

Building a Hierarchical and Progressive Teaching Model for PythonT Programming Empowered by Generative AI

Shihao Wang*

Anhui University of Chinese Medicine, Hefei 230013, Anhui, China

**Author to whom correspondence should be addressed.*

Copyright: © 2026 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), permitting distribution and reproduction in any medium, provided the original work is cited.

Abstract: Against the background of the digital era, the in-depth integration of artificial-intelligence technology and higher education has become a core trend of educational reform. Intelligent teaching, personalized teaching and targeted talent cultivation have turned into the core development directions of university curriculum teaching in the new era. Featuring concise syntax, wide application scope and strong expandability, Python programming covers multiple technical fields including big data, artificial intelligence, data analysis, back-end development and automated operation-and-maintenance. It constitutes a fundamental programming skill essential for contemporary college students, and also a core university course for cultivating computational thinking and science-and-innovation-oriented talents. The hierarchical and progressive teaching model for Python programming empowered by generative AI realizes precise adaptation and continuous advancement in the whole teaching process through constructing a dynamic student-stratification system, a stepped teaching-content system, an intelligent hierarchical-tutoring system and a process-oriented hierarchical-evaluation system. Accordingly, this paper conducts research on constructing the hierarchical and progressive teaching model for Python programming empowered by generative AI, hoping to provide certain references for relevant researchers.

Keywords: Generative AI; Python programming; Hierarchical and progressive; Teaching model

Online publication: October 27, 2026

1. Construction of the framework for hierarchical and progressive teaching model of Python programming empowered by generative AI

1.1. Dynamic student-stratification system to eliminate drawbacks of static stratification

The hierarchical and progressive teaching model abandons previous static and fixed stratification modes. It leverages generative AI to build a dynamic adaptive stratification system, which realizes gradual upgrading, dynamic adjustment and real-time stratification with reference to students' whole-process learning behaviors. Within this teaching model, teachers divide students into three gradients: basic consolidation layer, competence-improvement layer and innovation-elite layer according to the learning rules of Python courses

and students' ability characteristics. Each gradient has no fixed quota and no permanent membership ^[1].

The basic consolidation layer targets students with zero-foundation or weak foundations. Its main goal is to enable students to master basic syntax, get familiar with code-writing specifications and acquire the capability to write simple programs. The competence-improvement layer is for students who have mastered basic syntax and possess fundamental practical capabilities. It aims to help students skillfully utilize various functions and data structures, and gain the ability to independently debug code and develop medium-sized programs. The innovation-elite layer is oriented toward students with solid foundations and strong logical thinking. Its objective is for students to master module application and framework development, and obtain the ability to independently finish comprehensive projects and innovative development.

Generative AI keeps track of students' classroom performance, project completion quality, code-debugging speed and other conditions throughout the process. It automatically judges whether students' current layers are suitable and adjusts students' gradient levels on a regular basis. It encourages students at each layer to make steady progress and keep improving, so as to create a benign stepped-growth environment ^[2].

1.2. Stepped teaching-content system to adapt to progressive growth demands

With the help of the intelligent-generation function of generative AI, teachers integrate the knowledge logic of Python courses and match it with divided student layers, so as to establish a three-level progressive teaching-content system that achieves content stratification, increasing difficulty and systematic coherence.

For students in the basic consolidation layer, AI mainly focuses on core basic knowledge points of the course, simplifies complex theoretical logic, and generates lightweight and vivid courseware, introductory cases, and basic exercises. The contents center on syntax cognition, simple code writing and basic error handling, helping students rapidly build programming thinking, consolidate course foundations and dispel their fear of learning.

For students in the competence-improvement layer, AI mainly carries out integrated application and logical expansion of knowledge points, and generates special-training question banks, typical-case analyses and code-optimization training materials. It takes key contents such as function encapsulation, nested loops, data processing and exception handling as training targets, so as to improve students' proficiency in code writing and logical-debugging capabilities.

