Online-Journals.org

iJEP International Journal of Engineering Pedagogy

iJEP | eISSN: 2192-4880 | Vol. 16 No. 5 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijep.v16i5.62296

PAPER

Achievement Goal Motivation Influences Self-Efficacy in AI-Supported Collaborative Programming: Mediating Roles of Growth Mindset and Mastery Experience

Wen-Shyong Tzou[1],
Chia-Hsiang Ma[2],
Yu-chu Yeh[2] (✉),
Melissa T. A. Simarmata[2]

[1]National Taiwan Ocean University, Keelung, Taiwan

[2]National Chengchi University, Taipei City, Taiwan

ycyeh@nccu.edu.tw

ABSTRACT

This study designed an AI-integrated collaborative problem-based learning (CPBL) course aimed to bolster programming self-efficacy among undergraduate and to elucidate the underlying psychological mechanisms that drive self-efficacy. A total of 179 Taiwanese undergraduates participated in an 18-week intervention that integrated AI toolkits, CPBL frameworks, and instructional scaffolding to facilitate interdisciplinary application. To achieve our goals, two instruments were developed to measure programming-specific achievement goals and mindsets. The results indicated that the AI-supported CPBL environment effectively fostered a growth mindset, mastery experiences, and self-efficacy in programming. Mediation model analysis revealed that growth mindset and mastery experience functioned as dual mediators linking achievement goal orientation to self-efficacy. Notably, mastery goals exerted a markedly stronger influence on self-efficacy compared to performance goals. The findings highlight the necessity of prioritizing mastery-oriented pedagogy in the design of AI-integrated programming curricula.

KEYWORDS

achievement goal, growth mindset, mastery experience, self-efficacy, programming

1 INTRODUCTION

In recent years, programming education has attracted significant global attention in many developed countries [1], particularly in the current digital era, where programming ability has become an essential interdisciplinary competence [2]. Despite its importance, programming is characterized by a steep learning curve and high cognitive demand. Even among computer science majors, the complexity of syntax and logical abstraction often leads to significant attrition or course withdrawal [3]. This challenge is further compounded for non-specialist students, who may lack

Tzou, W.-S., Ma, C.-H., Yeh, Y., Simarmata, M. T. A. (2026). Achievement Goal Motivation Influences Self-Efficacy in AI-Supported Collaborative Programming: Mediating Roles of Growth Mindset and Mastery Experience. *International Journal of Engineering Pedagogy (iJEP)*, 16(5), pp. 125–143. https://doi.org/10.3991/ijep.v16i5.62296

Article submitted 2026-05-12. Revision uploaded 2026-07-14. Final acceptance 2026-07-20.

the foundational mental models required for rapid proficiency. It has been found that non–computer science majors tend to exhibit low programming self-efficacy [4]. Developing programming self-efficacy [5] is therefore critical for fostering programming competence. To cultivate learners' programming self-efficacy, both instructional factors and learners' psychological characteristics must be considered. By understanding key psychological mechanisms, instructors can design more effective training that enhances learners' resilience, confidence, and long-term success in programming education.

To improve the effectiveness of programming instruction, research has increasingly focused on the development of instructional tools and techniques [6]. In response, this study developed a training course integrating AI toolkits into collaborative project-based learning (CPBL). The goal was to explore the psychological factors that may influence the development of programming self-efficacy in an AI-supported learning environment. To ensure the robustness of the learning environment, training effects were also examined.

This study focused on three psychological factors: achievement goals, growth mindset, and mastery experience. Achievement goal theory is one of the most prominent and empirically supported frameworks for understanding learning motivation [7]. Growth mindset refers to the belief that one's abilities can be developed over time [8]. A strong achievement goal orientation may foster a growth mindset. Together, these two motivation-related traits can influence the development of mastery experiences. Furthermore, mastery experience is considered the most powerful source for enhancing self-efficacy [9], [10]. Based on these theoretical connections, we hypothesize that growth mindset and mastery experience serve as mediators between achievement goals and programming self-efficacy in an AI-supported learning context.

Achievement goals are generally categorized into mastery goals, which focus on improving one's own competence, and performance goals, which focus on demonstrating competence relative to others [7]. While mastery goals are consistently found to benefit learning, the effects of performance goals are more variable [11]. Accordingly, this study examines how different achievement goal orientations influence programming self-efficacy through the mediating roles of growth mindset and mastery experience in an AI-supported Python programming environment.

2 LITERATURE REVIEW

2.1 Constructs of achievement goals

Achievement goal theory generally distinguishes between two primary goal orientations. Mastery goals emphasize learning and self-improvement. Learners with mastery goals tend to adopt deep learning strategies and persist in the face of challenges. In contrast, performance goals focus on demonstrating competence relative to others. Learners with performance goals often prioritize outcomes and may avoid challenging tasks to protect their self-image [12], [13].

To extend the explanatory power of the theory, Elliot and McGregor [14] proposed a 2 × 2 model that differentiates between approach and avoidance orientations within both mastery and performance goals. The four resulting categories are: (1) Mastery-approach (e.g., striving to fully understand learning content); (2) Mastery-avoidance (e.g., striving to avoid missing important concepts);

(3) Performance-approach (e.g., striving to outperform others); and (4) Performance-avoidance (e.g., trying to avoid being judged poorly for poor performance). Among these orientations, mastery-approach goals are most consistently associated with adaptive outcomes, including higher self-efficacy, greater intrinsic motivation, and sustained academic achievement [15], [16].

Research has shown that achievement goal orientation plays a crucial role in computer-supported collaborative learning [17]. Furthermore, peer cooperation and competition have both been found to be positively related to mastery-approach goals [18]. Although achievement goal theory has been widely applied across educational domains, its application in programming education remains limited. To address this gap, the present study incorporates achievement goals to examine their role in programming learning. For this purpose, we first developed an inventory to measure programming achievement goals.

2.2 Constructs of mindsets

Mindset theory offers a valuable framework for understanding how students interpret challenges and respond to setbacks [19]. Originally, the concept of mindset referred to an individual's beliefs about intelligence [8]. Two core beliefs define this construct: a fixed mindset, in which intelligence is viewed as static, and a growth mindset, in which intelligence is seen as malleable and capable of development through effort and effective strategies [8], [20]. Students who endorse a growth mindset are more likely to embrace challenges as opportunities for development [21]. This belief promotes adaptive learning behaviors, such as persistence, self-regulation, and the adoption of effective learning strategies [22]. Furthermore, a growth mindset is positively associated with academic performance [23].

Importantly, mindset is domain-specific; individuals may hold different beliefs across different areas. Consequently, mindset research has expanded to diverse fields, such as creativity [24] and STEM education [25]. However, mindset has rarely been examined in the context of programming education.

To specify the explanatory power of creativity mindsets, Yeh et al. [26] proposed a 2 × 2 model that distinguishes between growth and fixed mindsets with internal and external attributions. The four types of mindsets are as follows: (1) Growth–internal mindset: Belief that ability can be improved through self-directed learning; (2) Growth–external mindset: Belief that ability can be improved through supportive learning environments or assistance from others; (3) Fixed–internal mindset: Belief that ability is an innate talent and cannot be improved through self-directed learning; and (4) Fixed–external mindset: Belief that ability cannot be improved even with strong external support. Based on this theoretical framework, this study developed an inventory to measure programming mindset.

