IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 22 November 2022, accepted 16 January 2023, date of publication 25 January 2023, date of current version 1 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3240105

RESEARCH ARTICLE

Research on Construction and Practice of Precision Teaching Classroom for University Programming Courses

FANG YU[1], YAN LIU[2], AND FENGYAN XIAO[3]

[1]School of Information Science and Technology, Baotou Teachers' College, Baotou, Inner Mongolia 014030, China
[2]College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, Inner Mongolia 010010, China
[3]Faculty of Teacher Education, Baotou Teachers' College, Baotou, Inner Mongolia 014030, China

Corresponding author: Fang Yu (yuyueer520@163.com)

This work was supported in part by the Inner Mongolia Natural Science Foundation of China under Grant 2019MS06037(2019), in part by the Key Project of Teaching Reform of Baotou Teachers' College under Grant BSJG21Z002(2021), and in part by the Key Project of Teacher Education of Baotou Teachers' College under Grant JSJY2021Z03(2021).

ABSTRACT Facing the challenges of programming courses teaching in university, this study constructs a precision teaching classroom for programming courses that integrates precision teaching theory. Firstly, this paper analyzes the realization conditions for precision teaching in programming courses, then designs the implementation process of precision teaching including three teaching phases and five teaching links, and finally carried out teaching practice in C Language Programming course. The practice results showed that in the precision programming teaching classroom, teacher-student interactions were more frequent and students had higher programming learning efficiency than in ordinary programming teaching classroom. Compared with the control class, students in the experimental class were more capable of grasping programming knowledge and had better programming problem-solving abilities, and most of them were highly satisfied with teaching design of precision teaching classroom for C Language Programming. Precision programming teaching classroom constructed by this research provides a more optimized and effective teaching method for university programming teaching. On the other hand, this study provides a practical case for the integration and application of precision teaching theory and university curriculum teaching.

INDEX TERMS Precision teaching, programming teaching, learning efficiency, programming abilities, learning satisfaction, C language programming course.

I. INTRODUCTION

Programming courses are very important for computer specialty in university, many researchers believe that programming courses could promote students' Computational Thinking and skills of programming [1], [2].

However, how to effectively conduct programming teaching to develop the skills of programming for students has become a challenging issue [3]. There are many researches from different angles, different theories and curriculum design methods to optimize the learning effect of programming courses. There was research that used Web Programming Grading Assistant (WPGA) to study students' learning effectiveness of lower-division blended-instruction computer science course. WPGA tracked and modeled students' programming learning behaviors and results proved that diligent students achieved higher exam scores on average. Using programming learning analytics empowered instructors to better advise students as to how they should improve their learning processes [4]. The researcher presented the instruction of computer programming using adaptive learning activities considering students' cognitive skills based on the learning theory of the Revised Bloom Taxonomy (RBT). They developed an adaptive tutoring system for supporting undergraduate students in the C# programming language course, and used the technology of fuzzy weights in a rule-based decision-making module and the learning theory of a RBT for designing the learning material. The results

The associate editor coordinating the review of this manuscript and approving it for publication was Aasia Khanum.

showed that the presented approach outperforms others which lack adaptivity in domain knowledge and learning theories, improving significantly the students' programming learning outcomes [5]. A programming learning design with online gamification activities was proposed. The purpose of this study was to identify the effect of the type of player on the learning outcomes of online gamification activity programming. The Kruskal-Wallis test showed that there was no significant difference between the types of players on student learning outcomes. These results proved empirically that the type of player did not significantly affect the learning outcomes of online gamification activity programming [6]. No matter from which perspective to improve students' programming learning, if students don't pass the programming course, it will affect their subsequent learning of other professional courses [7]. Although there are many methods to optimize programming teaching [8], [9], there are still many students who give up learning programming course and fail to pass the programming course [10]. As a result, while students may be familiar with programming knowledge, it is difficult to complete programming tasks independently. Gradually, students feel that programming courses are difficult to learn and lose their confidence and interest in learning programming [14].

In the 1960s, Ogden Lindsley put forward the concept of Precision Teaching (PT) [12]. The precision teaching at this stage was based on Skinner's behavioral learning theory, which embodied five main principles: ''learners know best'', ''focus on directly observable behaviors'', ''evaluate students' performance based on behavioral frequency'', ''use standard charts to evaluate learning progress'', and ''describe the environment and conditions under which learning affects behavior'' [13]. In the early 1970s, precision teaching was continuously developed under the influence of the ''zone of proximal development theory'', ''learning hierarchy theory'' and ''mastering learning theory''. Precision teaching began to move from the field of special education to the field of general education, gradually entered the primary school classroom, from a single teaching object to class oriented group teaching, which had been accepted and applied by a wider audience [14]. In 1992, the tenth precision teaching conference established the standard speed change association, and precision teaching was officially and more widely recognized [15]. In order to grasp the effect of accurate teaching, researchers had constructed a measurement index to describe students' mastery of knowledge and skills, which was called ''Fluency''. This indicator was used to measure students' mastery of learning, and was considered to be the core of precision teaching [16]. Fluency measured the mastery of knowledge with a series of clear words, such as smooth, free, skilled, etc., and abandoned the traditional method of using accuracy to describe the mastery of knowledge.

For more than half a century year, many countries have carried out practical research on precision teaching, mainly through some teaching experiments to evaluate and verify the effect of precision teaching. These studies have shown that precision teaching can effectively remedy students' lack of academic skills, and been proved to have a certain role in promoting different groups of learners [17]. A Study proved that precision teaching could shorten learners' learning time and significantly improve students' reading ability [18], and another study found that PT was helpful to solve students' mathematical learning difficulties [19]. Also, there was research focused on investigating the effect of precision teaching framework on mathematical ability of students with intellectual and developmental disabilities [20]. After more than 50 years of development, traditional precision teaching has formed a complete set of theoretical methods, mainly focusing on evaluating precision teaching effects through teaching experiments. Therefore, precision teaching has been applied in teaching in many countries since its birth, and has become a reference framework for evaluating the effectiveness of other disciplines and teaching methods [21].

In order to provide a more effective teaching method for university programming courses, this research constructs a programming teaching classroom with the integration of precision teaching theory. Therefore, this study mainly conducts in-depth research on the following three issues.

1.What are the realization conditions of precision teaching classroom for university programming courses?

2.What is the process of precision teaching in university programming courses?

3.What is the teaching effect of constructed precision programming teaching classroom?

The rest of the paper is organized as follows: Section II describes the current challenges in teaching programming courses. Section III explains the realization conditions of precision teaching classroom for university programming courses. Section IV details the teaching process, and Section V shows a teaching practice case based on constructed precision teaching classroom. Section VI further discusses the practice results and teaching effects of this case, and conclusions and future work are described in Section VII.

II. CHALLENGES IN TEACHING PROGRAMMING COURSES

A. SELECTING APPROPRIATE METHODS FOR TEACHING PROGRAMMING

Although there are many methods in programming teaching, such as learning by doing and learning by examples, problem-based learning, active learning and demonstrations, collaborative learning, peer instruction and pair programming, however, the application results of these methods are not yet conclusive [22]. The study has reported on the application of specific teaching methods in programming courses and made many interesting changes, including adding positive exercises, demonstrating how code features work, arranging small programming homework exercises and large programming projects, and other opportunities to develop students' valid thinking models. Like many institutions, in fact, some of these activities seem to do as much harm as help, and the dropout rate of programming courses is still high and students' performance is poor [23].

Therefore, the selection of programming teaching methods is important and challenged for teachers [24]. Current programming teaching methods should be rethought, and teachers are encouraged to provide students with programming learning experience that more closely matches their expectations. It is necessary to optimize and adjust teaching methods to improve students' problem-solving ability, which is one of the most necessary skills for them to learn programming [25].

B. RECORDING TEACHER-STUDENT INTERACTION AND FEEDBACK IN REAL TIME

The way to provide teacher-student interaction and feedback is a decisive factor that affects students' learning enthusiasm [26]. This kind of feedback is not only judgment, but also to get student's learning status, and utilize the feedback to improve teaching and learning [27]. In addition to its inherent complexity, the feedback process is also affected by factors such as time, number of students, and course format. Some study has shown that there are too many students in a class and lack of harmonious communication between students leads to failure of teacher-student communication and feedback [28]. Good programming teaching should focus on students' learning process and effective communication between teacher and students. How to record teacher-student interactions and give timely and appropriate feedback is another challenge in programming teaching.

C. KEEPING STUDENTS' MOTIVATION, ENGAGEMENT AND PERSISTENCE

In teaching programming courses, it is fundamental to maintain students' enthusiasm, participation and persistence in learning [27]. However, actual situation is that teachers cannot grasp the changes in learning status and abilities of each student, and programming problems encountered by students cannot be discovered and resolved by teachers in time. Because teaching and learning cannot be synchronized and matched, teachers cannot give differentiated and personalized guidance and help. Over time, students find it difficult to learn programming courses, which will affect their learning mood and enthusiasm. Teachers should become the motivator for students to learn programming, not just the provider of programming knowledge [28], which is also a challenge for teachers in programming teaching

D. EVALUATING LEARNING EFFECTS OF PROGRAMMING COURSE COMPREHENSIVELY

In programming teaching, teachers always strive to develop students' problem-solving ability. However, due to the differences in students' knowledge level, Computational Thinking and learning style, it is difficult for teachers to find an evaluation method suitable for each student. If only programming exams are used to evaluate learning effects of programming course, this just reflects the achievement degree of cognitive and application goals of programming knowledge, while ignoring to evaluate students' learning programming process and programming ability development. How to evaluate learning effects in multiple dimensions is a big challenge in programming teaching evaluation at present.

III. REALIZATION CONDITIONS OF PRECISION TEACHING CLASSROOM FOR UNIVERSITY PROGRAMMING COURSES

A. PRECISION TEACHING THEORY

Precision Teaching is a teaching method proposed by Lindsley in the 1960s based on Skinner's behavioral learning theory [14]. Precision teaching was first used in children's rehabilitation centers in special education, and then gradually used to improve students' basic skills [29]. In the early 1970s, precision teaching was applied in primary school classrooms, mainly through the study of primary school students' learning performance to make data decision [30]. Precision teaching is also essentially a teaching method to monitor students' acquisition of basic educational skills. By recording and measuring students' learning data to monitor students' learning process and learning results, teachers adjust teaching decisions and implement interventions according to feedback information to ensure students' best learning performance [31]. In recent decades, precision teaching has been applied in teaching in many countries, and the research results in the field of precision teaching have gradually developed into a reference framework for evaluating the effectiveness of teaching methods in other disciplines and teaching stages [32]. Precision teaching can stimulate students' learning motivation, make them more engaged in learning, improve their academic performance [33], [34], and become an effective way to strengthen students' acquisition of professional skills [35], [36].

