Elinvo (Electronics, Informatics, and Vocational Education) ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)
Vol 11, No. 1, 2026, pp. 18-33

AI Chatbot-Enhanced Flipped Classroom and Its Impact on Personalized Learning and Procedural Knowledge Acquisition in Vocational Computer Programming Education

Admaja Dwi Herlambang[1*], Aditya Rachmadi[2]

[1,2] Universitas Brawijaya, Malang, Indonesia
[1] herlambang@ub.ac.id*; [2] rachmadi.aditya@ub.ac.id
* corresponding author

Article Info

Article history:
Received March 06, 2026
Revised May 08, 2026
Accepted June 19, 2026
Available Online June 19, 2026

Keywords:
Flipped classroom; artificial intelligence; chatbots; personalized learning; procedural knowledge; computer programming education; vocational education

Abstract

This research investigates whether an AI chatbot can further enhance the effectiveness of the flipped classroom model by facilitating personalized learning in computer programming. However, the acquisition of procedural programming knowledge through AI-driven personalisation is seldom addressed within vocational frameworks. A quasi-experimental pretest-posttest with a matched samples control group design was used with a population of 60 in a vocational high school in Indonesia. The experimental treatment group of 30 students received flipped instruction supported by AI chatbot; the 30-student strong control group had traditional flipped instruction. Cognitive test data, psychomotor tests, and self-reported learning questionnaires measured data. Results show that the AI-enhanced flipped classroom outperformed the traditional flipped classroom to a great degree in both personalized learning (adjusted mean difference = 12.30, $p < 0.001$, partial $\eta^2 = .335$) and procedural knowledge acquisition (mean difference = 9.70, $p < 0.001$). The effect sizes were large for individualized instruction (Cohen's d = 0.89, 95% CI: 0.75 to 1.03) and medium for procedural knowledge (Cohen's d = 0.62, 95% CI: 0.49 to 0.75). Of particular note was that the experimental group not only did better on coding tasks (d = 1.18), but even more so on debugging efficiency (d = 1.48).

***Corresponding Author:**
Email: herlambang@ub.ac.id

INTRODUCTION

Literacy in programming as an important capability in the computing-dominated environment of the contemporary world. Learning programming skills has been considered necessary for people aspiring to join the job market [1]. There is ample evidence of challenges in terms of instruction under conditions of limited resources, while the supply of competent programmers who can teach others how to program is often deficient in many vocational schools. Traditional instructional methodologies have been shown to fall short when it comes to tackling the conceptual character of programming [2]. It is flipped teaching: take the theory from the classroom and use face-to-face moments of contact to practice. It is highly effective in programming education, active participation, and feedback [3], [4]. AI chatbots have changed education by offering personalised learning and flexible teaching [5]. They provide ongoing access, customised explanations, adjustable help, fast feedback, and teamwork simulations. The technology excels at customising content and complexity based on learner performance. Despite interest in flipped methodologies and AI learning, research gaps exist in their integration within programming education. Though flipped learning studies exist in the realms of STEM [6], [7], [8], [9], analyses of AI-integrated flipped models are still scant. These deficits need an approach based on evolution in programming languages, increased enrollment in computer science, and continued pedagogical challenges.

Equally, the interplay between AI and flipped learning strategy has been extensively researched through a host of theoretical frameworks that underpin their role within the educational discourse-the TPACK framework was used for investigating an intersection of technological and pedagogical competencies in programming education [10]; from another perspective, the social constructivist learning theory was used for explaining how AI-enhanced learning environments foster the collaborative generation of knowledge and guide learning processes [11]. Self-regulated learning theory has illuminated how AI chatbots support metacognitive strategies and autonomous learning behaviour within the context of programming education [12], [13], [14]. According to available literature, individualized learning enabled by artificial intelligence affects the formation of procedural knowledge. Current research shows that there are gains in problem-solving skills in the context of instruction made possible through artificial intelligence [15]. This research is far from being unique. There exists an alignment of different approaches and findings in current scientific publications, which point to a lack of instructional approaches that combine artificial intelligence and inverted learning [16]. Therefore, there are sufficient grounds for conducting a research project dedicated to AI-powered flipped classrooms as a solution to persistent problems in programming instruction.

In spite of numerous cases proving the efficiency of the flipped classroom strategy, there exist only a few examples of applying it to the development of procedural knowledge in the area of vocational programming. AI-powered personalisation has not been systematically studied within this specific instructional setting. This gap represents a substantive limitation in the existing body of empirical literature. Prior studies have examined the flipped classroom model and AI integration as separate constructs. No empirical investigation has been conducted using a propensity-score-matched quasi-experimental design to assess both constructs simultaneously. The present study addresses this absence directly. A rigorously controlled empirical dataset was generated to examine the effects of AI chatbot integration on personalised learning experiences and procedural knowledge acquisition within vocational computer programming education in Indonesia.

Contemporary research in educational technology has highlighted several critical aspects of AI integration in programming instruction that warrant further investigation. Cognitive load theory has also been used to explore how AI-aided teaching may optimise the cognitive load of the activities involved in programming [17], [18]. Other academic research conducted within an adaptive learning system showed how AI algorithms can provide personalised teaching consistent with differentiated learning behaviour and performance measures [19], [20]. The notion of proximal development has been extended to investigate how AI chatbots provide scaffolded support that adjusts to the learners' developing capabilities [21]. In fact, significant gaps persist in understanding how best to integrate AI capabilities into flipped learning environments to inspire better development of procedural knowledge and problem-solving competencies [22], [23], [24], [25]. These research strands combined underline one important implication of the empirical investigation that needs to be pursued regarding AI-supported flipped classrooms' effectiveness within the context of programming education. The identified research gaps and theoretical positions have informed the current study's purposes and methodological approach.

The work presented herein discusses how AI-enhanced flipped classroom methodologies affect individual learning experiences and procedural knowledge in programming education. Guided by four significant objectives, like the implementation effectiveness of AI chatbots, influences of the model on various dimensions of individual learning, retention of procedural knowledge, strategic enabling implementation frameworks, and identification of potential obstacles, the present research forms a basis for the above discussion. Two hypotheses are fundamental to this research: first, students would have better-individualised learning outcomes when taught in an AI-enhanced flipped classroom compared to those taught within a standard flipped classroom, and second, students will reveal significantly better procedural knowledge in cognitive evaluations concerning programming performance tests. Temporally, this research aligns with the increased need for computational literacy at all levels. With educational institutions facing the need to prepare learners for digital workforce integration, there is an increasing demand for novel methodologies using contemporary technologies. AI chatbot integration within flipped learning frameworks shows promising potential for optimising scalability, learning personalisation, and instructional efficiency in programming education.

Literature Review

The flipped learning paradigm revolutionises traditional instruction by relocating knowledge transmission outside classroom boundaries and freeing up in-class time for active engagement to a maximum [26], [27]. Such is in congruence with the constructivist approach, whereby knowledge is created through experience in interaction. This model is based on the self-regulated learning theory since Zimmerman stresses that learners should structurally guide their cognitive processes to reach educational goals [28]. Empirical evidence in the form of implementation outcomes in STEM education demonstrates improved engagement patterns as well as increased opportunities for practical applications [29]. However, success relies on students' self-regulatory abilities and time management [30], [31], [32]. Lo & Hew present the results of a meta-analysis that shows the gains achieved in computer science learning are moderate when compared to traditional approaches [33]. These best practices identified involve embedding interactive coding activities across the learning process, collaboration on programming projects, and immediate feedback. Again, these practices introduce methods of extending the direct interaction afforded by the flipped model itself to meet the complex and highly abstract nature of programming concepts.