2.3 The relationship between programming achievement goals, growth mindset, mastery experience, and self-efficacy during training

Programming self-efficacy, defined as a learner's belief in their ability to successfully design, structure, implement programming solutions, and solve problems, is a critical factor influencing motivation and achievement in programming education [27], [4]. For students from non-computer science backgrounds, low self-efficacy is often associated with anxiety, disengagement, and even course attrition [28]. Fostering programming self-efficacy is therefore essential for supporting these learners.

Prior research has suggested that programming self-efficacy is associated with students' prior programming experiences and performance outcomes [29], [30]. Meta-analytic evidence also indicates that mastery experience is the most influential source of self-efficacy [31]. Mastery experiences enhance learners' sense of competence and self-confidence, particularly when tasks are iterative, feedback-rich, and demand active problem-solving [32]. In programming education, mastery experience involves acquiring coding skills and applying them to solve real problems. It is, therefore, crucial for building programming self-confidence.

Furthermore, social cognitive theory posits that self-efficacy is shaped by an interplay of personal, behavioral, and environmental influences [33]. Motivation serves as the foundation for learner engagement in programming tasks and shapes their mindset [34]. Research indicates that highly motivated learners, particularly those with a mastery goal orientation, are more likely to adopt a growth mindset, believing that programming skills can be developed through persistence [35]. In the same vein, it is found that students with a growth mindset are more likely to persist through errors and debugging, achieving incremental successes in programming tasks [36]. This mindset, in turn, fosters sustained engagement and the accumulation of mastery experiences, which ultimately strengthen programming self-efficacy.

In summary, the relationship between programming motivation and self-efficacy can be conceptualized as a serial mediation model: achievement goal motivation → growth mindset → persistence and mastery experience → enhanced programming self-efficacy.

2.4 Artificial intelligence-supported CPBL

Collaborative project-based learning promotes both knowledge acquisition and teamwork by engaging learners in real-world problem-solving projects. CPBL emphasizes process-oriented, context-related, and student-centered learning [37]. Research on CPBL has shown that collaboration fosters shared responsibility, motivation, and essential teamwork skills [38]. Active peer interaction enables students to articulate ideas, solve problems collectively, and, in turn, develop confidence and self-efficacy [39].

In programming education, CPBL provides opportunities for guided practice, allowing students to build mastery experiences that strengthen their self-efficacy [40]. This approach is particularly effective for non–computer science majors, as it situates programming within meaningful, interdisciplinary contexts [41], [42]. Moreover, CPBL enables students to collaboratively engage in authentic, complex projects that promote the development of knowledge and skills while systematically completing interconnected learning tasks to achieve the project goals. This approach has been shown to significantly enhance college students' programming performance [37].

On the other hand, the interdisciplinary nature of CPBL further encourages learners to integrate IT and programming skills into broader problem-solving contexts [41], [42]. The integration of AI support into CPBL offers additional benefits. AI can provide instant assistance, personalized feedback, and adaptive guidance, all of which have been shown to enhance programming performance [43]. For example, the use of a generative AI tool, MindScratch, has been found to help learners complete creative programming projects [44]. Moreover, it was found that incorporating

ChatGPT into weekly programming practices improved students' programming self-efficacy [4]. Building on these strengths, the present study developed an AI-supported CPBL framework aimed at enhancing learners' growth mindset, mastery experience, and programming self-efficacy.

2.5 Present study and hypotheses

Given the growing importance of Python programming and the potential of AI-supported CPBL, we designed an 18-week training course, by which we examined how achievement goals influence programming self-efficacy. To achieve these objectives, we developed two instruments: the inventory of programming motivation (IPMo) and the inventory of programming mindsets (IPM).

We further anticipated that different types of achievement goal motivation—specifically, mastery goals and performance goals—might exert distinct effects on self-efficacy. The following hypotheses were proposed:

- H1: Growth mindset would mediate the influence of achievement goal motivation on programming self-efficacy among non–computer science college students.
- H2: Mastery experience would mediate the influence of achievement goal motivation on programming self-efficacy among non–computer science college students.
- H3: Growth mindset and mastery experience would jointly mediate the influence of achievement goal motivation on programming self-efficacy among non–computer science college students.
- H4: Mastery goal motivation would exhibit stronger overall explanatory power in predicting programming self-efficacy compared to performance goal motivation.

Notably, in an exploratory manner, we also examined the training effects on mastery experience and self-efficacy in programming, as well as fixed and growth mindsets toward programming. The effects of achievement goal motivation were not assessed because the study did not include any manipulation of this construct. Instead, we assumed that achievement goal motivation would influence students' learning processes during the training.

3 METHOD

3.1 Participants

A total of 179 college students (88 females, 80 males, and 11 unspecified) who had participated in the inventory development also took part in the training course. All participants were undergraduate students from the School of Life Science, with a mean age of 19.45 years (*SD* = 8.74), enrolled in the liberal education course Programming Language and Data Processing offered by the Department of Bioscience and Biotechnology at a public university in Taiwan. This study was conducted as part of a regular university course. Participation involved no risk beyond normal educational activities. All data were anonymized before analysis. Participation in research analysis was voluntary, and informed consent was obtained from all participants at the outset of the study. Each participant received a reward equivalent to approximately USD 7.

3.2 Instruments

Four instruments were used in this study: the IPMo, the IPM, the *Inventory of Programming Mastery Experience (IPME)*, and the *Inventory of Programming Self-Efficacy (IPSE).* The first two inventories were developed in this study. All instruments employed a 6-point Likert scale ranging from 1 (strongly disagree) to 6 (strongly agree).

IPMo and IPM. The development of the IPMo and IPM involved a series of reliability and validity analyses to assess internal consistency and construct validity. The final version of the IPMo consisted of nine items across two factors: Mastery goal (5 items) and Performance goal (4 items) (refer to Table 1). The IPM comprised ten items across two factors: Growth mindset (5 items) and Fixed mindset (5 items) (refer to Table 2). Details of the scale development process are provided in Section 4.1.

IPME and IPSE. The IPME [45] was used to measure participants' mastery experience in Python programming. The scale was validated through both exploratory and confirmatory factor analyses. It comprised five items across two factors: Foundational ability (3 items) and Advanced ability (2 items). The correlation between the two factors was .810 ($p < .001$), and the correlations between each factor and the total score were .965 and .936 ($ps < .001$). Cronbach's α coefficients were .930 for the overall scale, .909 for Foundational ability, and .857 for Advanced ability. Example items include: "I can apply basic programming syntax in my code" (Foundational ability) and "I can use programming to explore what interests me" (Advanced ability).