B. TEACHING SUPPORTING ENVIRONMENT

It is argued that the following three teaching environment are supportive conditions for construction of precision teaching classroom for university programming courses: offline teaching classroom based on smart teaching tools, online teaching platform supporting mobile ubiquitous learning, and programming practice environment with automatic assessment, see Figure 1.

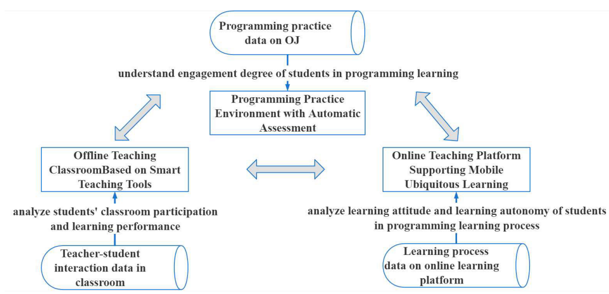


FIGURE 1. Teaching supporting environment of precision teaching classroom for university programming courses.

Offline teaching is "main battlefield" of programming teaching. Smart teaching tools can automatically record teacher-student interactions, such as student attendance, class-room tests, group discussions, etc., so as to help teacher to analyze students' classroom participation and learning performance in offline teaching classroom.

Online teaching is an important way to expand teaching space of programming courses, which allows teacher to upload programming learning resources, assign autonomous learning tasks on the online platform and students to learn programming any-time and anywhere. This platform can automatically record such data as login frequency, learning time, and number of tasks completed, etc., thus facilitating teacher to analyze learning attitude and learning autonomy of students in programming learning process.

Online Judge (OJ) is designed for reliable evaluation of codes submitted by users [37], which is chosen as programming practice environment for precision teaching. OJ features real-time recording and automatically assessment of programming process and programming assignments, including platform login time, code submission times, code accuracy rate, code error types and code scores. Teacher can verify the correctness of solutions submitted by students with higher accuracy and shorter evaluation time, and students receive almost instant feedback as to whether their codes are correct. The programming practice environment based on OJ can provide sufficient data support for teacher to understand engagement degree of students in programming learning.

IV. PROCESS OF PRECISION TEACHING FOR UNIVERSITY PROGRAMMING COURSES

This study proposes that teaching process of precision teaching classroom for university programming courses consists of three phases, namely, preparation phase driven by student data, implementation phase driven by learning process data and evaluation phase driven by multivariate evaluation data. There are five teaching links in this teaching process: precise student analysis, precision teaching design, precision teaching activities, precision teaching decisions and interventions, and precision teaching evaluation, as illustrated in Figure 2.

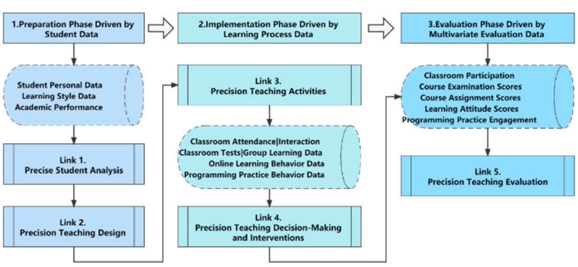


FIGURE 2. Process of precision teaching classroom for university programming courses.

A. PREPARATION PHASE DRIVEN BY STUDENT DATA

Precision teaching preparation phase involves two teaching links of precise student analysis and precision teaching design.

1) PRECISE STUDENT ANALYSIS

R.M. Gagne, an expert in teaching design, put forward the statement of Learner's Characters in his book Principles of Teaching Design. He believed that learner's characteristics include cognition and metacognition, motivation and emotion, development and sociality, and individual differences [38]. Student analysis is an analysis of all students' situations in a broad sense, and an analysis of students' learning in a narrow sense. Professor Smith proposed the concept of Analyze Learners, believing that student analysis was to analyze the similarities, differences, stability and variability of learners in four dimensions of cognition, physiology, emotion and society [39]. As for the function of student analysis, different researchers have different introductions. Generally speaking, there are three views: improving teaching [40], [41], promoting students' learning [42], and promoting teachers' professional development [43].

A comprehensive understanding of students' basic characteristic and learning style is quite necessary before programming teaching, which is conducive to formulating precise teaching objectives, selecting appropriate teaching contents and designing meticulous teaching activities in the link of precision teaching design. Precise student analysis proposed in this paper refers to multi-dimensional, fine-grained and accurate analysis of students' basic learning situation. Precise student analysis in precision teaching preparation phase makes a detailed and comprehensive analysis of the characteristics of students' demographic attributes, learning styles, academic achievements of previous courses and professional qualities from four dimensions. The purpose of the analysis is to help teachers quickly understand the characteristics of students before teaching activities. Precision student analysis proposed in this paper is different from other student analysis of learning situation, that is, it not only includes the analysis of students' basic attributes and learning style characteristics, but also includes the analysis of students' professional quality and previous learning achievements, which is particularly important and helpful for teachers to facilitate the comparison of students' programming ability improvement after precise programming teaching.

This paper analyzes the data in the preparation stage to build a student profile model to accurately describe students' learning situation. Data of preparation phase derive mainly from personal information, learning style, academic performance of students, and so on. Based on these data, this study builds student profile model (SPM, see Figure 3) in four dimensions: personal characteristics, learning style, previous academic performance and level of Computational Thinking. SPM is very helpful for teacher to quickly understand basic information of students and obtain their learning style characteristics and academic performance in previous courses. This provides the basis for the accurate student grouping.

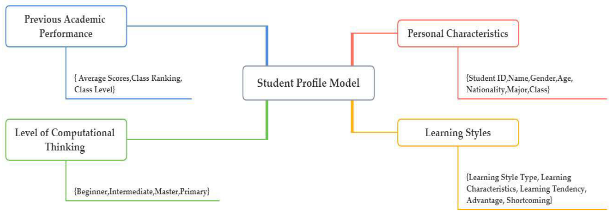


FIGURE 3. **Student profile model (SPM) for precise student analysis.**

2) PRECISION TEACHING DESIGN

Course information is combined with course knowledge map to establish mapping binary relationship between SPM and course contents, thus further refining the knowledge and ability objectives of programming course. Afterward, according to the set teaching objectives, highly matched teaching contents and teaching resources are selected to develop a detailed teaching calendar. Based on the differences in characteristics of SPM, homogenized grouping and heterogeneous grouping are conducted to design common and differentiated teaching activities as well as supporting teaching activities including classroom tests, group discussion, programming exercises, collaborative programming tasks and so on.

Precise student analysis and precision teaching design are driven by the data collected in teaching preparation phase. Its ''precision'' is embodied in the high matching level of teaching objectives with individual characteristics, learning style and learning needs of students and in the detailed and quantifiable descriptions of knowledge and skill objectives for programming courses. All these provide sufficient data evidence and an implementation basis for designing precision teaching activities.

B. IMPLEMENTATION PHASE DRIVEN BY LEARNING PROCESS DATA

Precision teaching implementation phase involves two teaching links, namely carrying out precision teaching activities, adjusting teaching decisions and implementing precise teaching interventions.

1) PRECISION TEACHING ACTIVITIES

According to the supporting conditions of precision programming teaching, precision teaching activities in this research mainly occur in three teaching scenarios of ''offline classroom teaching based on smart teaching tools'', ''online autonomous learning based on mobile ubiquitous learning platforms'', and ''programming practice based on online judge platforms''.

Specifically, the first teaching scenario refers that teacher imparts programming knowledge face-to-face in classroom and use smart teaching tools to conduct teaching activities including automatic classroom sign-in, real-time classroom tests and group discussion. The second teaching scenario means that teacher carries out teaching activities such as publishing course resources, pre-class learning tasks and course-themed discussions on online learning platforms. The third teaching scenario is that teacher assigns programming practice tasks on OJ, where students log in to submit program source codes. Then OJ evaluates the submitted source codes based on test data designed by teacher in advance to verify the correctness and offers real-time error feedback on the submitted code.

Programming learning process data are from these three scenarios of precision teaching activities. The data generated in the first teaching scenario is recorded by smart teaching tools, such as classroom attendance, correct rate of tests, test rankings, times of classroom interaction and mutual evaluation in group learning, and so on. The data generated in the second teaching scenario is autonomous learning behavior data of students automatically recorded by online learning platforms, including login times, learning time and discussion times. The data generated in the third teaching scenario is progress data and result data of programming practice recorded on OJ, such as code submission times, error types, number of completed programming assignments, and correct rate of programming codes and rankings of programming exercise score.

Programming learning profile model (PLPM) is constructed through analysis of these data, as shown in Figure 4. PLPM provides teacher with real-time and visual presentation of students' learning attitude, learning engagement and learning performance in learning programming. And this profile helps teacher to adjust teaching decisions in due time, and precisely identify learning status and potential problems of individual students, thus taking targeted intervention measures.

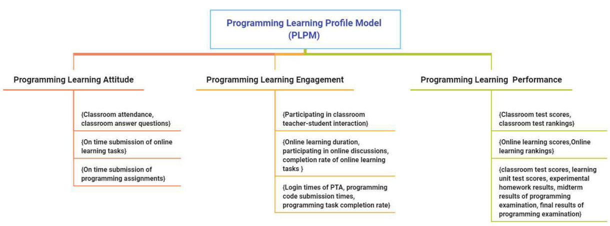


FIGURE 4. **Programming learning profile model (PLPM).**

2) PRECISION TEACHING DECISION-MAKING AND TEACHING INTERVENTIONS

Data-driven teaching decision-making ability refers to the comprehensive ability of teacher to systematically collect, process and analyze data from multiple sources, and use these data as evidence to intervene in the learning process and optimize students' learning effectiveness [44]. Teaching decision-making ability based on data is the core component of teachers' data literacy. Many countries have evaluated the level of Data-driven teaching decision-making in the evaluation practice of teachers' data literacy. For example, Dutch Ministry of Education takes teachers' Data-driven decision-making ability as the main content of evaluating teachers' data literacy [45]. Australia regards teachers' Data-driven teaching decision-making ability as one of the core knowledge and skills of their performance assessment [46].