Within educational psychology, several theoretically sound and well-supported frameworks support the cognitive aspects of flipped learning settings [34]. Programming concepts in flipped learning conditions may be explained within the working memory using Cognitive Load Theory [35]. Information Processing Theory may be utilised to explore the knowledge structure development from the structured presentation of programming concepts within flipped environments [36], [37]. Although incremental development of the frameworks for programming knowledge is derived through Schema Theory tenets, consolidation of these experiences heralds practice instead. The social cognitive theory of self-efficacy underpins the metacognitive dimensions that flipped learning offers [38], which develops from successive mastery of programming-related experiences. More empirical studies testifying to increased cognitive engagement within technology-enhanced learning further complement these theoretical underpinnings. Combining such cognitive theories provides excellent support for understanding how complex programming skills are supported within a flipped learning environment.

In modern learning environments, for example, AI would help support flipped learning conditions utilising various theories and teaching frameworks [39], such as the Adaptive Learning Theory, wherein AI systems may even alter instructional routes according to individual learning pathways [40]. This continuous assessment and modification of instructional sequences verify the methodology of Dynamic Assessment in programming education. Artificially intelligent network learning experiences promote the core ideas of Connectivist Learning Theory toward constructing improved knowledge . This deeper engagement furthers the Technology Acceptance Model by the learners concerning AI-enhanced programming instruction [41]. The capability of AI systems to analyse and react to in-air learning data extends the Learning Analytics Framework. These combined frameworks enable the understanding of how AI-enhanced flipped learning environments support improved personalised education. Coupling such theoretical insights with empirical results is a promising avenue to explore individualised pedagogical strategies in programming education.

Flipped classroom activities outside the classroom allow students to learn independently [9]. Individualised education frameworks include learner profiles, personal trajectories, and adjustable settings [42]. Modern adaptive technologies can help supervise flipped classroom learning outside the classroom so that personalised learning can still be appropriately realised. Modern adaptive technologies use various algorithms for real-time assessment of performance and the presentation of students with individual content, while learning analytics allow for evidence-based practice in pedagogies [43]. Empirical investigations demonstrate that personalised teaching is positively linked with students' academic achievement [44]; the motivational levels were also higher because learners have more freedom [45]. In personalised learning, learners will assume greater responsibility for their learning process and consequences relevant to the core concept of self-directed learning [42], [46]. Thus, it can be concluded that flipped classrooms can still be guaranteed to have the potential to accommodate personalised and individual learning as long as they are supported by relevant technology.

Cognitive architecture and adaptive learning theories underpin the core theories of personalised learning environments. Based on cognitive load theory, those principles have been identified to ensure an optimum amount of mental effort on behalf of the learners owing to AI-assisted instructional design for programming education [47]. Again, the Zone of Proximal Development concept finds its manifestation in the systematised scaffolding system through adaptive technologies in the personalised learning environment. The increase in metacognitive strategies and autonomous learning behaviour in technology-enhanced environments can underline the notions of the Self-Regulated Learning Theory. Facilitating collaborative knowledge creation in artificially intelligent learning environments enriches Social Constructivist Theory, while Technology Acceptance Model principles are strengthened as learners increasingly interact with educational systems that use AI technology. These theories combine to form a framework that provides a structured way of understanding how technology helps create personalized learning environments. From recent research, there appears to be a point of empirical convergence between these theories that could be considered a fruitful avenue for exploring how theory can be turned into practice in the educational field.

A number of these theories are directly applicable to learning through artificial intelligence tools. The Learning Analytics Framework involves the careful data collection and analysis of learners to guide pedagogic decisions [43]. The Adaptive Learning Systems Theory incorporates the ability to modify instructional content in real time according to learner's performance. Information Processing Theory is used to provide guidance in delivering structured programming information. Finally, Connectivist Learning Theory explains the process of networking of knowledge through digital instruction. The approach of Dynamic Assessment is valid regarding the possible ongoing adjustment and modification of learning pathways. These theoretical lenses highlight important evidence regarding designing and implementing personalised learning environments within programming education. These theoretical frameworks embedded in the integration provide a concrete framework to study how personalised learning environments will be effectively executed in different educational settings, particularly in programming education.

The advantages of the flipped classroom are expected to be applied in various fields of study. One field of study currently hot and much needed in the 21st century is computer programming education [48]. Procedural knowledge is foundational in the education of programming skills and includes such operational skills as compiling, debugging, and optimising the code [49], [50]. Declarative knowledge is important for conceptual understanding, but procedural mastery determines practical implementation capabilities [51]. To acquire this knowledge, structured practice principles are paramount, entailing systematic repetition for performance enhancement and step-by-step solution analysis. The assessment of procedural knowledge requires models that would enable to evaluate practical programming competences effectively [52]. The traditional methods of assessment were found to be unable to cover all aspects of the process of procedural skill development. That is why professionals have been actively seeking new assessment techniques that would make it possible to approach the needs of modern learners adequately.

Modern assessment techniques are centered around demonstrating one's programming skills practically. Problem-solving operations and error detection are used as methods to assess actual abilities of individuals engaged in this activity[53], [54]. The methods are considered to be more reflective of the procedural aspects involved in programming. Educational technologies including AI chatbots represent yet another group of tools used in programming education. There are two major types described in the existing literature: rule-based chatbots and machine-learning chatbots [55]. The former type of AI system employs a conditional logic, while the latter uses natural language processing. They serve as tools for making static programming teaching and learning structures adaptable [54]. One of the major challenges related to the problem of programming education is the need to provide scalable personalized instruction [19], [55]. Personalized learning and tutoring is possible through automated response mechanisms that use machine learning algorithms to customize their content according to the learners' level.

METHODS

The current study is a quasi-experiment that was done to explore how an AI-enhanced flipped classroom model influences learning outcomes about computer programming procedural knowledge at an individual level. Indeed, the quasi-experimental design is particularly suitable in educational research when randomisation is hard to implement, if not impossible [56]. This method boosts external validity and real-world applicability despite limited experimental control. It employed a matching-only pretest-posttest control group design to compare treatment effects while addressing initial group differences [57]. In the matched pretest-posttest control group design, participants with relevant characteristics are assigned to experimental and control groups. A pretest is administered, an intervention is provided to the experimental group, and then a posttest to both groups. This design diminishes selection biases and strengthens internal validity from a non-matched design. It is considered an educational setting without random assignment. A proper and strict matching procedure was done to equate the control and experimental groups. In order to achieve balance between covariates of the two groups, propensity score matching was used to reduce any potential selection biases in this study [58].

Before the treatment was undertaken, an implementation protocol that spanned six weeks was designed. Treatment fidelity was maintained through a strict adherence to the protocol for both the control and experimental groups. For the control group, the traditional flipped classroom format was applied where video lessons as well as reading materials were given before class at the beginning of each week. Coding exercises and debriefing activities occurred during class time. The experimental group received an identical content structure with one additional component: an AI chatbot was deployed via a dedicated learning management system module and made accessible during both pre-class and post-class phases. Full weekly treatment schedule for both groups is presented in Table 2. Complete lesson plans, chatbot configuration specifications, and treatment protocols are available upon request from the corresponding author to support independent replication of the intervention.

Several steps were involved in the PSM procedure. First, using several pre-treatment covariates, including, but not limited to, age, gender, prior programming experience assessed on a 5-point Likert scale, and baseline cognitive ability test scores, a logistic regression model estimated the Propensity score for each participant. These variables were incorporated upon previous mentions in the literature for their potential effect on the learning results of computer programming (Robins et al., 2019). Estimations were presented using the propensity score, including a nearest neighbour matching algorithm with a 0.2 standard deviation of a logit as a calliper. This method matched each experimental participant with a similar control participant within a calliper distance [59] which is a 1:1 matching ratio created 30 pairs. Standardised mean differences (SMDs) were calculated for all covariates pre- and post-matching to assess quality. An SMD less than 0.1 implies that the differences between groups are negligible [60]. Variance ratios were checked to confirm similar distributions of the continuous covariates. The balance diagnostic after matching indicated that all the SMDs were less than 0.1, with the maximum being 0.08 for cognitive ability test scores. The variance ratios for continuous variables varied between 0.92 and 1.07, indicating good balance. Therefore, this will indicate that the matching procedure created comparable groups and enhanced internal validity in this study.