For students in the innovation-elite layer, AI lays emphasis on comprehensive, innovative, and practical content, and generates industry practical projects, extended module applications, and framework-development cases. It guides students to conduct comprehensive practical training in scenarios such as big-data processing, automated scripts and simple software development, and cultivates their project-development and innovative-application capabilities.

The three-level content system advances step by step and upgrades progressively. It enables students with weak foundations to get steadily started, and meanwhile provides room for expansion and improvement for outstanding students ^[3].

1.3. Intelligent hierarchical-tutoring system for all-weather targeted teaching

Relying on the real-time interaction and intelligent Q&A functions of generative AI, teachers build a hierarchical-tutoring system covering pre-class, in-class and after-class sessions, so as to solve problems such as lack of pertinence and precision in traditional teaching tutoring.

In the pre-class stage, AI pushes preview content in a stratified manner according to students' learning situations. Students at the basic layer preview basic concepts and simple cases; students at the improvement

layer preview key and difficult knowledge and application skills; students at the elite layer preview extended knowledge and project ideas, laying foundations for subsequent classroom teaching.

In the in-class stage, AI is deeply integrated into classroom teaching. It captures students' performance in classroom exercises and practical operations in real time, provides real-time Q&A and instant error correction for classroom questions raised by students of different layers: basic questions are explained on the spot, logical questions are sorted out carefully, and extended questions receive special guidance.

In the after-class stage, AI generates personalized after-class assignments, consolidation exercises and extended tasks for students of each layer according to their learning performance. It automatically generates error-correction analyses, exercises of similar question types, and knowledge-point reviews aimed at wrong codes and weak knowledge points in students' assignments, so as to realize "accurate correction of wrong questions, targeted reinforcement of weak points, and continuous improvement of strengths". Meanwhile, students can raise questions to AI at any time and obtain one-on-one practical-operation guidance, idea guidance and knowledge-explanation services. It provides personalized learning companionship round the clock and makes up for shortcomings in manual tutoring by teachers ^[4].

1.4. Process-oriented hierarchical-evaluation system for building a multi-dimensional progressive evaluation mechanism

Teachers make use of generative AI's whole-process data tracking and intelligent-analysis functions to reconstruct the process-oriented evaluation system for hierarchical-progressive teaching and break the traditional single-form evaluation. The evaluation system abandons unified evaluation criteria and sets differentiated evaluation indicators, which mainly examine students' progress range in learning, knowledge-mastery degree, capability-improvement status and effects of autonomous learning ^[5].

For example, students in the basic consolidation layer are evaluated mainly from three dimensions: mastery of basic knowledge points, assignment-completion quality and learning-progress range as well as introductory practical capabilities, with the focus on consolidating foundations and growth in learning attitudes. Students in the competence-improvement layer are evaluated mainly from three dimensions: code-writing proficiency, logical-thinking competence, problem-debugging competence and knowledge-application capacity, focusing on capability improvement and skill consolidation. Students in the innovation-elite layer are evaluated mainly from four dimensions: project-development capacity, innovative thinking, extended knowledge-application and comprehensive practical capabilities, with emphasis on innovative breakthroughs and improvement of comprehensive qualities.

In addition, AI tracks students' exercise answers, training data, and the whole learning process, and automatically generates personal learning-evaluation reports and growth files. It timely identifies weak links and progress directions of students at each layer, and offers powerful support for students' continuous development and teachers' improvement of targeted teaching ^[6].

2. Implementation paths for generative-AI-empowered hierarchical and progressive teaching of Python programming

2.1. Pre-class intelligent hierarchical lesson-preparation to build a personalized preview system

Before the hierarchical-progressive Python teaching, teachers adopt generative AI for intelligent and stratified

lesson-preparation, and construct a preview system satisfying the demands of students at all layers to lay the foundations for in-class hierarchical teaching.