Teaching decision-making referred to here are decisions made during the teaching implementation process and interaction stage based on teaching data. The problem of empirical teaching decision-making is lack of decision-making power to accurately judge the teaching process, and it is easy to fall into common sense misunderstandings. PLPM proposed in this paper depicts students' learning behavior, learning attitude and learning performance in the process of learning programming, so that teacher can timely pay attention to students' learning needs and learning status at different learning stages.

According to the data service provided by PLPM, it provides support for teacher to adjust teaching decisions in a timely manner, for example, adjust the teaching progress, the difficulty of teaching content and collaborative learning methods, etc. At the same time, based on PLPM, teacher can find groups of students with different learning attitudes and different learning investment, especially for students who are not active enough and not engaged enough, teacher will take corresponding teaching intervention measures in a targeted manner, and give these students personalized learning guidance to help them gain confidence in their learning and encourage them to stay motivated to study hard on programming courses.

C. EVALUATION PHASE DRIVEN BY MULTIVARIATE EVALUATION DATA

Precision teaching evaluation phase targets to evaluate learning attitudes, classroom participation, programming practice engagement, and learning outcomes of students throughout the whole programming teaching process. Learning attitudes are to evaluate students' subjective emotions and their intentions to continuously learn programming. Classroom participation is to evaluate students' subjective initiative of participating in classroom programming teaching activities. Programming practice engagement is to evaluate students' time investment and task completion on OJ during their programming practice. Learning outcomes are to evaluate the abilities of students to solve programming problems with relevant programming knowledge, algorithms, tools, and methods.

To sum up, no matter at which stage, timely summary and reflection are an important driving force to promote the effective operation of precision teaching of programming courses. Teaching design is based on student analysis, and programming teaching activities are carried out in an orderly manner based on the teaching design. The real-time feedback of classroom teaching, online autonomous learning and programming practice links prompts teacher to adjust teaching strategies in a timely manner, and accurately intervene in the individual programming learning process of students.

V. PRECISION TEACHING PRACTICE OF C LANGUAGE PROGRAMMING COURSE

A. TEACHING CALENDAR

According to the precise teaching process of university programming courses proposed in Part IV, teaching calendar of C Language Programming course is shown in Table 1.

TABLE 1. Teaching calendar of C language programming course.

Teaching Weeks	Teaching Theme	Precision Teaching Arrangement
one week before class starts	–	1.Build up student profile 2.Set teaching contents and learning goals 3.Design teaching activities based on student profile
week 1 to week 4	Basic Knowledge of C Language	1.Build up programming learning profile. 2.Implement Precision teaching activities. 3.Precision teaching intervention (class overall intervention, learning group intervention and student individual intervention).
week 5 to week 8	Basic Structure of Programing	
week 9 to week 10	Array	
week 11 to week 14	Function and Pointer	
week 15 to week 16	Structure	
two weeks after the course	–	1.Course examination 2.Precision teaching evaluation

Teaching week of C Language Programming Course is 16 weeks, with three hours per week. Before the course started, teacher obtained students' basic information from educational administration system, student management system, etc., including student number of class, proportion of boys and girls, students' previous academic performance and other data. Based on student profile model (SPM) constructed in Figure 3, student profile will be created and be used in setting learning goals, establishing study groups and designing differentiated teaching activities.

Teaching theme of this course includes five modules, namely, basic knowledge of C language (week 1 to week 4), basic structure of programing (week 5 to week 8), array (week 9 to week 10), function and pointer (week 11 to week 14), and structure (week 15 to week 16). In the teaching process of these five learning topics, student programming learning profiles were generated based on SPM (refer to Figure 3). As students' learning attitude, learning behavior and learning performance at each learning stage were dynamic, these learning profiles were updated constantly, and this effectively helped teachers to adjust teaching strategies and implement teaching intervention in a timely manner.

Teaching interventions were tailored to the individual differences and learning characteristics of students in the learning process, guiding them to improve their learning methods, encouraging them to increase their learning confidence, and providing targeted guidance and assistance to facilitate the continuous optimization of their subsequent learning behavior. Precision teaching intervention mentioned in this paper is a targeted ''one to many'' or ''one to one'' intervention measures taken by the teacher according to learning emotions and learning behavior characteristics of individual students and class groups in the learning process, and to help students adjust their learning strategies and promote personalized learning.

The types of precision teaching intervention include class overall intervention, learning group intervention and student individual intervention. According to the students' learning behavior and learning performance reflected in the learning profile at this stage, class overall intervention usually took place in the unit tests at the end of a certain teaching theme, such as explanation of core knowledge points and test questions with high error rate, etc. In the teaching process, teacher carried out group learning intervention on students with same or similar learning style, learning performance and learning behavior characteristics. For example, for students with good learning attitude and sufficient learning input, teacher appropriately increased the amount and difficulty of homework. Teacher would adjust teaching strategies of students with poor learning performance, such as last ten in the class, to reduce the difficulty of course assignments. Teacher also accurately analyzed students' learning performance based on their programming learning profile, found their learning characteristics at the current stage, so that teacher gave timely and point-to-point intervention and guidance. About two weeks after the end of the course, students of the same grade who study the same course had a unified final exam and a course teaching evaluation.

B. PARTICIPANTS

This research conducted precision teaching practice of C Language Programming course among first-year undergraduate students majoring in computer science and technology in a Chinese university from September 2020 to January 2021. In this teaching practice, the first author served as the teacher of C Language Programming course for two classes (A and B) at the same time. This study selected class A (48 people) as experimental class and adopted proposed precision teaching classroom for C Language Programming course's teaching. Class B (55 people) was control class, which adopted ordinary teaching method for this course. Finally, teaching effects of the experimental class and the control class are analyzed and compared in detail.

C. SUPPORTING ENVIRONMENT OF PRECISION TEACHING CLASSROOM FOR C LANGUAGE PROGRAMMING COURSE

This research selected ''Teachermate'', ''Xuexitong'' and ''Programming Teaching Assistance (PTA)'' as supporting environment of precision teaching classroom for C Language Programming course.

Teachermate is a smart teaching tool supporting classroom teaching, which has been widely used in Chinese university teaching because of its strong classroom interactivity [47]. It can automatically record data of teacher-student classroom interaction such as attendance data, group discussions data, and classroom tests data. In addition, Teachermate can count number of each student's participation in classroom interactions, and the score and class rankings of each quiz. Teachermate provides data basis for evaluating students' learning attitudes and classroom participation.

Xuexitong is widely used in online teaching in Chinese universities. It is based on micro-service architecture, which integrates course learning, knowledge dissemination, and knowledge sharing [48]. It not only allows teachers to release learning resources, learning tasks, course discussions, and other online learning tasks but also automatically records online autonomous learning behavior data of students. Xuexitong offer data evidence for analyzing students' learning attitudes.

PTA is an auxiliary teaching platform for programming experiments, which supporting online automatic judgment of codes [49]. It can record process data of programming practice, such as login times, code submission times, correct submission times, error types, and correct rate of submitted codes. These data provide data supports for evaluating students' programming practice engagement and programming learning outcomes in C Language Programming course.

D. PRACTICE PROCESS OF PRECISION TEACHING CLASSROOM FOR C LANGUAGE PROGRAMMING COURSE

This section illustrates specific practice process in C Language Programming course according to precision teaching classroom for programming course proposed in this re-search.

1) PRECISE STUDENT ANALYSIS

In the first class of C Language Programming course, with students' consent, they were asked to fill in basic information form and learning style scale. The basic information form collected such main data as student ID, gender, age, ethnicity, academic performance, and academic performance rankings of students.

This study took Kolb learning style scale to collect students' learning style data. Kolb model theory is the detection of different combinations of learners in the four-step experimental learning process, which is closer to formal learning process with certain learning goals [50]. Kolb scale in this study contained 12 test questions, and adopted a five-point Likert scale from 1 (least like me) to 5 (most like me). Then, based on the results of this scale, learning style of Class A were divided the into four types: diverging, assimilating, converging, and accommodating. The results of learning style classification were helpful for teacher to build heterogeneous learning groups in programming teaching activities.

In this study, student profiles of Class A were established based on collected student information. Take student DQ as an example, her profile is shown in Figure 5. It can be seen from DQ's profile that her previous academic performance was good, ranking about 12th in the class. DQ's learning style was diverging, and she was characterized by concrete experience and reflective observation in learning. People with this learning style are good at observing specific situations from multiple angles, like group activities, and per-form well in learning activities such as "brainstorming". Therefore, DQ was selected as a group leader in group studying and led group members to complete collaborative programming tasks.

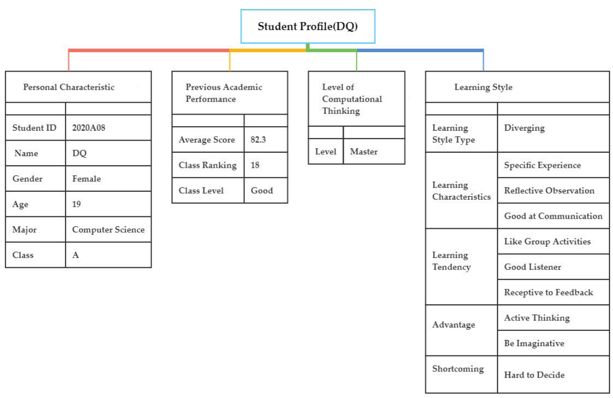


FIGURE 5. Student profile of DQ.

2) PRECISION TEACHING DESIGN

Take teaching theme of "One-Dimensional Array" in C Language Programming course as an example to illustrate how to set precision teaching objectives. In ordinary programming teaching, learning objective of this theme is "being able to define and use One-Dimensional Arrays proficiently". But this objective not details enough for a quantitative analysis of how it is achieved and evaluated. In precision programming teaching, learning objectives of One-Dimensional Array thoroughly are quantitatively described and explained from the perspectives of knowledge and skills. Therefore, learning objectives of One-Dimensional Array teaching topic are redefined as three sub-objectives: 1) Learning Outcome 1, being able to define One-Dimensional Array of int type, float type and char type; 2) Learning Outcome 2, being able to use "for-loop", "while-loop", and "do-while loop" to access these three types of arrays, and 3) Learning Outcome 3, being able to use One-Dimensional Array to solve practical problems.