Participants were drawn from Indonesian vocational high schools and targeted at computer programming students. The sample size was determined using G*Power 3.1 software [61] with a medium effect size, given Cohen's d = 0.62 for procedural knowledge according to the results of previous flipped classroom studies [62]. The power analysis indicated a needed sample size of 60 participants (30 per group) for 80% power at a 0.05 significance level. Participants had to have enrolled in one vocational high school in Indonesia with a computer programming course. Exclusion Criteria: Students who have already experienced an AI chatbot will be excluded to avoid the confounding effect. Recruitment: Collaboration with teachers and school staff will be necessary, targeting only the eligible students and permissions. Ethical issues played a significant role in this research. Free and informed consent, with all the details pertinent to the study, including all risks and benefits, was provided to the participants and guardians. Data collection was anonymised and securely stored on encrypted servers for use by authorised teams only.

The intervention in this study was delivered in the traditional flipped classroom-testing condition, and the AI-enhanced flipped classroom-treatment condition. While both were flipped models, the experimental group received support in flipped class conditions with support given through an AI-driven chatbot. Traditionally, pre-class material, such as video lectures, was given, followed by in-class problem-solving activities. In-class sessions involved teamwork, peer talks, and tutor guidance. It is a modification of the Flipped Classroom Model, yet with an integrated AI-driven GPT chatbot. The 24/7 presence of the chatbot nurtured preparation together before a class through collaborative work and supported reflections afterwards for more profound learning.

To determine the effectiveness of the interventions, three methods were adopted. One method was procedural knowledge. This entailed a 30-item multiple choice cognitive test aimed at assessing the level of knowledge about programming skills and procedure used in problem-solving. The internal consistency of this measure was determined using Cronbach's Alpha, which had a score greater than

0.80. The other measure of assessment was psychomotor performance. This consisted of 5 coding tests, each accompanied by a rubric for consistent evaluation. A sub-sample of coding tests was rated independently by two raters, whose inter-rater reliability was above Cohen's Kappa value of 0.75 [63]. Personalised learning was assessed using a questionnaire drawn from a validated instrument, which is the Personalised Learning Environment Questionnaire [64]. In the pilot test, 30 participants were tested. Item-total correlations were greater than 0.30, suggesting construct validity and internal consistency.

The duration of data collection took eight weeks in total, including one week before and after intervention. Two days of training were provided to research assistants before the start of data collection. Standardisation of measurement was ensured to eliminate evaluator biases and to maintain confidentiality of participants. Descriptive measures were obtained to describe characteristics of the sample and the distribution of scores. Box plots and histograms were constructed to aid in score distribution examination and detect outliers. Comparison between groups was done using ANCOVA where pre-test scores were taken as covariates to reduce error variance and improve the precision of the estimate of effect size [65]. The effect sizes were determined using Cohen's d. Traditionally, 0.2, 0.5, and 0.8 were taken as indicators of small, medium, and large effects, respectively [66]. Precision was reported with 95% confidence intervals for the effect sizes. Assumptions of the statistical tests were checked and, where violated, subjected to procedures including a robust method and data transformation to ensure that results were valid. This research work aims to contribute to the solid evidence on the effectiveness of AI-enhanced Flipped Classrooms for personalised learning and programming education. The instruments were well validated in a quasi-experiment, permitting thorough statistical analysis. Hence, the results are reliable and instructive for educators and researchers in technology in education and computer science.

Table 1. Experimental Design and Data Collection Overview

Aspect	Description
Research Design	Quasi-experimental, Matching-only pretest-posttest control group design
Sample Size	Total N = 60 (30 per group)
Groups	Control Group (CG): Traditional flipped classroom
	Experimental Group (EG): AI-enhanced flipped classroom
Intervention Duration	6 weeks (2 cycles of 3 weeks each)
Data Collection Points	Pretest (1 week before intervention)
	Posttest 1 (after 3 weeks)
	Posttest 2 (after 6 weeks)
Instruments	1. Cognitive test for procedural knowledge (30 multiple-choice items)
	2. Psychomotor performance assessment (5 coding tasks)
	3. Self-reported personalised learning questionnaire
Reliability Measures	Cognitive test: Cronbach's alpha > 0.80
	Psychomotor assessment: Inter-rater reliability (Cohen's kappa > 0.75)
	Personalised learning questionnaire: Item-total correlations > 0.30
Primary Outcome Measures	1. Procedural knowledge scores (0-100)
	2. Personalised learning scores (0-100)
Statistical Analyses	1. Descriptive statistics
	2. ANCOVA for between-group comparisons
	3. Effect size calculations (Cohen's d)

Table 2. Weekly Treatment Schedule for Both Groups

Week	Control Group (CG)	Experimental Group (EG)	Learning Objective
1	Pre-class: Video lecture on programming syntax. In-class: Guided coding exercises	Pre-class: Video lecture & AI chatbot Q&A on syntax. In-class: Guided coding exercises & chatbot scaffolding	Understand basic programming syntax and data types
2	Pre-class: Reading on control structures. In-class:	Pre-class: Reading & AI chatbot adaptive drill on conditionals. In-class: Pair programming & chatbot hints	Apply control structures in coding tasks

	Pair programming on conditionals		
3	Pre-class: Video on functions and procedures. In-class: Function-writing exercises	Pre-class: Video & AI chatbot personalised function examples. In-class: Function-writing & chatbot feedback	Construct and call user-defined functions
4	Pre-class: Reading on arrays and loops. In-class: Loop-based coding problems	Pre-class: Reading & AI chatbot adaptive loop practice. In-class: Loop coding & real-time chatbot correction	Implement iterative structures using arrays and loops
5	Pre-class: Video on debugging strategies. In-class: Debugging exercises (timed)	Pre-class: Video & AI chatbot step-by-step debugging walkthroughs. In-class: Debugging exercises & chatbot error analysis	Diagnose and correct syntactic and logical errors in code
6	In-class: Summative coding task (no assistance). Post-class: Reflection worksheet	In-class: Summative coding task (no chatbot). Post-class: AI chatbot-guided reflection and review	Demonstrate integrated procedural knowledge under assessment conditions

RESULT AND DISCUSSION

Result

This quasi-experiment explores the effects of the improved Flipped Classroom with AI on the personalisation effect and procedural knowledge acquisition in terms of computer programming. The data included demographic descriptions of the participants, personalisation's effect, procedural knowledge acquisition, correlation tests, and feedback about AI Chatbot assistants. Sixty vocational high school students majoring in information technology participated in the study. The sample consisted of 32 males (53.3%) and 28 females (46.7%), with ages ranging from 16 to 18 years ($M = 17.2$, $SD = 0.73$). The participants were randomly distributed in an experimental group of 30 and a control group of 30. Previous programming experience was measured with a 5-point Likert scale. No significant differences were found between the two conditions (experimental condition: $M = 2.87$, $SD = 0.94$; Control condition: $M = 2.73$, $SD = 0.87$), $t(58) = 0.59$, $p = .556$. Independent samples t-tests of pretest scores on the measures of personalised learning and procedural knowledge were conducted where necessary to ensure that the groups were equivalent at baseline. Comparatively, neither personalised learning $t(58)=0.42$, $p=.676$ or procedural knowledge $t(58)=0.38$, $p=.705$ yielded a significant difference between the experimental and control groups. Thus, the groups were equal going into the study.

Table 3. Participant Demographics and Baseline Measures

Characteristic	EG (n = 30)	CG (n = 30)	p-value
Age (years)	17.3 (0.75)	17.1 (0.71)	.279
Gender (% male)	56.70%	50.00%	.612
Prior experience	2.87 (0.94)	2.73 (0.87)	.556
Pretest PL	62.43 (8.91)	61.57 (7.84)	.676
Pretest PK	58.27 (9.63)	57.40 (8.79)	.705

Note: Values are presented as mean (standard deviation) unless otherwise noted; EG: experimental group; CG: control group; PL: personalised learning; PK: procedural knowledge.