The IPSE [45] was used to assess participants' self-efficacy in Python programming. The scale was validated through both exploratory factor analysis (EFA) and confirmatory factor analysis (CFA). It consists of seven items across two factors: Application self-efficacy (4 items) and Learning self-efficacy (3 items). The correlation between the two factors was .843 ($p < .001$), and the correlations between each factor and the total score were .973 and .945 ($ps < .001$). Cronbach's α coefficients indicated excellent internal consistency, with values of .943 for the overall scale, .912 for Application self-efficacy, and .899 for Learning self-efficacy. Example items include: "I believe I can adapt example code to solve similar problems" (Application self-efficacy) and "I believe I can learn and understand basic programming syntax" (Learning self-efficacy).

3.3 Experimental design and procedures

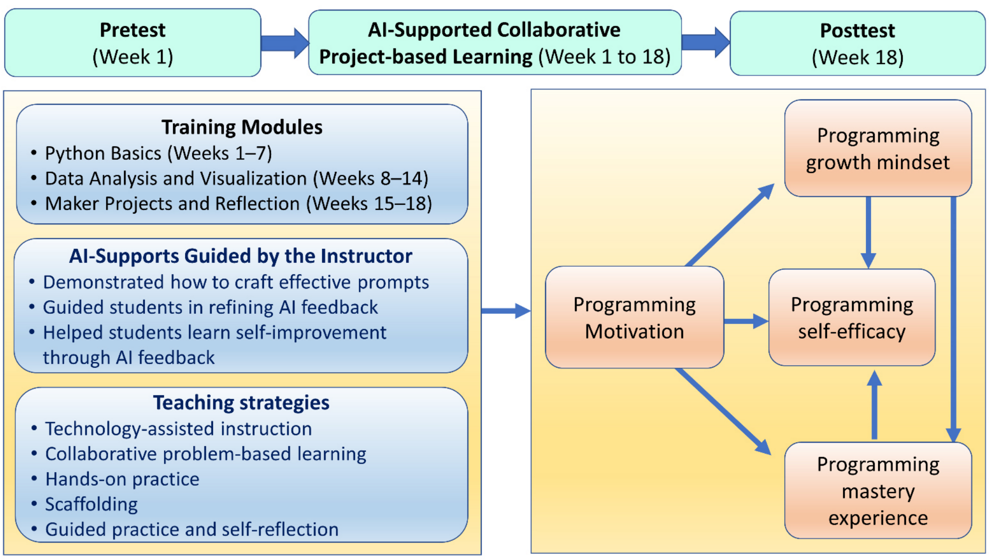

Fig. 1. Experimental design and key features of the instructional design

Data for inventory development were collected via SurveyCake (https://www.surveycake.com), with no time constraints imposed on participants. For the experimental teaching, the study aimed to enhance students' growth mindset, mastery experience, and self-efficacy through an 18-week instructional course. During the first week, participants completed a pretest assessing achievement goal motivation, growth mindset, mastery experience, and self-efficacy. They then engaged in the 18-week training course, after which they completed a posttest containing the same measures as the pretest. We anticipated that the training course would enhance participants' programming motivation, growth mindset, mastery experiences, and self-efficacy. See Figure 1 for the experimental design and the key features of the instructional design.

The course was conducted in a blended learning format via the TronClass teaching platform. The instructor—also the first author—has over 20 years of experience teaching the Programming Language and Data Processing course. The instructional design was organized into three stages:

1. Python basics (Weeks 1–7): This stage introduced fundamental Python syntax and the use of comments. Problem-based learning was emphasized through hands-on coding exercises, weekly assignments, and group discussions.
2. Data analysis and visualization (Weeks 8–14): This stage focused on progressively developing students' data analysis skills, including data visualization with Matplotlib, exploring relationships among variables, and applying linear regression models. Learning activities centered on practical applications of the taught techniques.
3. Maker projects and reflection (Weeks 15–18): This stage implemented CBPL using the Flag's Maker Workshop: Learning Python Through Maker Activities kit. Students engaged in peer sharing, discussion sessions, and reflective practices throughout the project process. Three projects were completed with an ESP32-powered AI maker kit: a Morse Code Communicator, a Lie Detector, and a Pulse Oximeter. The Morse Code Communicator required students to program LED blink patterns to simulate Morse code. The Lie Detector used a galvanic skin response (GSR) sensor to detect changes in skin conductivity, analyzing fluctuations to infer stress levels and potential deception. Finally, the Pulse Oximeter employed an optical sensor to measure blood oxygen saturation (SpO_2) and pulse rate, with real-time data processing and display via the ESP32. Together, these projects showcased how sensing technologies and microcontrollers can be applied in interactive and wearable systems. Examples of applications using the AI kit are illustrated in Figure 2.

To support active learning, the course integrated multiple technologies—including Python, multimedia resources, PowerPoint slides, and a software development kit—aimed at strengthening conceptual understanding, facilitating hands-on practice, promoting collaboration and discussion, and enhancing problem-solving skills and confidence. A central resource was the kit *Flag's Maker Workshop*, which provided structured tasks to increase student engagement and reinforce practical Python skills and self-efficacy through group CPBL. The instructional design followed a scaffolding approach, gradually guiding students from foundational concepts to applied projects and reflection. Notably, throughout the course, students participated in peer sharing, collaborative project development, and in-class presentations.

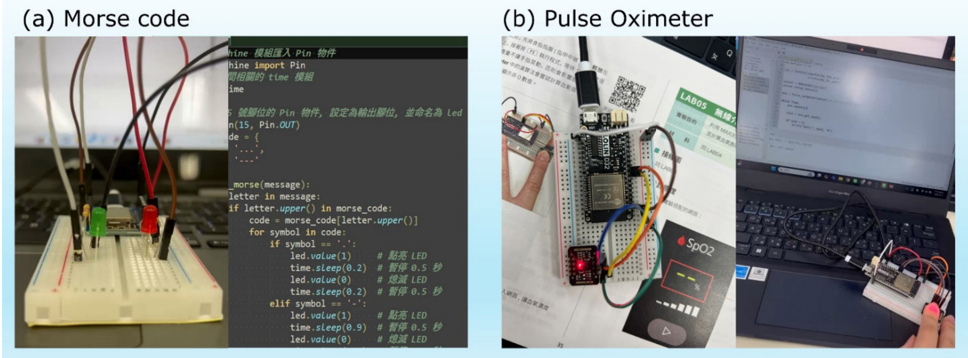

Fig. 2. Examples of applications using the kit of "Flag's Maker Workshop: Learning Python Through Maker Activities"

Notes: The Morse code for SOS was coded into the LED lights to send a message. The letter S (· · ·) is shown by three quick flashes, and the letter O (– – –) is shown by three long flashes. Together, the short and long flashes spell out SOS.

Notably, generative AI—primarily ChatGPT—was systematically integrated to support students' learning in both individual and group assignments. At the beginning of the course, the instructor demonstrated how to formulate effective prompts. Throughout the learning process, the instructor also provided explicit guidance when students were unable to obtain useful responses from the generative AI on their own, particularly during the final Maker Project. In addition, students were taught how to improve their programming skills by interpreting and refining the feedback generated by ChatGPT. Within this AI-supported, collaborative problem-solving environment, students learned to leverage generative AI effectively to enhance and facilitate their learning.