According to Bloom's Taxonomy [51], LO1 is at "Remember and Understand" level, that is, students are able to define One-Dimensional Array of three data types. LO2 is in "Apply "level, that is, students are able to use loop structure to access One-Dimensional Array, such as initialize and output the elements of One-Dimensional Array. LO3 is in "Analyze" level, that means students can use One-Dimensional Arrays and loop structure to design algorithm and write codes to solve practical problems like sorting and searching.

3) PRECISION TEACHING ACTIVITIES

The process of precision teaching activities is first to conduct a pre-class test of prior knowledge before a new class, and determine whether to start new teaching theme according to students' performance. Then, teacher began to explain the programming knowledge, demonstrate programming process, and assign programming practice tasks, see Figure 6 for details.

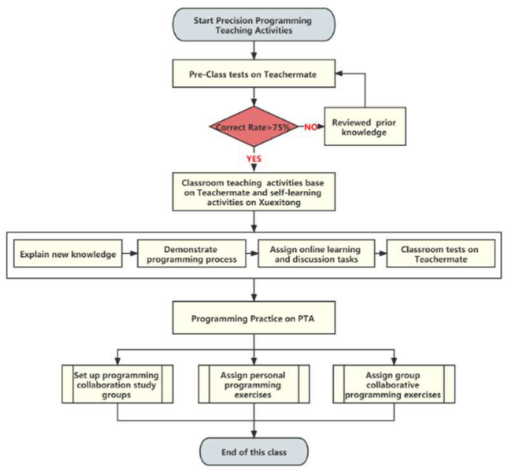


FIGURE 6. The process of precision programming teaching activities.

This part still takes teaching theme of "One-Dimensional Array" of C Language Programming course as an example. Before the new lesson, students should be tested on Teachermate about grammar knowledge and loop structure learned previously. Teachermate could automatically calculate correct rate for each question, which helped teacher to judge the extent to which students grasp previously learned knowledge. If the correct rate was lower than 75%, teacher could not start teaching new lesson immediately, and need to review the prior knowledge points, explain test questions, and conducted the pre-class test again until more than 75% of the students passed the tests. Teacher identified that students had mastered prior knowledge to learn the new lesson, and then started new teaching contents.

Teacher explained key knowledge of "One-Dimensional Array" and demonstrated programming process targeting all students, also arranged online autonomous learning and discussion tasks on Xuexitong, and then assigned classroom tests on Teachrmate. Students were required to complete these tasks within the given time. Teacher checked the scores of students and their rankings in real time on Teachermate. Table 2 list the correct rate of one classroom test of "One-Dimensional Array definition".

Table 2 shows that Class A obtained correct rate of over 90% in the test of defining three data types of One-Dimensional Array, which revealed that students adequately grasped the definition methods of One-Dimensional Array. Moreover, about 25% of the students failed to correctly

TABLE 2. Correct rate of classroom test of "one-dimensional array definition."

Definition of One-Dimensional Integer Array	Definition of One-Dimensional Float Array	Definition of One-Dimensional Character Array	Input and Output One-Dimensional Array by Using Loop	Ranking of One-Dimensional Array Elements
94.8%	92.3%	90.6%	73.4%	63.7%

answer the questions about using a loop structure to input and output One-Dimensional Array, suggesting that these students did not master the method of inputting and outputting One-Dimensional Array elements, so the teacher need to explain and demonstrate the solution of this problem again. Regarding the question of ranking the elements of One-Dimensional Array, only 63.7% of the students submitted correct answers, which indicated that one-third of the students in Class A had poor analytical and design skills to solve practical problems with One-Dimensional Array. There-fore, teacher had to explain the programming method of this question and increased the relevant exercises in programming practice.

First step of teaching activities of programming was to set up a heterogeneous programming study group. Teacher divided classroom test results into five levels: excellent (90 points or more), good (80-90 points), medium (70-80 points), poor (60-70 points), and failed (below 60 points). Class A was composed of 48 students, who were classified into 10 programming collaboration groups according to their classroom test results and student profiles. Teacher selected one student from each level to form a programming collaboration study group, and ensured that there were students with different learning styles in each group. The student with excellent test performance served as the leader of every group to take charge of allocating learning tasks, organizing group discussions and reporting group learning outcomes. Members of every group conducted collaborative programming learning and group discussions under the leadership of group leader.

Then, teacher assigned personal programming exercises and group collaborative programming exercises on PTA. Teacher required students to complete the personal programming exercises independently, and group collaborative programming exercises could be completed collaboratively in study group. If group member's personal programming exercise tasks failed, problems would be solved with the assistance of this group leader. As for collaborative programming exercises, every group leader led the group members to discuss and jointly designed solutions. After the tasks were all completed, each group selected a representative to show the completing process to the teacher, who then provided each group with explanations for different programming problems encountered as well as individual guidance.

E. PRECISION TEACHING DECISION ADJUSTMENT AND TEACHING INTERVENTION

1) ADJUST TEACHING STRATEGIES

During the process of precision programming teaching activities, teacher would adjust teaching contents, teaching progress and study grouping based on programming learning profile of Class A. Taking the data in one-dimensional array teaching activities as an example, data contents and data quantification method as shown in Table 3.

TABLE 3. Learning process data contents and data quantification.

Data Category	Data Contents	Data Quantification Method
Programming Learning Attitude Data	Classroom learning attitude	1 indicates normal check-in, 0.5 indicates late, 0.2 indicates leave, and 0 indicates absent
	Online learning attitude	1 means completing learning tasks on time, 0.5 means partially completing learning tasks and 0 means not completing the learning tasks
	Programming practice attitude	1 means completing programming tasks on time, 0.5 means partially completing programming tasks and 0 means not completing programming tasks
Programming Learning Engagement Data	Participating in classroom teacher-student interaction	1.2 means actively answering questions and correct, 1 means actively answering questions, and 0 means not participating in the interaction
	Online learning duration Completion rate of online learning tasks	Obtain the corresponding value according to the completion rate of online learning tasks
	Login times of PTA Programming code submission times Programming task completion rate	Obtain the corresponding value according to the completion rate of programming tasks
Programming Learning Performance Data	Classroom test scores Learning unit test scores Programming homework scores	Assign value according to actual score of the tests and exams
	Midterm examination scores Final examination scores	Assign value according to class rankings

This study made descriptive analysis of quantified learning attitude data and learning engagement data, and the results are shown in Table 4 and Table 5. The results showed that the distribution of learning attitude data and learning engagement data was reasonable, and cluster analysis could be carried out.

TABLE 4. **Descriptive analysis of learning attitude data.**

Sample Size	Min Value	Max Value	Mean Value	Standard Deviation	Median	Standard Error
48	0	1	0.593	0.196	0.626	0.028

TABLE 5. **Descriptive analysis of learning engagement data.**

Sample Size	Min Value	Max Value	Mean Value	Standard Deviation	Median	Standard Error
48	0	1	0.589	0.209	0.611	0.03

After the above data were quantified and normalized, this study conducted cluster analysis on students' learning attitude and learning engagement in SPSS 25, and obtained four types of learning attitude groups (''Initiative Learner'', ''Active Learner'', ''Passive Learner'', ''Averse Learner'') and four types of learning engagement groups (''very devoted'', ''comparative engagement'', ''less engagement'' and ''insufficient engagement''). Then counted the test scores and class ranking obtained by students in the learning process of One-Dimensional Array teaching theme, and generated DQ's learning profile (Figure 7) by using PowerBI.

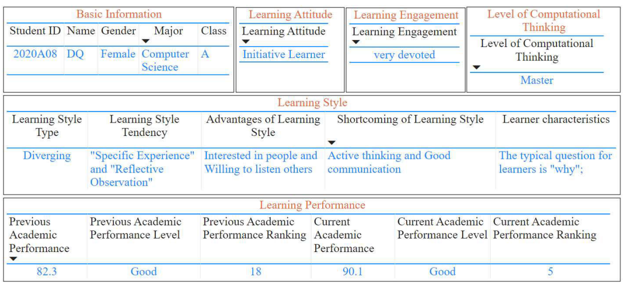
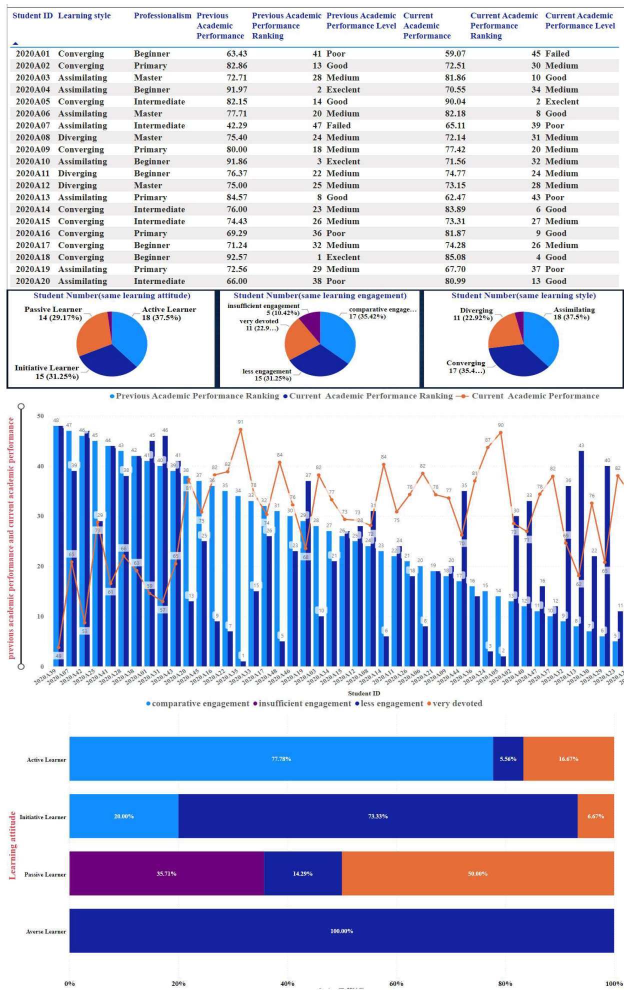


FIGURE 7. **Learning profile of DQ.**

The individual profile of students showed the learning characteristics of each student in current teaching process in the form of short text labels. Teachers can clearly see each student's learning attitude, learning engagement and learning performance, especially the changes of students' academic performance, so as to provide intuitive data evidence for teachers to adjust teaching strategies.