We tested the effect of the AI-enhanced flipped classroom on personalised learning using ANCOVA, having posttest scores as the dependent variable and group assignment as the independent variable. The result was a significant difference in outcome due to the assignment into groups: $F(1, 57) = 28.76$, $p < .001$, partial $\eta^2 = .335$. The adjusted mean difference between the experimental group ($M = 78.63$, $SE = 1.14$) and the control group ($M = 66.33$, $SE = 1.14$) was 12.30 points (95% CI: 9.47 to 15.13), indicating a substantial improvement in personalised learning outcomes for the AI-enhanced flipped classroom group. The calculations for Cohen's d yielded an immense effect size of $d = 0.89$ (95% CI: 0.75 to 1.03), showing the robustness of the intervention effect concerning personalised learning.

Table 4. ANCOVA Results for Personalised Learning Outcomes

Source	df	F	p	Partial η^2
Pretest PL	1	142.35	< .001	.714
Group	1	28.76	< .001	.335
Error	57			

Procedural knowledge was assessed using cognitive tests and psychomotor tasks. The repeated-measures ANOVA showed an interaction between time-pretest versus posttest-and group that reached significance, $F(1, 58) = 19.84$, $p < .001$, partial $\eta^2 = .255$. The experimental group improved more from pretest ($M = 58.27$, $SD = 9.63$) to posttest ($M = 79.13$, $SD = 8.72$) than the control group (pretest: $M = 57.40$, $SD = 8.79$; posttest: $M = 69.43$, $SD = 9.15$). The mean difference in procedural knowledge scores was 9.7 points (95% CI: 7.2 to 12.2, $p < 0.05$). These analyses yielded significant differences between the groups in posttest scores adjusted for the pretest score ($F(1, 57) = 22.91$, $p < .001$, partial $\eta^2 = .287$). Psychomotor performance and debugging efficiency were assessed via coding task scores. At posttest the experimental group outperformed the control group, $M = 82.17$, $SD = 7.94$, $M = 72.50$, $SD = 8.63$, respectively, $t(58) = 4.56$, $p < .001$, $d = 1.18$. The experimental group debugged more efficiently than the control group, $M = 8.23$ min, $SD = 2.14$, compared to $M = 11.87$ min, $SD = 2.76$, $t(58) = -5.72$, $p < .001$, $d = 1.48$. The corresponding effect size calculations, $d = 0.62$, 95% CI: 0.49-0.75, confirm the medium to large effect size as expected for cognitive tests. When considering the scores of the coding tasks, the effect size was $d = 1.18$, and for debugging, the effect size was $d = 1.48$, which also showed larger effect sizes than expected for actual programming skills improvement.

Table 5. Procedural Knowledge Acquisition Results

Measure	EG	CG	MD	t	p	Cohen’s d
CT (Pretest)	58.27 (9.63)	57.40 (8.79)	0.87	0.38	.705	0.09
CT (Posttest)	79.13 (8.72)	69.43 (9.15)	9.70	4.79	< .001	0.62
CTS	82.17 (7.94)	72.50 (8.63)	9.67	4.56	< .001	1.18
DT (minutes)	8.23 (2.14)	11.87 (2.76)	-3.64	-5.72	< .001	1.48

Note: Values are presented as mean (standard deviation); CT: cognitive test; CTS: coding task score; DT: debugging time; EG: experimental group; CG: control group; MD: mean difference.

We calculated Pearson’s product-moment correlation coefficient between the posttest results of personalised learning and procedural knowledge. Quite strong positive correlations were obtained: $r = .724$, $p < .001$, $n = 60$, suggesting that the higher the personalised learning, the higher the increase in procedural knowledge. First, a PSM procedure was performed to equate experimental and control groups. It allowed the balance in the distribution of covariates to minimise selection biases and enhance the validity of the causal inferences. Table 4 describes the SMD and variance ratio for the covariates before and after matching.

Table 6. Standardised Mean Differences and Variance Ratios Before and After Matching

Covariate	Before Matching		After Matching	
	SMD	Variance Ratio	SMD	Variance Ratio
Age	0.267	1.115	0.052	1.021
Gender	0.135	-	0.067	-
Prior Experience	0.156	1.167	0.048	0.973
Cognitive Ability	0.298	1.203	0.080	1.068

Note: SMD = Standardized Mean Difference; Variance ratios are not applicable for binary variables (e.g., gender).

After matching, the test results also show a significant equalisation of balance between the experimental and control groups. The greatest SMD before matching, 0.298, was for cognitive ability test scores, reflecting a sharp difference between the two groups. After matching, all SMDs were less than 0.1, with the highest value equalling 0.080 for cognitive ability scores. These were the balanced variance ratios or the comparison of spreads of continuous variables across groups. The variance ratios before matching were between 1.115 and 1.203, showing a difference in distribution. After matching, variance ratios were closer to 1, between 0.973 and 1.068, thus a similar distribution. After matching, the resultant decrease in SMDs and variance ratios approximating 1 indicate a better balance between experimental and control groups. This balance enhances internal validity by reducing the confounding effect attributed to its baseline characteristics. Because of this, the differences that might be found concerning outcomes in experimental versus control would have to relate to the treatment concerning AI-enhanced flipped classrooms. Besides, propensity score matching enhances balance for possible improvement in the robustness of analyses and study conclusions.

Thematic analysis of comments provided by the experimental group allowed some main themes identified in AI chatbot support: 1) He explained what students wanted to know, showed, and explained it. 2) This chatbot is always available, which, for them, is a leading factor in getting help out of school hours. 3) The speed with which the chatbot responded kept the learning rhythm. 4) To them, simplifying complex concepts on how to program was just incredible. 5) The chatbot allowed for much deeper exploration and more ownership of the learning. Sample quotes from students on these issues are: 1) "The AI chatbot served as a personal tutor and thus helped me understand difficult theories without having to wait for the next session" [Student A]; 2) "My mistakes in coding were instantly detected, thus helping me realize what I was doing wrong" [Student B]; and 3) "The way the chatbot explained things step by step made complex programming ideas much easier to grasp. It was like it knew how to break things down for me." [Student C].

Two persistent challenges in programming education were addressed by the findings. The first concerns the delivery of individualised instruction at scale. A personalised learning gain of 12.30 points was recorded in favour of the experimental group. The second challenge relates to procedural competence development beyond declarative understanding. A mean difference of 9.70 was observed in cognitive test scores between the two groups. Debugging efficiency yielded an effect size of $d = 1.48$. These outcomes indicate that AI chatbot integration produced measurable improvements across both dimensions.

Discussion

Results showed that such an AI-enhanced flipped classroom can support the development of personalised learning, developing students' improvement in procedural knowledge of programming. Our results confirm both hypotheses and contribute to studies of new approaches for teaching STEM subjects. The experimental group's better performance in personalised learning metrics confirms our hypothesis. The enormous effect size (Cohen's $d = 0.89$) emphasises AI chatbot integration's substantial impact on flipped learning, supporting past research on AI's educational potential [67]. The AI system's ability to provide individualised instruction and instant feedback dramatically improves the personalised learning experience over traditional flipped methods.

The educational implications of AI-enhanced flipped classrooms are further supported by established learning theories and frameworks. Markedly enhanced levels of collaborative knowledge construction in the AI-supported educational environment verify the Social Constructivist Theory [68]. The framework of the Zone of Proximal Development is realised by the potential of the AI chatbot to provide personalised scaffold support concerning individual learning paths [69]. Hence, the role of an AI system in enhancing cognitive resources through focused and timely support concerning the comprehension of complex programming concepts shows various key aspects of Cognitive Load Theory. Further development of the TPACK framework by subtle inclusions of AI functions is done within an educational context related to programming [10]. The theory of Self-Regulated Learning is thus supported because the AI chatbot encourages the students to adopt metacognitive strategies and independent learning behaviours. The empirical results of this study further support the basic idea of the Adaptive Learning Systems framework: live adaptation of personalised instruction, depending on performance metrics. These theoretical underpinnings are further complemented by the palpable improvement in procedural knowledge learning, as collected through quantitative tests on learning outcome measures. Theoretical factors discussed earlier provide a clear grounding for the following empirical discussion. The current research investigates concrete achievements in the area of procedural knowledge acquisition as well as their educational implications.