4 RESULTS

4.1 Inventory development

For both inventories, internal consistency reliability and EFA with Promax rotation were conducted to establish reliability and preliminary construct validity. Subsequently, CFA using maximum likelihood estimation, along with inter-factor correlation analyses, was performed to further validate the construct structure.

Development of the IPMo. Exploratory factor analysis results for the IPMo revealed a two-factor structure—Mastery goal and Performance goal—across nine items. Factor loadings ranged from .697 to .966, accounting for 75.606% of the total variance (refer to Table 1). The correlation between the two factors was .575 (p < .001), and the correlations between each factor and the total score were .892 and .883 (ps < .001). Cronbach's α coefficients indicated strong internal consistency, with values of .913 for the overall scale, .897 for the Mastery goal, and .916 for the Performance goal. Item–total correlations ranged from .591 to .788, further supporting the reliability of the scale.

Confirmatory factor analysis further supported the two-factor structure of the IPMo. The model fit indices were: χ^2 (N = 193, df = 26) = 124.151, p < .001; goodness-of-fit index (GFI) = .877; adjusted goodness-of-fit index (AGFI) = .788; standardized root mean square residual (SRMR) = .059; root mean square error of approximation (RMSEA) = .140; comparative fit index (CFI) = .926; and normed fit index (NFI) = .909. The composite reliability (CR) values for the Mastery goal and Performance goal factors were .895 and .917, respectively, and the average variance extracted (AVE) values were .647 for Mastery goal and .738 for Performance goal.

Table 1. Test items and factor loadings of the IPMo

No.	Factors and Items	Factor Loadings	
	During the Learning of Programming	1	2
	Factor 1: Mastery goal (α = .897)		
3	I strive to avoid missing any important concepts during my learning.	.923	
2	I do my best to learn as many important concepts from class as possible.	.893	
6	I make an effort to integrate what I've learned, hoping to apply it effectively.	.876	
1	I try hard not to take a perfunctory approach, so as not to reduce the effectiveness of my learning.	.802	
7	I strive to avoid a superficial understanding that prevents true comprehension and integration.	.697	
	Factor 2: Performance goal (α = .916)		
9	I put in great effort to perform well so others won't look down on me.		.966
5	I work hard to avoid being ridiculed by others.		.912
8	I strive to perform well to impress others.		.908
4	I make an effort to perform well in hopes of gaining others' recognition.		.717

The development of IPM. Exploratory factor analysis revealed a two-factor structure—Fixed Mindset and Growth Mindset—across ten items. Factor loadings ranged from .814 to .902, accounting for 77.283% of the total variance (refer to Table 2). The correlation between the two factors was –.381 (p < .001). The correlations between each factor and the total score were .551 (p < .001) for Fixed Mindset and .562 (p < .001) for Growth Mindset (refer to Table 2). Cronbach's α coefficients indicated excellent internal consistency, with values of .929 for Fixed Mindset and .920 for Growth Mindset. Item–total correlations ranged from .758 to .851 for Fixed Mindset and from .744 to .846 for Growth Mindset.

Confirmatory factor analysis further supported the two-factor structure of the IPM. The model fit indices were: χ^2 (N = 193, df = 34) = 133.550, p < .001; GFI = .871; AGFI = .791; SRMR = .046; RMSEA = .123; CFI = .937; and NFI = .918. CR values were .930 for Fixed Mindset and .919 for Growth Mindset, with AVE values of .702 and .727, respectively.

Table 2. Test items and factor loadings of IPM

No.	Factors and Items	Factor Loadings	
		1	2
	Factor 1: Growth mindsets (α = .920)		
8	If I have good learning opportunities, my programming skills can improve significantly.	.897	
7	As long as I am willing to learn, I can improve my programming skills.	.886	
4	If I have a good learning environment, my programming skills can improve.	.859	
2	With the help of teachers and teaching assistants, my programming skills will improve.	.823	
1	My programming skills can be improved through self-learning, and it's never too late to start learning now.	.814	

(Continued)

Table 2. Test items and factor loadings of IPM *(Continued)*

No.	Factors and Items	Factor Loadings 1	Factor Loadings 2
	Factor 2: Fixed mindsets ($\alpha = .929$)		
6	Even if there is a good learning environment, it's still difficult for me to improve my programming skills.		.902
5	Even if I am willing to learn, it's still hard for me to improve my programming skills.		.879
10	Even if I have great learning opportunities, my programming skills won't improve much.		.869
9	Even if I study very hard on my own, my programming skills won't improve significantly.		.867
3	It's hard for me to improve my programming skills; self-learning doesn't help either.		.815

4.2 Improvement of programming self-efficacy, mastery experience, and mindset

To assess the effects of the training course, repeated measures ANOVA was performed with pretest and posttest scores as within-subject variables for programming self-efficacy, mastery experience, growth mindset, and fixed mindset. Means and standard errors are presented in Figure 3. Results showed significant improvements in the overall and subscale scores of programming self-efficacy, $Fs(1, 166) = 7.538–48.906$, $ps < .01$, $\eta_p^2 = .090–.228$. Participants' growth mindset also significantly increased, $F(1, 166) = 5.248$, $p = .023$, $\eta_p^2 = .031$, whereas their fixed mindset significantly decreased, $F(1, 166) = 4.625$, $p = .033$, $\eta_p^2 = .027$ (refer to Table 3).

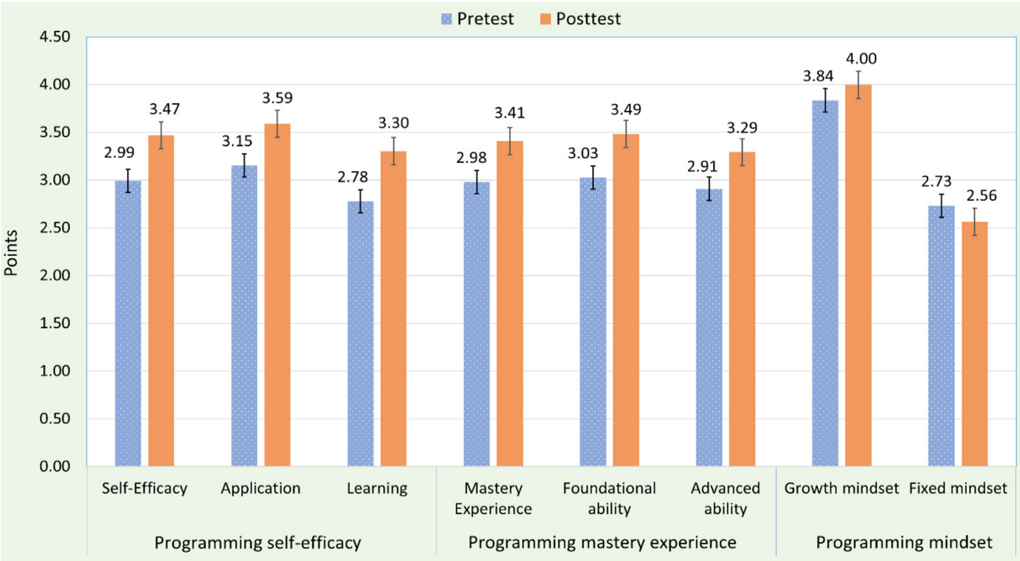

Fig. 3. Means and standard errors of programming self-efficacy, mastery experience, and growth mindset in the pretest and posttest

Table 3. Results of repeated measures ANOVA examining improvements in programming self-efficacy, mastery experience, growth mindset, and fixed mindset

Source	MS	$F(1, 166)$	p	η_p^2	Comparison
Self-Efficacy	18.821	48.906***	.000	.228	Posttest > Pretest
Application	15.955	34.829***	.000	.173	Posttest > Pretest
Learning	23.010	53.597***	.000	.244	Posttest > Pretest
Mastery experience	7.701	12.584***	.001	.142	Posttest > Pretest
Foundational ability	8.107	14.785***	.000	.163	Posttest > Pretest
Advanced ability	5.651	7.538**	.008	.090	Posttest > Pretest
Mindset					
Growth mindset	2.183	5.248*	.023	.031	Posttest > Pretest
Fixed mindset	2.415	4.625*	.033	.027	Posttest < Pretest

Notes: $^*p < .05$. $^{**}p < .05$. $^{***}p < .05$.