In DQ's learning profile, it was obvious that her academic performance had improved a lot. It was found that DQ was outstanding in learning attitude and she was an initiative learner, and DQ also had a high degree of engagement in learning programming. From a comprehensive point of view, the reason why DQ achieved great achievements and progress was largely due to her proactive learning attitude and serious learning engagement. This was also confirmed in the face-to-face communication with DQ. She said that ''she likes learning C language programming very much. When the code she wrote run smoothly and got the correct results, it makes her feel a great sense of achievement''. Based on DQ's learning profile, teachers' programming teaching guidance and strategies for DQ and learning group where DQ belongs can be changed accordingly.

Programming learning profile of class A after learning one-dimensional array is shown in Figure 8.



Student ID	Learning style	Professionalism	Previous Academic Performance	Previous Academic Performance Ranking	Previous Academic Performance Level	Current Academic Performance	Current Academic Performance Ranking	Current Academic Performance Level
2020A01	Converging	Beginner	63.43	41	Poor	59.07	45	Failed
2020A02	Converging	Primary	82.86	13	Good	72.51	30	Medium
2020A03	Assimilating	Master	72.71	28	Medium	81.86	10	Good
2020A04	Assimilating	Beginner	91.97	2	Execlent	70.55	34	Medium
2020A05	Converging	Intermediate	82.15	14	Good	90.04	2	Execlent
2020A06	Assimilating	Master	77.71	20	Medium	82.18	8	Good
2020A07	Assimilating	Intermediate	42.29	47	Failed	65.11	39	Poor
2020A08	Diverging	Master	75.40	24	Medium	72.14	31	Medium
2020A09	Converging	Primary	80.00	18	Medium	77.42	20	Medium
2020A10	Assimilating	Beginner	91.86	3	Execlent	71.56	32	Medium
2020A11	Diverging	Beginner	76.37	22	Medium	74.77	24	Medium
2020A12	Diverging	Master	75.00	25	Medium	73.15	28	Medium
2020A13	Assimilating	Primary	84.57	8	Good	62.47	43	Poor
2020A14	Converging	Intermediate	76.00	23	Medium	83.89	6	Good
2020A15	Converging	Intermediate	74.43	26	Medium	73.31	27	Medium
2020A16	Converging	Primary	69.29	36	Poor	81.87	9	Good
2020A17	Converging	Beginner	71.24	32	Medium	74.28	26	Medium
2020A18	Converging	Beginner	92.57	1	Execlent	85.08	4	Good
2020A19	Assimilating	Primary	72.56	29	Medium	67.70	37	Poor
2020A20	Assimilating	Intermediate	66.00	38	Poor	80.99	13	Good

FIGURE 8. **Programming learning profile of Class A.**

From the perspective of the ''learning attitude'' dimension of the learning profile, 37.5% of the students in this class had active learning attitude towards C language programming course, and 31.25% of the students were initiative towards learning, which showed that nearly 70% of the students in class 2020A can actively and actively treat the course learning. However, it also can be seen that 29.17% of the students had passive learning attitude, which showed that nearly one-third of the students in class 2020A were relatively passive in their learning at the current stage. This meant that these students only completed learning tasks arranged by teacher according to the requirements and did not actively participate in the teaching activities, which was particularly obvious in the classroom teaching activities. This was a strong signal for the teacher. Students' learning enthusiasm to participate in classroom teaching activities was not very high. Was it because classroom tests were too difficult? Was the distribution of study group members unreasonable? According to the students' learning performance reflected in the learning profile of Class A, the teacher adjusted the difficulty of classroom tests and the members of the learning group accordingly. Since the next new teaching topic was "two-dimensional array", it was more difficult to learn on the basis of "one-dimensional array". Therefore, when starting a new teaching theme, teacher must to redesign teaching activities, especially set more uniform test questions with different difficulty levels.

C Language Programming course requires a lot of effort in in daily homework and online programming practice. From the perspective of "learning engagement" dimension of learning profile, 22.92% of the students in class A were very engaged in learning, 35.42% were relatively engaged and 31.25% were generally engaged in learning. This was a good hint for teacher that about 90% of the students in Class A can invest time and energy in programming learning and practice. However, it also found that students who did not devote enough in learning had shown a certain degree of boredom towards learning. Psychological non-acceptance must lead to behavioral negativity. Therefore, teacher should pay timely attention to the "insufficient engagement" student groups, especially the students who had shown a negative learning attitude. If the proportion of this part was increasing, it meant that current teaching content and teaching difficulty gave certain pressure and burden on students, which would reduce enthusiasm for learning programming. At this time, teachers should re-examine the current teaching strategies and arrangements, whether they were in line with the current learning status of students, and adjusted the teaching plans and teaching content in time.

From the "learning performance" dimension of learning profile, teacher can find more hidden details in teaching. It was difficult for students who learned programming for the first time to master programming methods. Therefore, there would be students with good academic performance, but poor performance in learning programming courses. Many studies on programming learning have shown that poor programming performance can easily lead to students losing learning confidence and giving up learning. Teacher needs to pay special attention to such students and gave timely guidance and help. For example, in group collaborative programming learning, according to the display results of current academic performance, teachers observed the changes of learning performance of students in each group in real time. Specifically, if the learning group achieved relatively good results (such as "excellent level") in quizzes, tests and programming exercises as a whole and generally reflected that they had not encountered too many difficulties in mastering programming knowledge and completing programming tasks, teacher would assign more self-directed learning tasks on Xuexitong and separate exercises with upgraded difficulty to this group. However, if the group members achieved not so good results (such as "poor level") in classroom tests and programming exercises and had no high enthusiasm and felt depressed in learning programming, which indicated that the learning tasks assigned by the teacher were difficult for these students, teacher would give special guidance to them on the assigned exercises instead of assigning other learning tasks.

2) PRECISION TEACHING INTERVENTION

Taking teaching scene of students practicing programming on PTA as an example, this section shows how to conduct teaching interventions in the precision teaching class-room of C Language Programming course. Student DQ's programming exercises data of "One-Dimensional Array" on PTA are in Table 6.

TABLE 6. Programming exercises data of student DQ on PTA.

Contents of Programming Exercises	Corresponding Learning Objectives	Number of Code Submission	Pass /Fail	Correct Rate	Class Ranking
Definition of One-Dimensional Array	LO1	2	Pass	88.2%	5/48
Initialization of One-Dimensional Array	LO2	3	Pass	89.4%	12/48
Ranking of One-Dimensional Array Elements	LO2	6	Fail	56.1%	19/48
Solving Sequence summation problem by Using One-Dimensional Array	LO3	8	Pass	63.7%	22/48
Solving Factorial Problem by Using One-Dimensional Array	LO3	10	Fail	50.8%	38/48

As can be seen in Table 6, student DQ passed programming exercises of "Definition of One-Dimensional Array "with less code submission times, indicating that she had adequately mastered definition method of One-Dimensional Array. However, more code submission times and lower pass rate in design exercises (such as the solution of factorials) reflected that student DQ had insufficient problem decomposition ability and algorithm designing skill. Besides, although Student DQ finally passed the design exercises, her ranking was at lower level of whole class. Hence, after discovering the difficulties student DQ encountered in programming practice, teacher had face-to-face communication with her and provided targeted one-to-one programming counseling in time.

In addition, test rankings of class A on Teachermate, online Learning Rankings on Xuexitong and programming exercises score rankings on PTA reflected that student groups of different programming learning performance and level gradually appeared in Class A. For example, those students always in bottom 10 of class attracted attention from the teacher, who in turn constantly asked about and followed up on their learning emotion and learning progress and actively guided them to solve learning problems encountered in C Language Programming course. After class, teacher would relieve their learning pressure and encourage them to make continuous progress through communication via social media and emails.

F. PRECISION TEACHING EVALUATION

1) CONSTRUCTION METHOD

This research constructs precise teaching evaluation system of C Language Programming course by using Delphi [49] method. Delphi method, also known as expert consultation method, refers to that when evaluating and decomposing a complex problem, researchers cannot ensure that the level of problem decomposition is clear and the logic is correct due to their own limit level and subjectivity. They consult experts in the same field through the preparation of questionnaires to obtain the basis for establishing evaluation indicators. Delphi method has three core steps. The first step is to define the consultation questions and design questionnaires. The second step is to select experts and analyze their authority. The third step is to carry out expert advice consultation, feedback and statistical collation. Generally, at least two rounds of expert consultation are conducted [50].

2) CONSTRUCTION PROCESS

The construction process of precise teaching evaluation system for C programming course includes three steps: preliminary selection of evaluation indicators, optimization of evaluation indicators and determination of evaluation indicators. Precision teaching evaluation of programming courses evaluates the teaching quality and teaching effect based on students' classroom participation data, classroom practice data, programming practice data, and learning outcome data at each stage. Therefore, the teaching evaluation index system established in this paper includes four evaluation dimensions: learning attitude, classroom participation, programming practice engagement and learning outcomes. Each evaluation dimension also contains secondary evaluation indicators.

The preliminary selection of precision teaching evaluation system inevitably had the problem of strong personal subjectivity. Therefore, this paper used Delphi method to test and optimize the primary selection evaluation index. In this study, eleven teaching experts were invited to participate in the expert consultation, including 2 teaching experts, 4 subject experts, and 5 teachers of teaching the same course. They have 10 to 30 years of teaching experience in the field of teaching. Average authority coefficient of experts was 0.846, indicating that the selected experts generally had high authority and could give scientific and objective opinions on the proposed precision teaching evaluation system.