A significant increase in procedural knowledge was reported based on both cognitive and behavioral criteria. Effect sizes varied between 0.62 and 1.48, meaning that a moderate to large impact of the treatment can be registered. These results extend beyond previous findings on flipped classroom efficacy (Lo & Hew, 2017) by demonstrating the additional benefits of AI integration. Compared with the earlier studies of Maher et al. (2015), our research shows that AI augmentation significantly enhances the outcomes in programming education, especially in improving practical skills, with $d = 1.18$

for coding and d = 1.48 for debugging. The strong correlation of .724 between the personalised learning metrics and skill acquisition indicates a key link between individualised instruction and practical skills. In support of the theory underlying personalised learning, this suggests that tailored instruction improves learning outcomes [44], especially in programming education.

The AI-enhanced flipped model significantly enhances programming education in general and the unique needs of every learner for support in particular. By nature, the immediate, personalised support provided by the AI chatbot leads to overcoming the barriers associated with traditional classrooms and thus increases student engagement while decreasing frustration [70]. The significantly enhanced debugging efficiency among the experimental cohorts speaks volumes about the model's effectiveness in developing critical problem-solving capabilities through systematic scaffolding [71]. It is, therefore, an important tool in bridging theoretical knowledge with practical implementation, thus helping to address one of the most striking computer science education challenges uniting theory and practice [72]. Other issues concerning implementation include technical resource requirements and expertise [73]. Also, educators must balance AI support and skills development independently [67]. Other ethical problems arise, such as data protection and algorithmic bias. This present study extends the theory of flipped learning [74], [75] and developmental theory [69], [76] to show that artificial intelligence can effectively scaffold learning. The results support the ACT theory and the feedback model [77]; the TPACK framework is extended to show the ability of AI to support the combination of technological and pedagogical knowledge when teaching programming.

The present study relies on the basic notions of cognitive load theory and constructivist learning principles. It echoes the working of the cognitive model during the programming knowledge learning process by managing intrinsic, extraneous, and germane cognitive loads with great finesse [78]. It is further instigated by the metacognitive scaffolding provided by the AI system in concert with the Zone of Proximal Development framework [69], whereby learners receive assistance in treading the pathway via incrementally complex programming assignments. Thus, their role has been reflected in how the AI chatbots instigate collaborative knowledge construction through dynamic interaction and prompt feedback. The continuous improvement in one's ability to program (from basic syntax to complex debugging skills) can be used as an example to demonstrate how the Digital Taxonomy of Bloom can be applied [79]. The lower reliance on tools as skills improve is one of the ways that demonstrates the theory of Cognitive Apprenticeship [80]. Personalized learning and procedural knowledge are among the examples validating the theories being discussed [81]. Incorporating the above frameworks into one study enables constructing a solid theoretical background for research on programming classes supported by AI technologies.

Empirical findings correspond with recent studies on the effectiveness of intelligent adaptive learning and tutoring systems. Autonomy in problem-solving facilitated through artificial intelligence in relation to eLearning can be justified through Self-Regulated Learning Theory constructs [82]. The increased level of involvement in learning activities among students can be seen as evidence that is congruent with the Technology Acceptance Model [83]. The fact that the AI system has an ability to adjust itself to an individual approach confirms the theoretical assumptions underlying the Learning Analytics Framework. The relationship between technological tool use and the development of new knowledge can be explained by Connectivist Learning Theory [84].

Instructional Design Implications Multiple implications of using chatbot-integrated AI technology in programming lessons can be derived from this study. First, course learning goals are supposed to serve as a necessary prerequisite base for creating a suitable approach to chatbot deployment in programming classes. Second, individualized responses need to be considered as a critical element that enables the process of personalizing the learning experience. Third, sophisticated natural language processing must be provided for relevant interactions. Finally, complete integration and optimization of LMS in addition to strict policies for data privacy are highlighted as vital factors in this regard. All of them can be considered within a particular sequence predetermined by the process of teaching. In order to avoid dependency on artificial intelligence in the future, one should make sure to maintain a balance between educational methodologies. Monitoring measures need to be developed and used on an ongoing basis. Instructor training is also seen as a must. Iterative refinements based on learner performance should become routine practice. The chosen sample size (60 learners) limits the scope of the generalizability of results. Quasi experimental research can produce valid results, but they cannot be statistically generalized to other situations [57]. Future research is thus advised to involve more participants.

Six weeks of treatment also appears as a possible shortcoming. Extended interventions are deemed necessary to adequately assess the skills-acquisition dynamics [77]. Future longitudinal studies may be considered productive in that respect. Persistence of treatment effects beyond one semester also needs further analysis. Variability in AI chatbot architecture, including both rule-based and machine learning-based solutions, should also be taken into account. Personalization capacities and corresponding outcomes constitute another important topic of future research. As for the directions in which this methodology could be applied, it might prove useful to test its efficiency in mathematics, physics, and engineering courses. Moreover, its effectiveness across different academic levels should be explored (vocational programs, undergraduate curricula, graduate education). Cognitive processes in such an environment and adaptation strategies of instructors are also among the topics to consider. Ethics issues, such as data protection and effects of AI technology on critical thinking development, need dedicated empirical research.

CONCLUSION

The use of an AI-based chatbot proved effective in resolving scalability issues related to personalization in instructions. Adaptation of scaffoldings were given and adjusted based on the personalized path of each participant throughout the period of experimentation. It was seen that the performance of the experimental group improved significantly in terms of debugging, which is due to AI-based feedback. This feedback was designed to help learners overcome their shortcomings in procedural knowledge. Experimental results proved the two hypotheses of the study true. It indicates that the contribution of this research to educational technology is many-sided: it provided an example of how theory can be combined with practice within the AI-enhanced flipped classroom in computer science education and, considering the fast development of the subject matter and the necessity of adaptive ways of instruction. This finding suggests that AI-driven personalised learning works and builds a case for studies on adaptive systems. A strong correlation between personalised learning scores and procedural knowledge shows that tailored instruction works even more in complex learning.

It indicates that AI-enhanced flipped classrooms have a potentiality beyond computer programming education, at least concerning ongoing challenges within the STEM area of providing support on an individual basis, developing problem-solving skills, and ensuring autonomy in learning. The following study reviews how schools can put AI into use in learning for a smooth shift to give the necessary preparatory training to the students to meet the changing technological landscape. The investigation is based on the state of the art concerning AI in education, considering the sample size limitation and period. AI-enhanced flipped classrooms bear great potential to bring computer science educational provision up to more adaptive and learner-centred models. By further refining such approaches, AI's full potential in education will be realised for students' better acquisition of basic skills in the technology-dominated environment.

APPENDIX

Sample Cognitive Test Items For Procedural Knowledge Assessment

The cognitive test consisted of 30 multiple-choice items. Each item was designed to assess a specific aspect of procedural knowledge in computer programming. Items were validated by three subject matter experts and piloted with 30 students prior to the main study (Cronbach's alpha = 0.83). Sample items from each procedural knowledge domain are presented below.