4.3 The process model of motivation, growth mindset, mastery experience, and self-efficacy

We employed SPSS PROCESS to examine the potential mechanisms underlying the enhancement of programming self-efficacy during the training. Because we assumed that a fixed mindset would be negatively correlated with the other variables, we did not include it in the model. The correlation between the two types of achievement goals was .575 ($p < .001$). The mean scores (and standard deviations) for the overall motivation scale, mastery goal, and performance goal were 3.58 (0.80), 3.79 (0.83), and 3.31 (1.00), respectively. To assess whether mastery goal and performance goal motivations exerted distinct influences within the PROCESS framework, we conducted separate analyses for each motivational type. In the following model descriptions, The proposed mediation models are presented in Figures 4 and 5.

Model 1 (see Figure 4) examined the influence of mastery goal motivation on growth mindset, mastery experience, and programming self-efficacy. Three significant indirect pathways were identified: (1) Mastery goal → Growth mindset → Self-efficacy: Mastery goal positively predicted growth mindset (β = .696, $p < .001$), which in turn predicted self-efficacy (β = .438, $p < .001$); total indirect effect = .305. (2) Mastery goal → Mastery experience → Self-efficacy: Mastery goal directly predicted mastery experience (β = .388, $p < .001$), which subsequently predicted self-efficacy (β = .456, $p < .001$); total indirect effect = .177. (3) Mastery goal → Growth mindset → Mastery experience → Self-efficacy: Mastery goal predicted growth mindset (β = .696, $p < .001$), which predicted mastery experience (β = .485, $p < .001$), which then predicted self-efficacy (β = .456, $p < .001$); total indirect effect = .154.

Again, the direct effect of Mastery goal on self-efficacy was not significant (β = .051, $p = .361$). Overall, the total indirect effect was significant (β = .635), and the total effect of Mastery goal on self-efficacy was strong (β = .687, $p < .001$), explaining 47.1% of the variance ($R^2 = .471$).

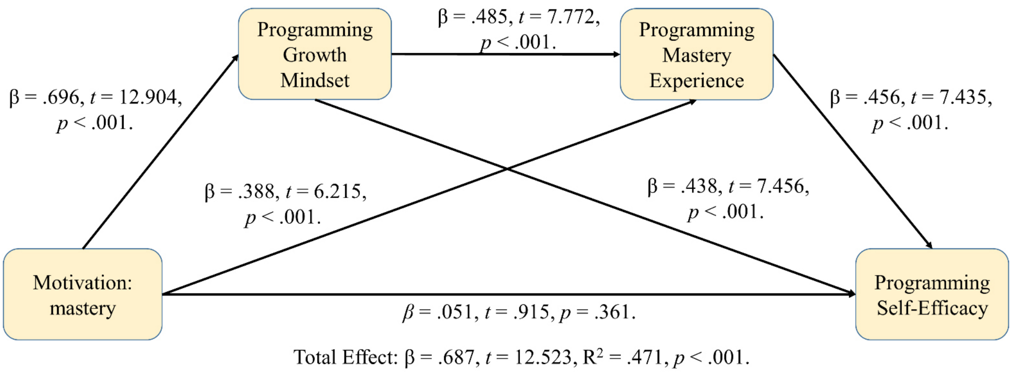

Fig. 4. PROCESS model of mastery goal, growth mindset, mastery experience, and programming self-efficacy

Model 2 (see Figure 5 examined the influence of performance goal motivation on growth mindset, mastery experience, and programming self-efficacy. Three significant indirect pathways emerged: (1) Performance goal → Growth mindset → Self-efficacy: Performance goal positively predicted growth mindset (β = .359, $p < .001$), which in turn predicted self-efficacy (β = .444, $p < .001$); total indirect effect = .159. (2) Performance goal → Mastery experience → Self-efficacy: Performance goal did not significantly predict mastery experience (β = .097, $p = .066$), although mastery experience predicted self-efficacy (β = .468, $p < .001$); total indirect effect = .045. (3) Performance goal → Growth mindset → Mastery experience → Self-efficacy: Performance goal predicted growth mindset (β = .359, $p < .001$), which predicted mastery experience (β = .720, $p < .001$), which then predicted self-efficacy (β = .468, $p < .001$); total indirect effect = .121.

Notably, the direct effect of performance goal on self-efficacy was not significant (β = .056, $p = .361$). Overall, the total indirect effect was significant (β = .323), while the total effect of motivation on self-efficacy was weaker than in the other two models (β = .382, $p < .001$), explaining 14.6% of the variance ($R^2 = .146$).

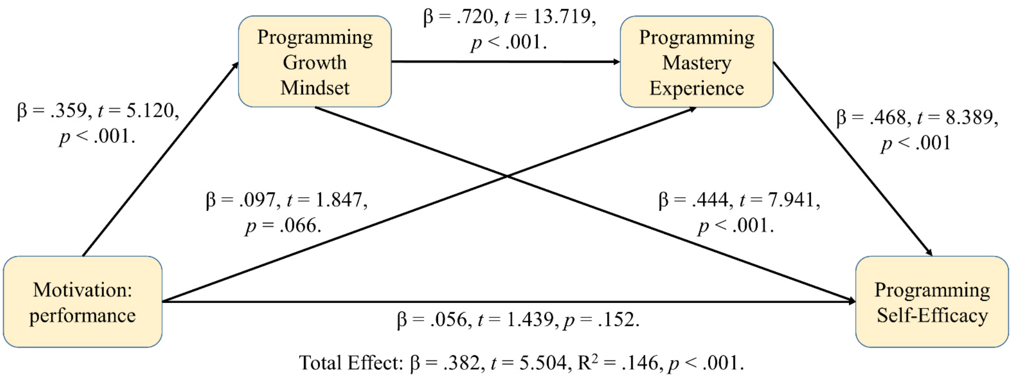

Fig. 5. PROCESS model of performance goal, growth mindset, mastery experience, and programming self-efficacy

5 DISCUSSION

5.1 Inventory development

Two inventories were developed in this study—the Inventory of Programming Achievement Goals and the IPM. Both demonstrated good reliability and validity.