3) CONSTRUCTION RESULTS

Two rounds of expert consultations were carried out, each of which provided the experts with relevant research materials and explained the content and requirements of the expert consultations by sending email. As required, the experts need to complete the corresponding questionnaire. The opinions of experts in the first round focused on the selection and nomenclature of secondary indicators. The second round of expert consultation mainly invited experts to assign the weight of the two-level indicators. Based on two rounds of expert opinions, this study determined the precision teaching evaluation system of programming courses, and gave corresponding weights to first-level index and second-level index in combination. Refer to Table 7 for the details of this evaluation system.

TABLE 7. Precision teaching evaluation system of C language programming course.

Primary Indicators	Weights	Seconary Indicators	Weights
Learning Attitude	0.12	Classroom Attendance	0.23
		Completion Rate of Online Learning Tasks	0.22
		Punctuality of Assignment Submissions	0.25
		Completion Rate of Programming Exercise	0.31
Classroom Participation	0.13	Classroom Interactions	0.24
		Classroom Discussions	0.20
		Correct Rate of Classroom Tests	0.56
Programming Practice Engagement	0.21	PTA Login Times	0.15
		Code Submission Rate	0.24
		Code Submission Accuracy	0.61
Learning Outcomes	0.54	Classroom Test Scores	0.13
		Online Learning Scores	0.10
		Assignment Scores	0.11
		Mid-Term Exam Scores	0.27
		Final Exam Scores	0.39

C Language Programming course teaching evaluation system includes 4 primary indicators and 17 secondary indicators. Four primary indicators are learning attitude, classroom participation, programming practice engagement and learning outcomes, with weights of 0.12, 0.13, 0.21, and 0.54 respectively.

Learning attitude dimension, which account for 12% of the total evaluation, includes four secondary indicators, the content and weight of each indicator are Classroom Attendance (0.23), Completion Rate of Online Learning Tasks (0.22), Punctuality of Assignment Submissions(0.25), and Completion Rate of Programming Exercise(0.31).This dimension evaluates students' emotions and attitudes in learning C Language Programming course to clarify whether they were confident and active in programming learning and whether they were satisfied with the precision teaching design and teaching activities.

In terms of classroom participation dimension, accounting for 13% of the total evaluation, is evaluated from three dimensions, namely classroom interactions (0.24), class discussions (0.20) and correct rate of answering questions (0.56). Classroom participation evaluate students' enthusiasm and initiative to participate in classroom teaching activities by quantities of classroom teacher-student interaction, times of group discussion and completion quality of classroom tests.

As for programming practice engagement dimension, occupying 21% of the total evaluation, includes three dimensions: PTA Login Times (0.15), Code Submission Rate (0.24), and Code Submission Accuracy (0.61). This dimension evaluates the degree of students' investment in programming practice, and determine whether students often log into PTA for programming exercises, time spent on programming exercises, and the accuracy of submitted programming codes.

Learning outcomes dimension, occupying 54% (the highest proportion) of the total evaluation, consists of five dimensions, namely Classroom Test Scores (0.13), Online Learning Scores (0.10), Assignment Scores (0.15), Mid-Term Exam Scores (0.25), Final Exam Scores (0.37). This dimension evaluates learning outcomes of programming knowledge of students by tests, assignments and examination scores, as well as assesses their development of programming abilities, such as problem understanding, algorithm design, code writing and debugging skills.

VI. DISCUSSION AND ANALYSIS

A. DISCUSSION OF LEARNING EFFECTS OF PRECISION TEACHING CLASSROOM FOR C LANGUAGE PROGRAMMING COURSE

This research discusses difference in learning effects of C Language Programming course between precision teaching classroom and ordinary teaching classroom by analyzing final programming examination scores at the end of the semester. This study carried out independent sample t-test on final programming examination scores of Class A and Class B. Table 8 shows the means and standard deviations of final programming examination scores.

TABLE 8. Means and standard deviations of final programming examination scores of Class A and Class B.

Class	N	Mean	Standard Deviation	Standard Error of Mean
Class A	48	73.55	14.13	2.64
Class B	55	67.84	15.29	3.47

In Table 8, the results reveal significant differences in final programming exam scores of two classes at 0.03. The average score of Class A was 73.55, higher than that of Class B, showing that the overall learning performance of programming practice of experimental class were better than those of control class. The standard error of mean of Class B was relatively large, which showed that programming skills gap between the students in control class was larger than that between the students in experimental class.

This research conducted a Null Hypothesis Significance Testing (NHST) and calculated the p-value of the distribution of the individuals' data to be sure that the null hypothesis can be ruled out in this teaching experiment study presented. Hypothesis testing of this research involved following three steps.

Step 1. Established null hypothesis and determine test level. The Null Hypothesis (H_0) selected in this study was that there was no difference in the teaching effect between the experimental class and the control class, that is, the teaching effect of the programming course using the precision teaching mode was the same as that under the traditional teaching mode. The Alternative Hypothesis(H_1) was that the teaching effects of the experimental class and the control class were different.

$$H_0 : \mu_1 = \mu_2, \quad H1 : \mu 1 \neq \mu 2$$

μ_1 and μ_2 represents the average value of the experimental class and the control class. In this study, the value of the size of a test(a) was determined as 0.05.

Step 2. Selected test methods and calculated test statistics. The known information of this study was the mean, standard deviation and sample size of the experimental class and the control class. Since the number of samples was more than 30, this study used the Z-test method to test hypotheses. Calculated the z fraction according to Formula 1.

$$z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{1}$$

$\bar{X}_1$ and $\bar{X}_2$ were the average of the experimental class and the control class. s_1 and s_2 were the standard deviations of the experimental class and the control class. n1 and n2 were the sample sizes of the experimental class and the control class. Based on the Formula 1, the value of Z was 1.98.

Step 3. Determined P value and make statistical inference conclusion. Queried the relationship table between z value and p value and found that $| z | \geqslant 1.96$, then p value $\leqslant 0.05$. According to the set size of test (a), H_0 was rejected and H1 was accepted. That meant, the hypothesis that there was no difference between the teaching effect of the programming course of the experimental class and the control class was not tenable, and the teaching effect of the two classes was different. Figure 9 compares final programming examination scores of Class A and Class B.

In Figure 9, scores of 90-100 mean "excellent," scores of 80-89 represent "good," scores of 70-79 stand for "medium," scores of 60-69 refer to "poor," and scores below 60 mean "failed." The "excellent" rate of Class was 13.7%, nearly 6% higher than that of Class B at 7.6%. The "failed" rate of Class A was about 10%, 8% lower than that of Class B at 18%. Furthermore, average score of Class A was increased by nearly 6 points and standard deviation was decreased, reflecting more uniform scores of students and improved polarization in the precision teaching classroom.

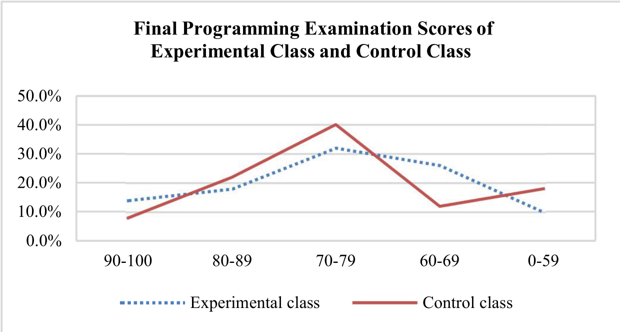


FIGURE 9. Comparison of final programming examination scores between Class A and Class B.

Achievement degree of learning goals of each teaching theme of Class A and Class B is compared in Figure 10.

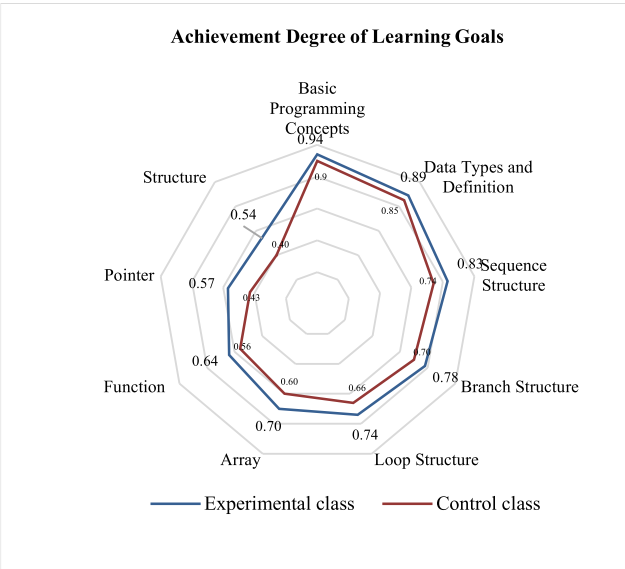


FIGURE 10. Learning goals achievement degree of experimental class and control class.

Teaching themes of Basic Programming Knowledge, Data Types and Definition and Program Structures of Sequence, Branch and Loop belong to basic contents of C Language Programming course, and most programming exercises in these parts were confirmatory. As can be seen from Figure 10, achievement of learning goals of class A on these teaching topics was higher than that of class B, which showed that precision teaching of C Language Programming course effectively improved students' understanding of conceptual programming knowledge by means of pre-class tests, classroom tests, group learning, and so on.

Teaching themes of Array, Function, Pointer and Structure are high-level contents of C Language Programming course. The programming exercises of these themes were all to design programs, which targeted in cultivating students' high-level ability to use programming knowledge to solve problems. In Figure 10, learning goals achievement degree of these teaching themes of Class A is about 10% higher than that of Class B. This is because teacher paid more attention to students' programming practice process and gave them real-time teaching intervention in C Language Programming precision teaching activities. Differentiated programming guidance given by teacher and group cooperative programming learning were very helpful to these students, especially who encountered programming problems that were difficult to solve, and they constantly tried to practice and solved the problems finally. Therefore, programming abilities of students in Class A had also been improved through continuous practice and attempts.

B. ANALYSIS OF LEARNING SATISFACTION WITH PRECISION TEACHING CLASSROOM FOR C LANGUAGE PROGRAMMING COURSE

The research designs a questionnaire on learning satisfaction with the precision teaching classroom for C Language Programming course to analyze students' learning satisfaction. At the end of the course, teacher distributed a total of 48 questionnaires to students of Class A and collected 48 valid questionnaires. The questionnaire includes five dimensions, namely satisfaction of instructional design, learning participation, achievement of learning goals, course evaluation system and programming skills improvement.