Domain 1: Programming Syntax and Data Types
Item 1:
Which of the following correctly declares an integer variable named score and assigns the value 85 in Python?
A. `int score = 85`
B. `score = 85`
C. `score: int = 85` (*correct*)
D. `declare score = 85`
Cognitive process assessed: knowledge of variable declaration syntax

Domain 2: Control Structures
Item 2:
Examine the following code fragment:

```
x = 10
```

```
if x > 5:
    print("High")
elif x == 5:
    print("Medium")
else:
    print("Low")
```

What will be printed when this code is executed?
A. High (*correct*)
B. Medium
C. Low
D. Nothing is printed
Cognitive process assessed: tracing conditional execution sequence

Domain 3: Functions and Procedures
Item 3:
A function named `calculate_area` is defined as follows:

```
def calculate_area(length, width):
    return length * width
```

Which of the following correctly calls this function and stores the result in a variable named `area`?
A. `calculate_area = area(5, 3)`
B. `area = calculate_area(5, 3)` (*correct*)
C. `def area = calculate_area(5, 3)`
D. `area = calculate_area`
Cognitive process assessed: procedural application of function invocation syntax and return value assignment

Domain 4: Iterative Structures
Item 4:
What is the total output produced by the following loop?

```
total = 0
for i in range(1, 6):
    total += i
print(total)
```

A. 10
B. 14
C. 15 (***correct***)
D. 21
Cognitive process assessed: tracing iterative accumulation procedures

Domain 5: Debugging and Error Identification
Item 5:
The following code contains one error. Identify the line that causes the error:

```
def greet(name):            # Line 1
    message = "Hello" + name  # Line 2
    return message          # Line 3

result = greet(123)         # Line 4
print(result)               # Line 5
```

A. Line 1
B. Line 2
C. Line 3
D. Line 4 (***correct***)
Cognitive process assessed: identification of type error in argument passing; procedural error diagnosis

Domain 6: Arrays and List Operations
Item 6:
Given the list below, which statement correctly retrieves the value 30?

```
numbers = [10, 20, 30, 40, 50]
```

A. `numbers = 30`
B. `print(numbers)` (***correct***)
C. `numbers(3)`
D. `get(numbers)`

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Cognitive process assessed: procedural application of list indexing operations

REFERENCES

[1] K. D. Boom, M. Bower, J. Siemon, and A. Arguel, “Relationships between computational thinking and the quality of computer programs,” *Educ. Inf. Technol. (Dordr).*, vol. 27, no. 6, pp. 8289–8310, Jul. 2022, doi: 10.1007/s10639-022-10921-z.

[2] Y. Hwang and C. Lim, “Development of instructional design principles for using ICT in resource-limited learning environments: a case of Bangladesh,” *Asia Pacific Education Review*, vol. 25, no. 5, pp. 1465–1481, Dec. 2024, doi: 10.1007/s12564-024-09996-9.

[3] Z. Kong, “Innovative Flipped Classroom Strategy by Computational Thinking,” *Advances in Educational Technology and Psychology*, vol. 2, no. 1, 2018, doi: 10.23977/aetp.2018.21028.

[4] X. Gao and K. F. Hew, “A Flipped Systematic Debugging Approach to Enhance Elementary Students’ Program Debugging Performance and Optimize Cognitive Load,” *Journal of Educational Computing Research*, vol. 61, no. 5, pp. 1064–1095, Sep. 2023, doi: 10.1177/07356331221133560.

[5] D. Sun, A. Boudouaia, C. Zhu, and Y. Li, “Would ChatGPT-facilitated programming mode impact college students’ programming behaviors, performances, and perceptions? An empirical study,” *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, 2024, doi: 10.1186/s41239-024-00446-5.

[6] C. Maya Díaz, J. Iglesias Sigüenza, and X. Giménez, “Synchronous flipped classroom in stem subjects,” *Revista de Educacion*, vol. 2021, no. 391, pp. 15–39, 2021, doi: 10.4438/1988-592X-RE-2021-391-469.

[7] R. Idsardi, I. Friedly, J. Mancinelli, N. Usai, and L. F. Matos, “Outcomes of Early Adopters Implementing the Flipped Classroom Approach in Undergraduate STEM Courses,” *J. Sci. Educ. Technol.*, vol. 32, no. 5, pp. 655–670, 2023, doi: 10.1007/s10956-023-10066-9.

[8] K. Naser, S. Alamassi, Z. Shana, J. Yousef, and S. H. Halili, “Designing of a Flipped STEM Classroom Engineering-Based Module: Fuzzy Delphi Approach,” *International Journal of Interactive Mobile Technologies*, vol. 17, no. 10, pp. 4–29, 2023, doi: 10.3991/ijim.v17i10.38217.

[9] A. D. Herlambang, R. Ririn, and A. Rachmadi, “The flipped-classroom effect on vocational high school students’ learning outcomes,” *International Journal of Evaluation and Research in Education (IJERE)*, vol. 13, no. 3, p. 1807, Jun. 2024, doi: 10.11591/ijere.v13i3.26757.

[10] H. Atun and E. Usta, “The effects of programming education planned with TPACK framework on learning outcomes,” *Participatory Educational Research*, vol. 6, no. 2, pp. 26–36, 2019, doi: 10.17275/per.19.10.6.2.

[11] W. R. A. Bin-Hady, J. K. M. Ali, and M. A. Al-humari, “The effect of ChatGPT on EFL students’ social and emotional learning,” *Journal of Research in Innovative Teaching and Learning*, vol. 17, no. 2, pp. 243–255, 2024, doi: 10.1108/JRIT-02-2024-0036.

[12] F. G. Deriba, I. T. Sanusi, and A. O. Sunday, “Enhancing Computer Programming Education using ChatGPT: A Mini Review,” 2023. doi: 10.1145/3631802.3631848.

[13] C. A. G. da Silva, F. N. Ramos, R. V. de Moraes, and E. L. dos Santos, “ChatGPT: Challenges and Benefits in Software Programming for Higher Education,” *Sustainability (Switzerland)*, vol. 16, no. 3, 2024, doi: 10.3390/su16031245.

[14] B. Banic, M. Konecki, and M. Konecki, “Pair Programming Education Aided by ChatGPT,” 2023. doi: 10.23919/MIPRO57284.2023.10159727.

[15] M. Urban *et al.*, “ChatGPT improves creative problem-solving performance in university students: An experimental study,” *Comput. Educ.*, vol. 215, 2024, doi: 10.1016/j.compedu.2024.105031.

[16] M. Wang, M. Wang, X. Xu, L. Yang, D. Cai, and M. Yin, “Unleashing ChatGPT’s Power: A Case Study on Optimizing Information Retrieval in Flipped Classrooms via Prompt Engineering,” *IEEE Transactions on Learning Technologies*, vol. 17, pp. 629–641, 2024, doi: 10.1109/TLT.2023.3324714.

[17] R. Avello-Martínez, T. Gajderowicz, and V. G. Gómez-Rodríguez, “Is ChatGPT helpful for graduate students in acquiring knowledge about digital storytelling and reducing their cognitive load? An experiment.,” *Revista de Educación a Distancia*, vol. 24, no. 78, 2024, doi: 10.6018/red.604621.

[18] T. Li, Y. Ji, and Z. Zhan, “Expert or machine? Comparing the effect of pairing student teacher with in-service teacher and ChatGPT on their critical thinking, learning performance, and cognitive load in an integrated-STEM course,” *Asia Pacific Journal of Education*, vol. 44, no. 1, pp. 45–60, 2024, doi: 10.1080/02188791.2024.2305163.

[19] K. Govindaraja *et al.*, “Enhancing Education with ChatGPT: Revolutionizing Personalized Learning and Teacher Support,” *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024, doi: 10.4108/eetiot.6998.

[20] R. N. Albdrani and A. A. Al-Shargabi, “Investigating the Effectiveness of ChatGPT for Providing Personalized Learning Experience: A Case Study,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, pp. 1208–1213, 2023, doi: 10.14569/IJACSA.2023.01411122.

[21] T. W. Kim, "Application of artificial intelligence chatbots, including ChatGPT, in education, scholarly work, programming, and content generation and its prospects: a narrative review," 2023. doi: 10.3352/jeehp.2023.20.38.