Achievement goals are critical for fostering the ability to tackle challenging tasks [12], [11]. Accordingly, we created the Inventory of Programming Achievement Goals, initially developing items based onElliot and McGregor's 2 × 2 model [14], which distinguishes mastery-approach, mastery-avoidance, performance-approach, and performance-avoidance goals. However, following reliability and validity testing, the items converged into two factors—Mastery goal and Performance goal—consistent with the earlier two-factor conceptualization of goal orientation [8].

Additionally, drawing on Yeh et al.'s 2 × 2 mindset framework [26], we developed items to measure four types of programming mindsets: Growth–internal, Growth–external, Fixed–internal, and Fixed–external. After reliability and validity analysis, these items also converged into two factors—Growth mindset and Fixed mindset—supporting the two-construct theory of mindset [19]. These two mindsets demonstrated a moderate negative correlation (r = –.649), indicating that they represent distinct, independent dimensions that should be assessed separately.

5.2 Instructional effects on programming growth mindset, mastery experience, and self-efficacy

In an exploratory manner, we examined the training effects on mastery experience and self-efficacy in programming, as well as fixed and growth mindsets toward programming. The effects of achievement goal motivation were not assessed because the study did not include any manipulation of this construct and only the pretest data were collected. To address the need for cultivating Python programming self-confidence among non-computer science majors, we designed and implemented an 18-week integrative training course. The results suggest that the instructional strategies, such as AI-assisted CPBL, scaffolding, hands-on activities, the use of an online learning platform, multimedia presentations, and reflective practices, effectively enhanced participants' growth mindset, mastery experience, and self-efficacy in Python programming, while reducing their fixed mindset.

The course integrated diverse technologies to strengthen conceptual understanding, foster collaboration and discussion, and develop problem-solving skills and confidence. A key component was the use of *Python Maker Activities* within a CPBL framework, which provided hands-on, real-world problem-solving experiences. These activities, combined with continuous feedback, created repeated opportunities for success, thereby building students' sense of competence and self-efficacy. This aligns with findings that CPBL supports mastery experiences and self-efficacy development [40].

The results also support prior evidence that AI-supported tools and adaptive learning platforms enhance personalization and engagement, enabling learners to improve their programming performance [43] and programming self-efficacy [4], as well as to complete creative programming projects [44]. Furthermore, the findings are consistent with research showing that project-based learning strengthens self-efficacy [40] and programming performance [37]. Overall, these results underscore the value of thoughtfully designed, AI-supported CPBL that actively engages learners, promotes reflective practice, and scaffolds both self-learning and psychological growth.

5.3 Relationship of motivation, growth mindset, mastery experience, and programming self-efficacy in collaborative and AI-supported learning

The PROCESS results showed that both forms of achievement goals, particularly mastery goals, demonstrated strong explanatory power for self-efficacy through growth mindset and mastery experience in Python programming. The findings revealed that achievement goal motivation influenced self-efficacy through one direct pathway and three indirect pathways. Although none of the "achievement goal → self-efficacy" direct effects were significant, all total indirect effects were significant.

Regarding the first indirect path (achievement goal → growth mindset → self-efficacy), mastery goal motivation yielded the largest total effect (.305), compared to performance goal (.159), with a particularly strong effect on growth mindset (β = .696, $p < .001$). For the second indirect path (achievement goal → mastery experience → self-efficacy), the mastery goal also had a larger total effect (.177) than the performance goal (.045). Notably, although the total indirect effect was significant, the influence of performance goal on mastery experience was not significant. Finally, regarding the third indirect path (achievement goal → growth mindset → mastery experience → self-efficacy), mastery goal also had a larger total effect (.154) than performance goal (121). For the *overall total effects* (direct plus indirect), mastery goal motivation again produced the strongest effect (β = .687, R^2 = .471), substantially exceeding that of performance goal motivation (β = .384, R^2 = .146).

Overall, growth mindset and mastery experience consistently emerged as critical mediators between achievement goal motivation and programming self-efficacy, suggesting the view that self-efficacy is shaped by the interplay of motivation, mindset, and mastery experience. Moreover, the findings support that motivation plays a foundational role in shaping mindset [34].

Among the motivational types, mastery goal motivation produced the most coherent and effective pathway for enhancing programming self-efficacy. Within the CPBL framework of this training—characterized by reflective thinking, peer collaboration, and AI-supported activities—students with mastery goals appeared more willing to embrace challenges, thereby strengthening their growth mindset. This finding aligns with research showing that learners with a mastery goal orientation are more likely to adopt a growth mindset [35] and that the growth mindset serves as a key mediator between motivation and mastery experience [24]. It also supports the view that the quality of achievement goals plays a crucial role in shaping learners' cognition, emotions, behavior, and ultimately, achievement outcomes [11].

In sum, the findings reinforce that effective programming instruction should address not only technical skill development but also the psychological readiness to persevere and succeed. This study provides an empirical profile of how non–computer science students' achievement goal orientations interact with technology- and AI-supported training activities to build programming self-efficacy.

6 CONCLUSIONS

Collaborative problem-based learning with AI-supported training holds strong potential for enhancing programming performance and self-confidence. In this study, these strategies and AI toolkits were integrated into an 18-week training course. Although the instructional effects required further examination, the findings suggest that AI-supported CPBL fosters a supportive learning environment with rich hands-on activities, contributing to the development of a growth mindset, mastery experience, and self-efficacy in programming. Importantly, growth mindset and mastery experience emerged as key mediators linking achievement goal motivation to self-efficacy.

In conclusion, this study contributes to programming education in the following aspects: (1) Developing an AI-supported CPBL framework: It demonstrates how AI toolkits, CPBL, technological support, and scaffolding can be integrated to enhance Python learning and its interdisciplinary applications. (2) Revealing psychological mechanisms in programming learning: Growth mindset and mastery experience emerged as key mediators linking achievement goal motivation to self-efficacy, offering insights into how programming self-confidence is built among non–computer science majors. (3) Clarifying the role of achievement goal types: It highlights that mastery goals exert a substantially stronger influence on programming self-efficacy than performance goals, highlighting the need to distinguish goal types in AI-supported CPBL design. (4) Conceptualizing and measuring programming mindsets: This study is among the first to conceptualize, distinguish, and validate the dual dimensions of programming mindsets, providing robust instruments for future research and instructional design in programming education.

7 LIMITATIONS AND SUGGESTIONS

Although the training effects were examined, the primary focus of this study was to investigate how achievement goal motivation shapes programming self-efficacy through the mediation of growth mindset and mastery experience in an AI-supported collaborative learning environment. Consequently, a control group was not included. Nevertheless, the substantial instructional effects observed provide partial evidence of the effectiveness of our instructional framework, which may serve as a model for teaching students who are apprehensive about programming. Future research could incorporate a control group and compare outcomes across different AI tools (e.g., MindScratch) to evaluate relative learning benefits.

As the adoption of generative AI becomes increasingly widespread, it is no longer sufficient to simply teach students how to use AI for programming tasks. More critically, educators must scaffold learners in applying AI tools to solve authentic, real-world problems through programming. Incorporating AI tools into CPBL warrants further empirical research to establish a stronger theoretical foundation. In doing so, personal traits such as motivation and mindset should be carefully considered, along with other potentially important psychological variables.