First of all, teachers utilize AI to sort out and summarize the hierarchical-knowledge structure of Python course chapters, identify basic knowledge points, key and difficult points, and extended-advancement points, and establish a stepped teaching syllabus to make teaching content more hierarchical and logical.

Secondly, teachers deeply analyze students' learning data from the previous stage. AI automatically completes initial stratification of students and accurately locates each student's foundation level and weak links. According to stratification results, AI intelligently generates preview resources and preview tasks in a stratified way. For instance, lightweight preview content, including knowledge popularization, syntax introduction and appreciation of simple cases, is provided for basic-layer students to lower learning difficulty; content including analysis of key and difficult points, practical cases of basic common-sense knowledge and special preview exercises are offered for improvement-layer students to deepen their knowledge understanding; extended-knowledge scopes, simple-project conception and industry-application cases are given for elite-layer students to stimulate their innovative awareness.

Through pre-class stratified preview, students of each layer finish preview work in advance, narrowing gaps in classroom learning and laying solid foundations for targeted and progressive classroom teaching^[7].

2.2. In-class targeted hierarchical teaching to realize stepped competence cultivation

In the in-class teaching stage, the mixed-teaching mode of “teacher-led + AI-empowered” is adopted to achieve the teaching goals of stratified lecturing, stepped training and progressive improvement.

For example, in the session of basic-theory explanation, teachers give unified explanations of key knowledge points to guarantee that all students grasp basic content. Generative AI is used to present stratified knowledge points in real time: basic-layer students consolidate their foundations through popularized interpretations of knowledge points; improvement-layer students deepen their understanding of the course via in-depth analyses of key and difficult points; elite-layer students broaden their knowledge scope by learning extended knowledge points.

In the in-class practical-training session, teachers push three-level stepped practical tasks via AI. Practical tasks for basic-layer students cover basic-syntax coding and simple program running; practical tasks for improvement-layer students include code optimization, logical nesting and exception handling; practical tasks for innovation-elite-layer students are small-scale function development and combined module application.

Meanwhile, while students independently finish practical tasks, AI monitors their operations in real time. It gives instant error correction aimed at students' personalized mistakes, provides timely guidance when students get stuck in thinking, and reasonably pushes advanced tasks. It ensures every student can improve to the maximum extent within their own capacity range, and realizes stratified adaptation and stepped growth in classroom learning^[8].

2.3. After-class stratified consolidation and expansion to build a closed-loop progressive learning system

The after-class period is a key link for students to consolidate knowledge and achieve progressive advancement. Teachers leverage generative AI to construct an after-class-learning system featuring stratified consolidation, dynamic expansion and closed-loop improvement, so as to effectively extend the scope of classroom teaching.

In the after-class-assignment session, AI abandons the unified-assignment mode. It generates personalized progressive stratified assignments according to students' real-time layers and learning status. The basic layer mainly focuses on consolidation of basic-type exercises and checking-and-filling of knowledge gaps; the improvement layer focuses on advanced special-type exercises and strengthening application capabilities; the innovation-elite layer focuses on cultivation of comprehensive programming tasks and practical-combat-ability training.

After assignments are finished, AI automatically generates correction reports and revision schemes for code assignments. It conducts personalized correction according to the types and characteristics of students' mistakes and pushes similar reinforcement exercises, so as to effectively make up for students' weak links.

At the same time, AI automatically generates stepped developmental learning materials according to students' learning situations. It enables students in the basic layer to advance step-by-step toward the improvement layer, students in the improvement layer to strive for the elite layer, and students in the elite layer to carry out project-oriented innovative development. Thus, a closed-loop learning system of “learning — consolidation — advancement — progression” is formed, guaranteeing students' sustained and stepped improvement in various capabilities ^[9].

2.4. Dynamic layer adjustment to achieve stepped-progressive growth for all students

To prevent stratification from becoming solidified and guarantee continuous progress for all students, teachers adopt AI to conduct layer evaluation and adjustment on a weekly basis.