[22] P. Denny, B. A. Becker, J. Leinonen, and J. Prather, "Chat Overflow: Artificially Intelligent Models for Computing Education - renAIssance or apocAIypse?," in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, New York, NY, USA: ACM, Jun. 2023, pp. 3–4. doi: 10.1145/3587102.3588773.

[23] M. A. Calsin, M. Aedo, and E. Castro, "Impact of ChatGPT on Teaching: A Flipped Classroom Approach for Programming Fundamentals," *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, vol. 2023, no. 52, pp. 97–112, 2023, doi: 10.17013/risti.52.97-112.

[24] G. Huesca *et al.*, "Effectiveness of Using ChatGPT as a Tool to Strengthen Benefits of the Flipped Learning Strategy," *Educ. Sci. (Basel).*, vol. 14, no. 6, 2024, doi: 10.3390/educsci14060660.

[25] A. M. M. Gasaymeh and R. M. Almohtadi, "The Effect of Flipped Interactive Learning (FIL) Based on ChatGPT on Students' Skills in a Large Programming Class," *International Journal of Information and Education Technology*, vol. 14, no. 11, pp. 1516–1522, 2024, doi: 10.18178/ijiet.2024.14.11.2182.

[26] G. Asiksoy and S. Canbolat, "The Effects of the Gamified Flipped Classroom Method on Petroleum Engineering Students' Pre-class Online Behavioural Engagement and Achievement," *International Journal of Engineering Pedagogy (iJEP)*, vol. 11, no. 5, p. 19, Oct. 2021, doi: 10.3991/ijep.v11i5.21957.

[27] A. D. Herlambang, O. D. Fransisca, and T. Afirianto, "The Flipped-Classroom Instructional Procedure Development and Its Implementation Effectiveness in Improving Procedural Knowledge Learning Outcomes at Vocational High Schools," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 8, no. 2, pp. 2477–2399, 2023.

[28] B. J. Zimmerman, "Becoming a Self-Regulated Learner: An Overview," *Theory Pract.*, vol. 41, no. 2, pp. 64–70, May 2002, doi: 10.1207/s15430421tip4102_2.

[29] J. C. Chen and C. Y. Liu, "Using knowledge building and flipped learning to enhance students' learning performance in a hands-on STEM activity," *J. Comput. Assist. Learn.*, vol. 40, no. 6, pp. 3474–3485, 2024, doi: 10.1111/jcal.13069.

[30] J. Jovanović, D. Gašević, S. Dawson, A. Pardo, and N. Mirriahi, "Learning analytics to unveil learning strategies in a flipped classroom," *Internet and Higher Education*, vol. 33, pp. 74–85, 2017, doi: 10.1016/j.iheduc.2017.02.001.

[31] E. M. W. Ng, "Integrating self-regulation principles with flipped classroom pedagogy for first year university students," *Comput. Educ.*, vol. 126, pp. 65–74, 2018, doi: 10.1016/j.compedu.2018.07.002.

[32] J. M. Campillo-Ferrer and P. Miralles-Martínez, "Effectiveness of the flipped classroom model on students' self-reported motivation and learning during the COVID-19 pandemic," *Humanit. Soc. Sci. Commun.*, vol. 8, no. 1, 2021, doi: 10.1057/s41599-021-00860-4.

[33] C. K. Lo and K. F. Hew, "A comparison of flipped learning with gamification, traditional learning, and online independent study: the effects on students' mathematics achievement and cognitive engagement," *Interactive Learning Environments*, vol. 28, no. 4, pp. 464–481, May 2020, doi: 10.1080/10494820.2018.1541910.

[34] Y. Shi, Y. Ma, J. MacLeod, and H. H. Yang, "College students' cognitive learning outcomes in flipped classroom instruction: a meta-analysis of the empirical literature," *Journal of Computers in Education*, vol. 7, no. 1, pp. 79–103, 2020, doi: 10.1007/s40692-019-00142-8.

[35] J. Jung, Y. Shin, H. J. Chung, and M. Fanguy, "The effects of pre-training types on cognitive load, self-efficacy, and problem-solving in computer programming," *J. Comput. High. Educ.*, 2024, doi: 10.1007/s12528-024-09407-3.

[36] K. Thongkoo, P. Panjaburee, and K. Daungcharone, "Integrating inquiry learning and knowledge management into a flipped classroom to improve students' web programming performance in higher education," *Knowledge Management and E-Learning*, vol. 11, no. 3, pp. 304–324, 2019, doi: 10.34105/j.kmel.2019.11.016.

[37] J. Pattanaphanchai, "An investigation of students' learning achievement and perception using flipped classroom in an introductory programming course: A case study of Thailand higher education," *Journal of University Teaching and Learning Practice*, vol. 16, no. 5, 2019, doi: 10.53761/1.16.5.4.

[38] E. Namaziandost and F. Çakmak, "An account of EFL learners' self-efficacy and gender in the Flipped Classroom Model," *Educ. Inf. Technol. (Dordr).*, vol. 25, no. 5, pp. 4041–4055, 2020, doi: 10.1007/s10639-020-10167-7.

[39] B. Li and M. Peng, "Integration of an AI-Based Platform and Flipped Classroom Instructional Model," *Sci. Program.*, vol. 2022, 2022, doi: 10.1155/2022/2536382.

[40] P. Meng, "An AI-enhanced teaching model using a novel deep learning approach," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, Jan. 2024, doi: 10.2478/amns-2024-2567.

[41] M. Cullen and M. Kirkpatrick, "Embracing Artificial Intelligence: Incorporating Artificial Intelligence Into Classroom Instruction," *Journal of Nursing Education*, pp. 1–2, Oct. 2024, doi: 10.3928/01484834-20240626-01.

[42] H. Peng, S. Ma, and J. M. Spector, "Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment," *Smart Learning Environments*, vol. 6, no. 1, p. 9, Dec. 2019, doi: 10.1186/s40561-019-0089-y.

[43] D. Ifenthaler and J. Y. K. Yau, "Utilising learning analytics to support study success in higher education: a systematic review," *Educational Technology Research and Development*, vol. 68, no. 4, pp. 1961–1990, Aug. 2020, doi: 10.1007/s11423-020-09788-z.

[44] J. Pane, E. Steiner, M. Baird, and L. Hamilton, *Continued Progress: Promising Evidence on Personalized Learning*. RAND Corporation, 2015. doi: 10.7249/rr1365.

[45] C. Walkington and M. L. Bernacki, "Personalization of Instruction: Design Dimensions and Implications for Cognition," *The Journal of Experimental Education*, vol. 86, no. 1, pp. 50–68, Jan. 2018, doi: 10.1080/00220973.2017.1380590.

[46] M. S. Knowles, *Self-directed learning: A guide for learners and teachers*. New York: Association Press, 1975.

[47] L. Feng, "Investigating the Effects of Artificial Intelligence-Assisted Language Learning Strategies on Cognitive Load and Learning Outcomes: A Comparative Study," *Journal of Educational Computing Research*, vol. 62, no. 8, pp. 1961–1994, Jan. 2024, doi: 10.1177/07356331241268349.

[48] J. Nouri, L. Zhang, L. Mannila, and E. Norén, "Development of computational thinking, digital competence and 21st century skills when learning programming in K-9," *Education Inquiry*, vol. 11, no. 1, pp. 1–17, 2020, doi: 10.1080/20004508.2019.1627844.

[49] D. Sun, F. Ouyang, Y. Li, and C. Zhu, "Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education," *Int. J. STEM Educ.*, vol. 8, no. 1, 2021, doi: 10.1186/s40594-021-00311-1.

[50] H. Yildiz Durak, "The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving," *Technology, Knowledge and Learning*, vol. 25, no. 1, pp. 179–195, 2020, doi: 10.1007/s10758-018-9391-y.