Finally, this study did not assess peer interaction during or outside the training course. Given its central role in CPBL, future research should examine the nature and impact of peer collaboration, potentially leveraging generative AI to facilitate or analyze these interactions.

7.1 Statements and declarations

Financial interests: The authors declare that they have no financial interests.

7.2 Declaration of generative AI and AI-assisted technologies

During the preparation of this work, the authors used ChatGPT in order to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

8 REFERENCES

[1] F. H. Tsai, "Using a physical computing project to prepare preservice primary teachers for teaching programming," *SAGE Open,* vol. 13, no. 4, 2023. https://doi.org/10.1177/21582440231205409

[2] U. Tellhed, F. Bjorklund, and K. K. Strand, "Sure I can code (but do I want to?). Why boys' and girls' programming beliefs differ and the effects of mandatory programming education," *Computers in Human Behavior*, vol. 135, pp. 1–11, 2022. https://doi.org/10.1016/j.chb.2022.107370

[3] M. Öztürk, "The effect of self-regulated programming learning on undergraduate students' academic performance and motivation," *Interactive Technology and Smart Education*, vol. 19, pp. 319–337, 2022. https://doi.org/10.1108/ITSE-04-2021-0074

[4] R. Yilmaz and F. G. K. Yilmaz, "The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy, and motivation," *Computers and Education: Artificial Intelligence*, vol. 4, p. 100147, 2023. https://doi.org/10.1016/j.caeai.2023.100147

[5] J. Liu, Q. Li, X. Sun, Z. Zhu, and Y. Xu, "Factors influencing programming self-efficacy: An empirical study in the context of Mainland China," *Asia Pacific Journal of Education,* vol. 43, pp. 835–849, 2021. https://doi.org/10.1080/02188791.2021.1985430

[6] Kanika, S. Chakraverty, and P. Chakraborty, "Tools and techniques for teaching computer programming: A review," *Journal of Educational Technology Systems,* vol. 49, no. 2, pp. 170–198, 2022. https://doi.org/10.1177/0047239520926971

[7] R. Rinas, M. Dresel, and M. Daumiller, "Faculty subjective well-being: An achievement goal approach," *International Journal of Educational Research,* vol. 115, p. 101942, 2022. https://doi.org/10.1016/j.ijer.2022.101942

[8] C. S. Dweck, "What having a 'growth mindset' actually means," *Harvard Business Review,* vol. 13, pp. 213–226, 2022.

[9] N. J. Beatson, S. T. M. Sithole, P. de Lange, B. O'Connell, and J. K. Smith, "Sources of self-efficacy beliefs in learning accounting: Does gender matter?" *Journal of International Education in Business*, vol. 18, no. 2, pp. 196–217, 2025. https://doi.org/10.1108/JIEB-02-2024-0014

[10] N. Hampel, K. Sassenberg, A. Scholl, and L. Ditrich, "Enactive mastery experience improves attitudes towards digital technology via self-efficacy – a pre-registered quasi-experiment," *Behaviour & Information Technology,* vol. 43, no. 2, pp. 298–311, 2023. https://doi.org/10.1080/0144929X.2022.2162436

[11] S. Schweder, G. Hagenauer, L. Grahl, and D. Raufelder, "Achievement goal profiles and profile transitions in teacher- and self-directed learning and the association with self-efficacy and interest," *Motivation and Emotio*n, vol. 49, no. 5, pp. 593–623, 2025. https://doi.org/10.1007/s11031-025-10142-0

[12] M. Daumiller, "Achievement goals: The past, the present, and possible future of achievement goal research in the context of learning and teaching," in *Motivation and Emotion in Learning and Teaching Across Educational Contexts*, G. Hagenauer, R. Lazarides, and H. Järvenoja, Eds., Routledge, 2024, pp. 35–53. https://doi.org/10.31234/osf.io/xn2v5

[13] P. R. Pintrich and D. Schunk, *Motivation in Education: Theory, Research, and Applications*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.

[14] A. J. Elliot and H. A. McGregor, "A 2 × 2 achievement goal framework," *Journal of Personality and Social Psychology*, vol. 80, no. 3, pp. 501–519, 2001. https://doi.org/10.1037/0022-3514.80.3.501

[15] Y. K. Lee and E. Seo, "Trajectories of implicit theories and their relations to scholastic aptitude: A mediational role of achievement goals," *Contemporary Educational Psychology,* vol. 59, p. 101800, 2019. https://doi.org/10.1016/j.cedpsych.2019.101800
[16] A. Camacho, R. A. Alves, J. R. Daniel, F. De Smedt, and H. Van Keer, "Structural relations among implicit theories, achievement goals, and performance in writing," *Learning and Individual Differences,* vol. 100, p. 102223, 2002. https://doi.org/10.1016/j.lindif.2022.102223
[17] K. Wang, B. Duan, X. Li, and Y. Xue, "A profile analysis of computer-supported collaborative learning motivation in Chinese higher education: Integrating achievement goal and expectancy-value perspectives," *Interactive Learning Environments*, vol. 34, no. 3, pp. 1355–1377, 2025. https://doi.org/10.1080/10494820.2025.2521342
[18] Y. Chen, J. Li, H. Chui, and R. B. King, "Peer cooperation and competition are both positively linked with mastery-approach goals: An achievement goal perspective," *British Journal of Educational Psychology,* vol. 95, no. 4, pp. 941–961, 2025. https://doi.org/10.1111/bjep.12784
[19] K. Bernecker and V. Job, "Mindset theory," in *Social Psychology in Action: Evidence-Based Interventions from Theory to Practice*, K. Sassenberg and M. L. W. Vliek, Eds., Springer, Charm, 2019, pp. 179–191. https://doi.org/10.1007/978-3-030-13788-5_12
[20] A. Kapasi and J. Pei, "Mindset theory and school psychology," *Canadian Journal of School Psychology,* vol. 37, no. 1, pp. 57–74, 2021. https://doi.org/10.1177/08295735211053961
[21] C. J. Fong *et al.*, "Do socializers' mindset beliefs matter for student mindset and achievement? A meta-analysis," *Learning and Individual Differences,* vol. 121, p. 102709, 2025. https://doi.org/10.1016/j.lindif.2025.102709
[22] J. B. Sarrasin, L. Nenciovici, L. M. B. Foisy, G. Allaire-Duquette, M. Riopel, and S. Masson, "Effects of teaching the concept of neuroplasticity to induce a growth mindset on motivation, achievement, and brain activity: A meta-analysis," *Trends in Neuroscience and Education,* vol. 12, pp. 22–31, 2018. https://doi.org/10.1016/j.tine.2018.07.003
[23] D. Platte, K. M. Xu, and R. H. M. de Groot, "The effect of fostering a growth mindset in primary school children: Does intervention approach matter?" *Education Sciences,* vol. 15, no. 3, p. 327, 2025. https://doi.org/10.3390/educsci15030327
[24] Y. S. Ting and Y. Yeh, "Growth-mindset intervention effects and the relationship of mindset, hope belief, and self-efficacy during creativity game-based learning," *Interactive Learning Environments*, vol. 32, no. 7, pp. 3146–3162, 2024. https://doi.org/10.1080/10494820.2023.2170418
[25] A. Chouvalova *et al.*, "Undergraduates' reactions to errors mediate the association between growth mindset and study strategies," *International Journal of STEM Education,* vol. 11, p. 26, 2024. https://doi.org/10.1186/s40594-024-00485-4
[26] Y. Yeh, Y. S. Ting, and J. L. Chiang, "Influence of growth mindset, fixed mindset, grit, and self-determination on self-efficacy in game-based creativity learning," *Educational Technology & Society,* vol. 26, no. 1, pp. 62–78, 2023. https://doi.org/10.30191/ETS.202301_26(1).0005
[27] S. Boguslawski, R. Deer, and M. G. Dawson, "Programming education and learner motivation in the age of generative AI: Student and educator perspectives," *Information and Learning Sciences,* vol. 126, nos. 1–2, pp. 91–109, 2025. https://doi.org/10.1108/ILS-10-2023-0163
[28] H. Y. Durak, F. G. K. Yilmaz, and R. Yilmaz, "Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities," *Contemporary Educational Technology*, vol. 10, no. 2, pp. 173–197, 2019. https://doi.org/10.30935/cet.554493