The study ran a confirmatory factor analysis (CFA) on learning satisfaction survey data. CFA is a type of structural equation modeling used in situations where the dimensionality of the variables is known. Before running a CFA, this study determined certain assumptions, such as linearity, independence, normality, and no outliers, through an examination of correlations, scatter plots and histograms in SPSS 25. Then this work determined fit by using traditional fit statistics (x^2/df = 1.85, RMSEA = 0.06, CFI = 0.98, TLI = 0.97, SRMR = 0.03). This questionnaire yielded a standard Cronbach's α coefficient of 0.885, indicating high reliability and validity. The study investigated satisfaction of instructional design in five aspects: teaching contents, learning resources, learning objectives, classroom exercises, and learning activities. The statistical results of each dimension are shown in Figure 11-13.

In Figure 11, more than 52% of the students were quite satisfied with the precision teaching design of the whole course and 32.68% were relatively satisfied, indicating that majority of the class was satisfied with the teaching design of C Language Programming course. In aspect of satisfaction of learning participation, 38.24% students of Class A reported that teacher-student interactions in this course were quite helpful, and 33.33% believed it was relatively helpful. Generally speaking, over half of the students thought that class-room atmosphere of C Language Programming course was active and teacher's guidance was targeted. Classmates' mutual evaluation contributed to the high classroom participation.

In terms of achievement of learning goals, see Figure 12, 90% of the students expressed that they achieved cognitive

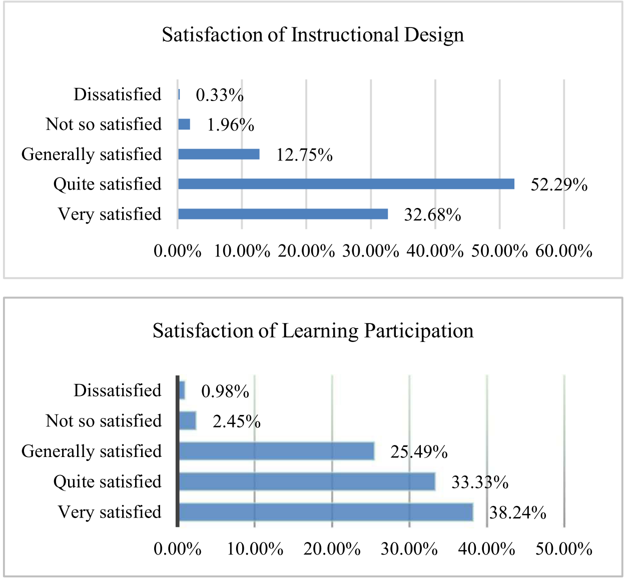


FIGURE 11. **Satisfaction of instructional design and learning participation.**

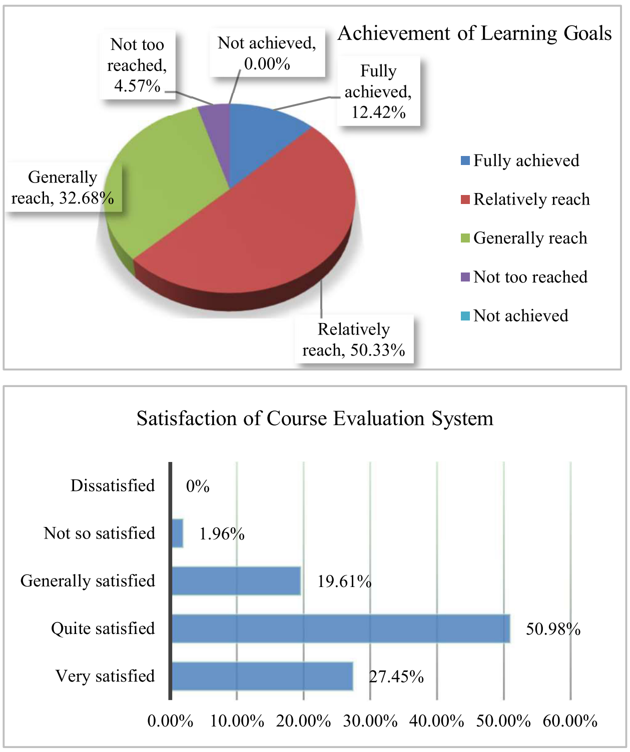


FIGURE 12. **Achievement of learning goals and satisfaction of course evaluation system.**

objectives and content objectives of the course in this class. Additionally, this study investigated satisfaction of course evaluation system in four aspects, namely evaluation subjects, evaluation forms, evaluation contents, and evaluation perspectives. Over 77% of students were relatively satisfied with the evaluation method of this course, demonstrating that most

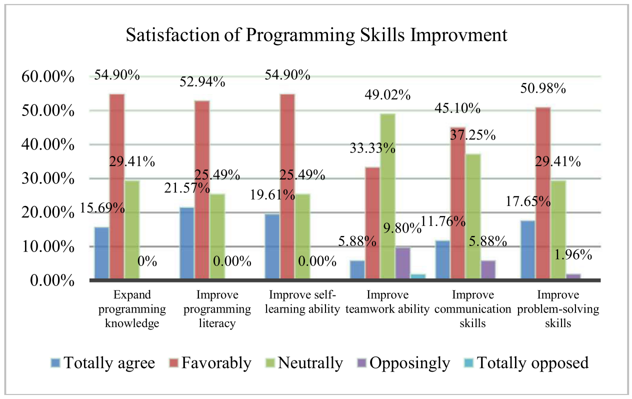


FIGURE 13. **Satisfaction of programming ability improvement.**

students recognized the multidimensional precision teaching evaluation system.

The research carried out investigation on the satisfaction of programming skills improvement in six aspects (as shown in Figure 13): expanding programming knowledge, improving programming literacy, enhancing self-study ability, improving problem-solving skills, and strengthening communication ability. Over 50% of the students thought that their programming knowledge, programming literacy, learning ability, and problem-solving skill were improved, and more than 45% thought that their team cooperation ability and communication ability were enhanced after learning programming course. All these results show that the precision teaching classroom of C Language programming course plays an obvious role in training students' programming literacy and improving their programming skills.

VII. CONCLUSION

In teaching programming courses, teachers always face the challenges of selecting appropriate teaching methods, effective teacher-student interactions in programming class, maintaining learning enthusiasm and confidence, and comprehensive evaluation of learning process and learning effects. This research constructs a precision teaching classroom for university programming courses. Therefore, this study mainly focuses on the following three questions: What are realization conditions of precision teaching classroom for university programming courses? What teaching phases and teaching links are included in this precision programming teaching classroom? What is the teaching effect of constructed precision programming teaching classroom?

This study considers that there are three supporting conditions for precision programming teaching, namely offline teaching classroom based on smart teaching tool (Teachermate), online teaching platform supporting mobile ubiquitous learning (Xuexitong), and programming practice environment with automatic assessment (PTA). In offline teaching classroom, teacher mainly teaches programming knowledge, conducts teaching interaction and classroom tests. Teachermate can automatically record and statistically analyze the data generated in this teaching classroom, and help teacher to evaluate students' learning participation and learning attitude in programming learning. In online autonomous learning, teacher mainly publishes learning resources and online programming learning tasks. Xuexitong automatically records student's online learning behavior trajectory, helping teachers to master students' consciousness and initiative in learning programming. In programming practice teaching, teacher releases programming practice tasks on PTA, which records programming practice process data of students in real time. These data reflect students' learning engagement and programming skills growth in programming practice, and assist teacher to provide differentiated programming learning guidance to students.

This research designs precision programming teaching process of three teaching stages: preparation phase driven by student data, implementation phase driven by learning process data, and evaluation phase driven by multidimensional evaluation data. Then this study carried out teaching practice in C Language Programming course in a Chinese university. The results of teaching practice reveal that the precision classroom of programming courses constructed in this paper improve the insights and decision-making ability of teacher in programming teaching and facilitate the teacher to formulate precision teaching objectives and design adaptive teaching activities.

In addition, Teachermate, Xuexitong and PTA record and analyze students' learning behavior and learning outcomes in real time and accurately depict their current learning states and emotions, so that the teacher could pay full attention to individual differences and changes of students' programming skills in programming learning. Eventually, teaching effects apparently superior to those of ordinary programming classroom were obtained, and students' programming learning efficiency and learning quality were improved. Practice has proved that the precision teaching classroom for university programming courses proposed in this article provides a complete set of effective teaching methods for improving programming teaching.

This study was completed with strong support of the university where the author works in. However, there were two limitations in this research. One was the lack of real-time data collection for programming learning process resulted in not very high data accuracy, which made the characterization of students' learning process and learning behavior not accurate enough. Another limitation was that in practice activities of precision teaching, sometimes students could not fully cooperate with teaching research, resulting in insufficient data sets. In view of these research limitations, the authors will continue to carry out multiple rounds of teaching practice and teaching effect evaluation in programming courses of different majors, so as to continuously enrich teaching data sets, optimize the proposed precision programming teaching model and improve teaching effects of university programming courses.

REFERENCES

[1] J. Keppens and D. Hay, "Concept map assessment for teaching computer programming," *Comput. Sci. Educ.*, vol. 18, no. 1, pp. 31–42, Mar. 2008.

[2] X. M. Wang, G. J. Hwang, Z. Y. Liang, and H. Y. Wang, "Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer assessment attempt," *Educ. Technol. Soc.*, vol. 20, no. 4, pp. 58–68, Apr. 2017.

[3] D. Krpan, S. Mladenović, and M. Rosić, "Undergraduate programming courses, students' perception and success," *Proc. Social Behav. Sci.*, vol. 174, pp. 3868–3872, Feb. 2015.

[4] I.-H. Hsiao, P.-K. Huang, and H. Murphy, "Integrating programming learning analytics across physical and digital space," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 206–217, Jan. 2020.

[5] C. Troussas, A. Krouska, and C. Sgouropoulou, "A novel teaching strategy through adaptive learning activities for computer programming," *IEEE Trans. Educ.*, vol. 64, no. 2, pp. 103–109, May 2021.