[51] D. Hurrell, "Conceptual Knowledge OR Procedural Knowledge or Conceptual Knowledge AND Procedural Knowledge: Why the Conjunction is Important to Teachers," *Australian Journal of Teacher Education*, vol. 46, no. 2, pp. 57–71, 2021, doi: 10.14221/ajte.2021v46n2.4.

[52] A. Luxton-Reilly *et al.*, "Developing Assessments to Determine Mastery of Programming Fundamentals," in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA: ACM, Jun. 2017, pp. 388–388. doi: 10.1145/3059009.3081327.

[53] M. Martínez-Comesaña, X. Rigueira-Díaz, A. Larrañaga-Janeiro, J. Martínez-Torres, I. Ocarranza-Prado, and D. Kreibel, "Impact of artificial intelligence on assessment methods in primary and secondary education: Systematic literature review," *Revista de Psicodidáctica (English ed.)*, vol. 28, no. 2, pp. 93–103, Jul. 2023, doi: 10.1016/j.psicoe.2023.06.002.

[54] S. Caspari-Sadeghi, "Artificial Intelligence in Technology-Enhanced Assessment: A Survey of Machine Learning," *Journal of Educational Technology Systems*, vol. 51, no. 3, pp. 372–386, Mar. 2023, doi: 10.1177/00472395221138791.

[55] A. Zirar, "Exploring the impact of language models, such as ChatGPT, on student learning and assessment," *Review of Education*, vol. 11, no. 3, 2023, doi: 10.1002/rev3.3433.

[56] J. W. Creswell and T. C. Guetterman, *Educational research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. New Jersey: Pearson, 2019.

[57] J. R. Fraenkel, N. E. Wallen, and H. H. Hyun, *How to Design and Evaluate Research in Education*. New York: McGraw Hill LLC, 2022. Accessed: Jun. 27, 2023. [Online]. Available: https://www.mheducation.com/highered/product/how-design-evaluate-research-education-fraenkel-wallen/M9781259913839.html

[58] E. A. Stuart, "Matching methods for causal inference: A review and a look forward," *Statistical Science*, vol. 25, no. 1, pp. 1–21, Feb. 2010, doi: 10.1214/09-STS313.

[59] P. C. Austin, "An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies," *Multivariate Behav. Res.*, vol. 46, no. 3, pp. 399–424, May 2011, doi: 10.1080/00273171.2011.568786.

[60] Z. Zhang, H. J. Kim, G. Lonjon, and Y. Zhu, "Balance diagnostics after propensity score matching," *Ann. Transl. Med.*, vol. 7, no. 1, pp. 16–16, Jan. 2019, doi: 10.21037/atm.2018.12.10.

[61] F. Faul, E. Erdfelder, A. G. Lang, and A. Buchner, "G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behav. Res. Methods*, vol. 39, no. 2, pp. 175–191, 2007, doi: 10.3758/BF03193146.

[62] K. F. Hew and C. K. Lo, "Flipped classroom improves student learning in health professions education: A meta-analysis," *BMC Med. Educ.*, vol. 18, no. 1, 2018, doi: 10.1186/s12909-018-1144-z.
[63] M. L. McHugh, "Interrater reliability: The kappa statistic," *Biochem. Med. (Zagreb).*, vol. 22, no. 3, pp. 276–282, 2012, doi: 10.11613/bm.2012.031.
[64] B. Waldrip *et al.*, "Personalised Learning Environment Questionnaire," Jul. 10, 2014. doi: 10.1037/t61484-000.
[65] G. A. Miller and J. P. Chapman, "Misunderstanding analysis of covariance," *J. Abnorm. Psychol.*, vol. 110, no. 1, pp. 40–48, Feb. 2001, doi: 10.1037/0021-843X.110.1.40.
[66] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. New York: Lawrence Erlbaum Associates, 1988.
[67] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education – where are the educators?," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, p. 39, Dec. 2019, doi: 10.1186/s41239-019-0171-0.
[68] T. T. A. Ngo, "Perception of Engineering Students on Social Constructivist Learning Approach in Classroom," *International Journal of Engineering Pedagogy*, vol. 14, no. 1, pp. 20–38, 2024, doi: 10.3991/ijep.v14i1.43101.
[69] K. Shabani, M. Khatib, and S. Ebadi, "Vygotsky's Zone of Proximal Development: Instructional Implications and Teachers' Professional Development," *English Language Teaching*, vol. 3, no. 4, Nov. 2010, doi: 10.5539/elt.v3n4p237.
[70] C. K. Lo, K. F. Hew, and M. S. yung Jong, "The influence of ChatGPT on student engagement: A systematic review and future research agenda," *Comput. Educ.*, vol. 219, 2024, doi: 10.1016/j.compedu.2024.105100.
[71] H. Keuning, J. Jeuring, and B. Heeren, "A systematic literature review of automated feedback generation for programming exercises," *ACM Transactions on Computing Education*, vol. 19, no. 1, pp. 1–43, Mar. 2018, doi: 10.1145/3231711.
[72] T. Gorichanaz, "Accused: How students respond to allegations of using ChatGPT on assessments," *Learning: Research and Practice*, vol. 9, no. 2, pp. 183–196, 2023, doi: 10.1080/23735082.2023.2254787.
[73] R. Winkler and M. Soellner, "Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis," *Academy of Management Proceedings*, vol. 2018, no. 1, p. 15903, Aug. 2018, doi: 10.5465/AMBPP.2018.15903abstract.
[74] J. Bergmann and A. Sams, *Flip Your Classroom: Reach Every Student in Every Class Every Day*. in Flipped Learning Series. International Society for Technology in Education, 2012. [Online]. Available: https://books.google.co.id/books?id=nBi2pwAACAAJ
[75] J. Bergmann and A. Sams, *Flipped Learning: Gateway to Student Engagement*. Washington DC: International Society for Technology in Education, 2014.
[76] L. S. Vygotsky, *Social Constructivism - Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978. [Online]. Available: https://autismusberatung.info/wp-content/uploads/2023/09/Vygotsky-Mind-in-society.pdf
[77] J. Hattie and H. Timperley, "The power of feedback," *Rev. Educ. Res.*, vol. 77, no. 1, pp. 81–112, Mar. 2007, doi: 10.3102/003465430298487.
[78] Ü. Çakiroğlu and Ş. Bilgi, "Exploring intrinsic cognitive load in the programming process: a two dimensional approach based on element interactivity," *Interactive Learning Environments*, 2022, doi: 10.1080/10494820.2022.2137527.
[79] E. H. S. Y. Elim, "Promoting cognitive skills in AI-supported learning environments: the integration of bloom's taxonomy," *Educ. 3 13*, pp. 1–11, Apr. 2024, doi: 10.1080/03004279.2024.2332469.
[80] V. Mekthanavanh, "ChatGPT Enhances Programming Skills of Computer Engineering Students," *Souphanouvong University Journal Multidisciplinary Research and Development*, vol. 10, no. 1, pp. 1–9, Sep. 2024, doi: 10.69692/SUJMRD100101.
[81] S. Hobert, "Fostering skills with chatbot-based digital tutors - Training programming skills in a field study," *I-Com*, vol. 22, no. 2, pp. 143–159, Aug. 2023, doi: 10.1515/icom-2022-0044.
[82] I. Molenaar, "The concept of hybrid human-AI regulation: Exemplifying how to support young learners' self-regulated learning," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100070, 2022, doi: 10.1016/j.caeai.2022.100070.
[83] G. Yun, K. M. Lee, and H. H. Choi, "Empowering Student Learning Through Artificial Intelligence: A Bibliometric Analysis," *Journal of Educational Computing Research*, vol. 62, no. 8, pp. 2042–2075, Jan. 2024, doi: 10.1177/07356331241278636.
[84] F. Kamalov, D. Santandreu Calonge, and I. Gurrib, "New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution," *Sustainability*, vol. 15, no. 16, p. 12451, Aug. 2023, doi: 10.3390/su151612451.