Generative AI conducts comprehensive evaluation of students by synthesizing multi-dimensional data including students' previous effects in the current week, in-class practical-training performance, after-class assignment quality, knowledge-mastery level, and autonomous-learning duration. Students' learning layers are appropriately adjusted according to their learning status, so that students always stay in layers suitable for themselves and march toward higher-level learning layers.

In addition, teachers conduct manual re-examination of students' layer adjustments according to AI evaluation reports. They provide personalized guidance in light of students' learning situations and psychological conditions, encourage students at lower layers to forge ahead and push students at higher layers to keep improving. It breaks the shackles of fixed stratification, creates a growth mechanism in which all can progress, layers can flow in the whole process and comprehensive improvement can be realized, and truly achieves the goals of teaching students in accordance with their aptitude and progressive talent cultivation ^[10].

3. Conclusion

To sum up, the generative-AI-empowered hierarchical-progressive teaching model for Python realizes effective solutions to the dilemmas of “one-size-fits-all” teaching and “static stratification” in traditional teaching through the coordinated linkage of four systems: dynamic stratification, stepped contents, intelligent tutoring and process-oriented evaluation. This model not only realizes precise matching of teaching resources and personalized customization of learning paths, but also constructs a benign growth ecosystem where all can progress, layers can flow throughout the whole process and comprehensive improvement can be achieved. In the future, with continuous technological iteration and deepening educational concepts, hierarchical-progressive teaching will release its educational value on a broader scope and at a higher level.

Disclosure statement

The author declares no conflict of interest.

References

- [1] Shi G, Hu Y, Lu C, 2026, Exploration on Stratified-Teaching Reform of the Course Practical Logistics Data Analysis Driven by AI. *Logistics Engineering and Management*, 48(2): 152–155.
- [2] Wang Y, Zhu Y, Yin T, et al., 2025, Teaching Practice of Web-Development Course Oriented Toward Vocational Competence: “AI-Empowered, Project-Led, Hierarchical-Progressive”. *Computer Knowledge and Technology*, 21(36): 166–170.
- [3] Zhao Z, 2025, Practice of AI Dynamic Stratified-Teaching Model in University Optional Football Courses. *Sports Style*, (19): 41–43.
- [4] Sun J, 2025, Construction of Multi-Dimensional Stratified-Case-Teaching Model for Higher-Vocational Economic-Law Course Empowered by AI. *Investment and Cooperation*, (9): 208–210.
- [5] Chang X, 2025, Empirical Research on AI-Supported Hierarchical Interactive Teaching Strategies and Effects in Java Classroom. *Information & Computer*, 37(17): 212–214.
- [6] Guo Y, Chu X, Zhao T, 2025, Experience in Applying Hierarchical-Progressive Teaching Mode to Clinical Immunoassay Internship Teaching. *Science & Education Article Collects*, (15): 154–158.
- [7] He Y, Huang B, 2025, Application of Hierarchical-Progressive Teaching Mode Based on Intelligent Technology in Internship Teaching for Medical-Laboratory-Technology Major. *China Continuing Medical Education*, 17(4): 80–84.
- [8] Dai W, Chu D, Wang S, et al., 2023, Research on the Reform of Integrating Online-Offline Mixed-Teaching Mode Under Hierarchical-Progressive Education Into Chemistry of Chinese Materia Medica Experimental Teaching. *Guangdong Chemical Industry*, 50(23): 172–174.
- [9] Zhou A, Fu X, Du S, et al., 2022, Research on Hierarchical-Progressive Practical-Teaching Mode for Computer-Aided Mapping Based on OBE Concept. *University Education*, (7): 66–68.
- [10] Zhang X, 2020, Exploration on the Reform of Hierarchical-Progressive Extracurricular-Practical-Teaching Mode for Internet-of-Things Major. *Office Automation*, 25(6): 33–34 + 61.

Publisher's note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.