deeper insights into programming thinking and project-based working thinking. Students can self-study theory via LMS and AI, while class time focuses on discussion, coding practice, debugging, and solving practical problems.

5. RESEARCH RESULTS AND TESTING OF THE NEW TEACHING MODEL AT HUNG VUONG UNIVERSITY

To evaluate the effectiveness of applying artificial intelligence in teaching the Basic Programming subject, the research team conducted testing of the AI-integrated teaching model with first-year Information Technology students at Hung Vuong University. The university is currently a multi-disciplinary public higher education institution with a training scale of over 6,000 students and trainees, in which the engineering - technology sector is being focused for development to meet the demand for digital transformation human resources in the midland and Northern mountainous regions.

The research was implemented on nearly 400 first-year Information Technology students in the Basic Programming module. Students were divided into two groups:

- The control group consisting of 100 students learning via the traditional method;
- The experimental group of 100 students learning via the AI-integrated model.

The teaching model and assessment methods implemented according to the above proposals have achieved many encouraging results. During the learning process, students in the experimental group were guided to use AI tools such as ChatGPT, GitHub Copilot, and Gemini to:

-
- Explain algorithms;
 - Suggest program bug fixes;
 - Create practice exercises;
 - Analyze source code;
 - Support building small programming projects.

However, lecturers required students to:

- Re-explain the algorithms using personal thinking;
- Analyze the pros and cons of AI-generated source code;
- Self-optimize programs;
- Perform direct coding during in-class tests.

After one semester of implementation, the research results show that the new teaching model brings many positive changes. The rate of students completing programming assignments on time in the experimental group reached about **88%**, significantly higher than the traditional learning group at only about **65%**. The average time for students to detect and fix program errors decreased by about **35%** thanks to AI support during the debugging process.

The end-of-module survey results show:

- About **82%** of students thought AI helped them understand lessons faster;
- **76%** of students felt more confident when learning programming;
- **79%** of students increased their self-study ability outside of regular class hours;
- **71%** of students reported clearly increased learning interest compared to the beginning of the semester.

In particular, the project-based learning method combined with AI helped students improve teamwork and practical problem-solving skills. The number of student groups completing the final product satisfactorily increased from **68%** to about **90%**.

Besides the positive results, the research also noted some remarkable limitations. About **30%** of students had a tendency to depend on AI in writing source code and faced difficulties when having to build algorithms from scratch by themselves. Some students also lacked the skills to verify the results generated by AI, leading to the use of unoptimized code snippets or containing logical errors. In addition, lecturers also faced difficulties in evaluating students' true capabilities when AI could support generating fairly complete code.

This forced the testing and assessment process to change in the direction of:

- Enhancing direct practical exams;
- Algorithmic explanation interviews;
- Assessing the learning process;
- Evaluating the ability to critique AI source code.

Research results show that integrating AI into teaching basic programming is an appropriate trend in the current digital transformation context. However, AI is only truly effective when used as a learning support tool instead of completely replacing students' programming thinking. Therefore, the training model in the AI era needs to focus on developing algorithmic thinking, problem-solving skills, and technological critical capacity for learners.

The initial testing results at Hung Vuong University show that the AI-integrated teaching model has the potential to improve the quality of Information Technology training, contributing to meeting the requirements for training high-quality digital human resources in the context of current technological integration and development.

6. CONCLUSION AND DEVELOPMENT DIRECTION

The study analyzed the impact of artificial intelligence on teaching and learning Basic Programming at Hung Vuong University, Phu Tho province. Results showed that integrating AI into the teaching process helped improve learning efficiency, increase self-study capacity, and improve programming practice skills for first-year Information Technology students.

However, AI cannot completely replace the role of learners' algorithmic thinking and problem-solving capabilities. Therefore, AI needs to be considered a learning support tool instead of a tool to replace programming thinking. The teaching model integrating Blended Learning, Project-Based Learning, and AI-assisted Learning is evaluated as suitable for the current digital transformation context and capable of improving Information Technology training quality at Hung Vuong University.

REFERENCES

1. A. Kumar, J. Lee, and M. Rodriguez, "Integrating Generative AI into Programming Education," *Education and Information Technologies*, vol. 30, no. 4, pp. 5213–5231, 2025.
2. G.-J. Hwang, C.-H. Chen, and C.-Y. Chang, "The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance," *International Journal of STEM Education*, vol. 12, no. 1, pp. 1–18, 2025.
3. B. A. Becker, J. Prather, P. Denny, and J. Leinonen, "Does generative AI help in learning programming?," *Computers and Education: Artificial Intelligence*, vol. 7, Article 100246, 2025.
4. S. Simon, L. Porter, and D. Zingaro, "Teaching introductory programming in top universities: A global study of languages, paradigms, assessment, and AI," *Journal of Computer Languages*, vol. 78, Article 101286, 2026.
5. J. Leinonen, A. Hellas, and P. Ihanola, "Teaching Introduction to Programming in the Times of AI: A Case Study of a Course Re-design," *arXiv preprint arXiv:2503.01876*, 2025.
6. J. Leinonen, A. Hellas, and P. Ihanola, "Three Years with Classroom AI in Introductory Programming," *Proceedings of the 2026 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 214–220, 2026.
7. J. Prather, B. A. Becker, and P. Denny, "AI-generated Code and Academic Integrity in Introductory Programming Education," in *Proceedings of the 48th International Conference on Software Engineering (ICSE 2026)*, Shanghai, China, 2026, pp. 134–145.