[6] Y. Soepriyanto, P. F. A. Fazarini, T. R. Maulida, A. Nurfahrudianto, and Z. Afandi, "The effect of player type in learning programming with online gamification activities on Student learning outcomes," in *Proc. 2nd Int. Conf. Inf. Technol. Educ.*, Zhangjiajie, China, Jan. 2022, pp. 382–386.

[7] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bull.*, vol. 37, no. 3, pp. 14–18, Sep. 2005.

[8] J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, and A. Millner, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009.

[9] S. Mohorovicic and V. Strcic, "An overview of computer programming teaching methods in higher education," in *Proc. Central Eur. Conf. Inf. Intell. Syst.*, 2011, pp. 47–52.

[10] A. Brooks, "Failure rates in introductory programming revisited," *Comput. Rev.*, vol. 56, no. 9, p. 570, Sep. 2015.

[11] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," *ACM SIGCSE Bull.*, vol. 39, no. 2, pp. 32–36, Jun. 2007.

[12] O. R. White, "Precision teaching—Precision learning," *Exceptional Children*, vol. 52, no. 6, pp. 522–534, Apr. 1986.

[13] O. R. Lindsley, "Precision teaching: Discoveries and effects," *J. Appl. Behav. Anal.*, vol. 25, no. 1, pp. 51–57, Mar. 1992.

[14] O. R. Lindsley, "Precision teaching's unique legacy from B. F. Skinner," *J. Behav. Educ.*, vol. 1, no. 2, pp. 253–266, Jun. 1991.

[15] L. Potts, J. W. Eshleman, and J. O. Cooper, "Ogden R. Lindsley and the historical development of precision teaching," *Behav. Analyst*, vol. 16, no. 2, pp. 177–189, Oct. 1993.

[16] R. M. Kubina, R. Morrison, and D. L. Lee, "Benefits of adding precision teaching to behavioral interventions for students with autism," *Behav. Intervent.*, vol. 17, no. 4, pp. 233–246, 2002.

[17] K. Johnson and E. Street, *Precision Teaching*. Hoboken, NJ, USA: Wiley, 2014. Accessed: May 19, 2014.

[18] A. C. Downer, "The national literacy strategy sight recognition programme implemented by teaching assistants: A precision teaching approach," *Educ. Psychol. Pract.*, vol. 23, no. 2, pp. 129–143, Jun. 2007.

[19] C. P. Griffin and L. Murtagh, "Increasing the sight vocabulary and reading fluency of children requiring reading support: The use of a precision teaching approach," *Educ. Psychol. Pract.*, vol. 31, no. 2, pp. 186–209, Apr. 2015.

[20] A. Vostanis, C. Padden, M. Chiesa, K. Rizos, and P. E. Langdon, "A precision teaching framework for improving mathematical skills of students with intellectual and developmental disabilities," *J. Behav. Educ.*, vol. 30, no. 4, pp. 513–533, Dec. 2021.

[21] D. Lambe, C. Murphy, and M. E. Kelly, "The impact of a precision teaching intervention on the reading fluency of typically developing children," *Behav. Intervent.*, vol. 30, no. 4, pp. 364–377, Nov. 2015.

[22] A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," in *Proc. 10th Annu. Conf. Int. Comput. Educ. Res.*, Jul. 2014, pp. 19–26.

[23] J. Holvikivi, "Conditions for successful learning of programming skills," in *Proc. IFIP World Comput. Congr.*, 2010, pp. 155–164.

[24] M. Hertz and M. Jump, "Trace-based teaching in early programming courses," in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2013, pp. 561–566.

[25] S. M. M. Rubiano, O. Lopez-Cruz, and E. G. Soto, "Teaching computer programming: Practices, difficulties and opportunities," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2015, pp. 1–9.

[26] N. Eltegani and L. Butgereit, "Attributes of students engagement in fundamental programming learning," in *Proc. Int. Conf. Comput., Control, Netw., Electron. Embedded Syst. Eng. (ICCNEEE)*, Kauai, HI, USA, Sep. 2015, pp. 101–106.
[27] A. Gomes and A. Mendes, "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Madrid, Spain, Oct. 2014, pp. 1–8.
[28] B. Hartanto, "Enhancing the student engagement in an introductory programming: A holistic approach in improving the student grade in the informatics department of the university of Surabaya," *Commun. Comput. Inf. Sci.*, vol. 516, pp. 493–504, 2015.
[29] D. Ramey, S. Lydon, O. Healy, A. McCoy, J. Holloway, and T. Mulhern, "A systematic review of the effectiveness of precision teaching for individuals with developmental disabilities," *Rev. J. Autism Develop. Disorders*, vol. 3, no. 3, pp. 179–195, Mar. 2016.
[30] C. Binder, "Precision teaching: Measuring and attaining exemplary academic achievement," *Youth Policy*, vol. 10, no. 7, pp. 12–15, Jul. 1988.
[31] R. Sundhu and M. Kittles, "Precision teaching: Does training by educational psychologist have an impact?" *Educ. Psychol. Pract.*, vol. 32, no. 1, pp. 13–23, Jan. 2016.
[32] M. Nancy, "Response to intervention and precision teaching: Creating synergy in the classroom," *Educ. Treat. Children*, vol. 37, no. 1, pp. 181–185, Jan. 2014.
[33] S. M. Datchuk, "A direct instruction and precision teaching intervention to improve the sentence construction of middle school students with writing difficulties," *J. Special Educ.*, vol. 51, no. 2, pp. 62–71, Aug. 2017.
[34] L. Mannion and C. Griffin, "Precision teaching through Irish: Effects on isolated sight word reading fluency and contextualised reading fluency," *Irish Educ. Stud.*, vol. 37, no. 3, pp. 391–410, Jul. 2018.
[35] S. S. Chapman, C. B. Ewing, and M. P. Mozzoni, "Precision teaching and fluency training across cognitive, physical, and academic tasks in children with traumatic brain injury: A multiple baseline study," *Behav. Intervent.*, vol. 20, no. 1, pp. 37–49, Feb. 2005.
[36] C. Gist and A. J. Bulla, "A systematic review of frequency building and precision teaching with school-aged children," *J. Behav. Educ.*, vol. 31, no. 1, pp. 43–68, Mar. 2022.
[37] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal, "A survey on online judge systems and their applications," *ACM Comput. Surv.*, vol. 1, no. 1, pp. 1–6, 2018.
[38] R. M. Gagne, L. J. Briggs, and W. Walter, "Principles of instructional design," *Perform. Improvement*, vol. 44, no. 2, pp. 44–46, 2005.
[39] P. L. R. Smith, *Instructional Design*. Hoboken, NJ, USA: Wiley,2004.
[40] Ü. Çakıroğlu, "Analyzing the effect of learning styles and study habits of distance learners on learning performances: A case of an introductory programming course," *Int. Rev. Res. Open Distrib. Learn.*, vol. 15, no. 4, pp. 161–184, Aug. 2014.
[41] Y. Z. Mao et al., "The investigation of analyzing learning situations in high school mathematics instructional designs," *J. Math. Educ.*, vol. 15, 2018.
[42] D. M. Clarke and A. Roche, "Students' fraction comparison strategies as a window into robust understanding and possible pointers for instruction," *Educ. Stud. Math.*, vol. 72, no. 1, pp. 127–138, Sep. 2009.
[43] M. A. Heng and A. Sudarshan, "'Bigger number means you plus!'—Teachers learning to use clinical interviews to understand students' mathematical thinking," *Educ. Stud. Math.*, vol. 83, no. 3, pp. 471–485, Jul. 2013.
[44] K. Schildkamp, M. K. Lai, and L. Earl, *Data-Based Decision Making in Education: Challenges and Opportunities*. Dordrecht, The Netherlands: Springer, 2013, pp. 55–59.
[45] W. B. Kippers, C. L. Poortman, K. Schildkamp, and A. J. Visscher, "Data literacy: What do educators learn and struggle with during a data use intervention?" *Stud. Educ. Eval.*, vol. 56, pp. 21–31, Mar. 2018.
[46] A. McDowall, C. Mills, K. Cawte, and J. Miller, "Data use as the heart of data literacy: An exploration of pre-service teachers' data literacy practices in a teaching performance assessment," *Asia–Pacific J. Teacher Educ.*, vol. 49, no. 5, pp. 487–502, Oct. 2021.
[47] Y. X. Yuantian, "Research on the application of mixed classroom teaching mode in universities," *Univ. Teach. China*, vol. 10, no. 8, pp. 78–86, Aug. 2020.
[48] F. F. Gong, "Time and space flip to 'deep learning': Analysis on the deepening path of information literacy flipping classroom teaching," *Library J.*, vol. 39, no. 11, pp. 74–78, Nov. 2020.
[49] *Programming Teaching Assistant*. [Online]. Available: https://pintia.cn/
[50] D. A. Kolb, *Experimental Learning: Experience as the Source of Learning and Development*. New York, NY, USA: Prentice-Hall, 1984, pp. 61–98.
[51] D. H. Granello, "Promoting cognitive complexity in graduate written work: Using Bloom's taxonomy as a pedagogical tool to improve literature reviews," *Counselor Educ. Supervision*, vol. 40, no. 4, pp. 292–307, Jun. 2001.
[52] J. Steurer, "The Delphi method: An efficient procedure to generate knowledge," *Skeletal Radiol.*, vol. 40, no. 8, pp. 959–961, Aug. 2011.
[53] H. A. Linstone and M. Turoff, "Delphi: A brief look backward and forward," *Technol. Forecasting Social Change*, vol. 78, no. 9, pp. 1712–1729, 2019.

FANG YU received the B.S. degree in computer science education from Inner Mongolia Normal University, in 2002, the M.S. degree in education from Liaoning Normal University, in 2009, and the Ph.D. degree in educational information technology from Central China Normal University, in 2022. She is the author of two books, more than 20 articles, and three inventions. Her research interests include education data mining and learning analysis.

YAN LIU received the B.S. and M.S. degrees in computer science and technology from the Inner Mongolia University of Technology, in 2013. He has published nearly ten journals in the related computer science education fields. His research interests include computer graphics and cloud computing.

FENGYAN XIAO received the B.S. degree in educational technology and the M.S. degree in education from Inner Mongolia Normal University, in 1999 and 2006, respectively. Since 2010, she has been an Associate Professor. She is the author of one book and more than ten articles. Her research interests include application of information technology in teaching and teacher professional development.