[29] Y. T. Kuo and Y. C. Kuo, "African American students' academic and web programming self-efficacy, learning performance, and perceptions towards computer programming in web design courses," *Education Sciences*, vol. 13, no. 12, pp. 1–17, 2023. https://doi.org/10.3390/educsci13121236

[30] Y. T. Kuo and Y. C. Kuo, "Learning programming: Exploring the relationships of self-efficacy, computational thinking, and learning performance among minority students," *Frontiers in Education*, vol. 10, p. 1623415, 2025. https://doi.org/10.3389/feduc.2025.1623415

[31] H. P. Phan and B. H. Ngu, "Sources of self-efficacy in academic contexts: A longitudinal perspective," *School Psychology Quarterly*, vol. 31, pp. 548–564, 2016. https://doi.org/10.1037/spq0000151

[32] A. L. Kleppang, A. M. Steigen, and H. S. Finbråten, "Explaining variance in self-efficacy among adolescents: The association between mastery experiences, social support, and self-efficacy," *BMC Public Health*, vol. 23, p. 1665, 2023. https://doi.org/10.1186/s12889-023-16603-w

[33] A. Bandura, *Self-Efficacy: The Exercise of Control*. New York, NY: W. H. Freeman, 1997. https://doi.org/10.1037/0033-295X.84.2.191

[34] D. S. Yeager and C. S. Dweck, "Mindsets that promote resilience: When students believe that personal characteristics can be developed," *Educational Psychologist*, vol. 47, no. 4, pp. 302–314, 2012. https://doi.org/10.1080/00461520.2012.722805

[35] Q. Liu and T. Kamioka, "The effects of employees' digital growth mindset and supervisors' coaching behaviour on digital self-efficacy," *Technology in Society*, vol. 81, p. 102875, 2025. https://doi.org/10.1016/j.techsoc.2025.102875

[36] S. Yang, M. Baird, E. O'Bourke, K. Brennan, and B. Schneider, "Decoding debugging instruction: A systematic literature review of debugging interventions," *ACM Transactions on Computing Education*, vol. 24, no. 4, pp. 1–44, 2024. https://doi.org/10.1145/3690652

[37] S. Yeom, N. Herbert, and R. Ryu, "Project-based collaborative learning enhances students' programming performance," *Computer Science Education*, vol. 1, pp. 248–254, 2022. https://doi.org/10.1145/3502718.3524779

[38] A. Zamecnik, V. Kovanović, S. Joksimović, G. Grossmann, D. Ladjal, and A. Pardo, "The perception of task cohesion in collaborative learning teams," *International Journal of Computer-Supported Collaborative Learning*, vol. 19, pp. 369–393, 2024. https://doi.org/10.1007/s11412-024-09424-5

[39] C. Crook, "CSsCL: The performance of collaborative learning," *International Journal of Computer-Supported Collaborative Learning*, vol. 17, pp. 169–183, 2022. https://doi.org/10.1007/s11412-022-09364-y

[40] L. Wijnia, G. Noordzij, L. R. Arends, R. M. J. P. Rikers, and S. M. M. Loyens, "The effects of problem-based, project-based, and case-based learning on students' motivation: A meta-analysis," *Educational Psychology Review*, vol. 36, p. 29, 2024. https://doi.org/10.1007/s10648-024-09864-3

[41] T. T. Wu, N. A. R. M. Sari, A. P. R. Z. Putri, H. R. Chen, and Y. M. Huang, "Fostering undergraduate accounting students' educational attainment through CT-enhanced collaborative project-based learning," *The International Journal of Management Education*, vol. 23, no. 3, p. 101195, 2025. https://doi.org/10.1016/j.ijme.2025.101195

[42] B. Tangney, K. Sullivan, and J. Lawlor, "Online collaborative PBL-The Bridge21 approach," *Computers and Education Open*, vol. 7, p. 100114, 2024. https://doi.org/10.1016/j.caeo.2024.100224

[43] L. Yan, S. Greiff, J. M. Lodge, and D. Gašević, "Distinguishing performance gains from learning when using generative AI," *Nature Reviews Psychology*, vol. 4, pp. 435–436, 2025. https://doi.org/10.1038/s44159-025-00467-5

[44] Y. Chen, S. Xiao, Y. Song, Z. Li, L. Sun, and L. Chen, "MindScratch: A visual programming support tool for classroom learning based on multimodal generative AI," *International Journal of Human–Computer Interaction*, vol. 41, no. 21, pp. 13650–13668, 2025. https://doi.org/10.1080/10447318.2025.2475991

[45] W. S. Tzou, Y. C. Yeh, C. H. Ma, and X. Yu, "Mastery experience bridges computational thinking and self-efficacy in programming among non-computer science majors," in *The 4th Paris Conference on Education*, 2025, pp. 10–14.

9 AUTHORS

Wen-Shyong Tzou is a Professor in the Department of Bioscience and Biotechnology at National Taiwan Ocean University. His research interests include translational genomics, computational protein structure modeling, bioinformatics, and generative AI–driven protein design, integrating nature-inspired algorithms to advance biomolecular engineering (E-mail: wstzou@mail.ntou.edu.tw).

Chia-Hsiang Ma is a PhD. Candidate of the Department of Psychology, National Chengchi University. His research interests focus on bilingual word recognition, with particular emphasis on logographic writing systems. He also works on several natural language processing topics, such as sentence similarity and applications of BERT (E-mail: 110752501@nccu.edu.tw).

Yu-chu Yeh is a distinguished Professor in the Institute of Teacher Education and a Research Fellow at the Research Center for Mind, Brain, and Learning at National Chengchi University. Her interdisciplinary research focuses on the intersection of generative AI-enhanced instruction, game-based learning, and neuroeducation. With a foundation in educational, cognitive, and positive psychology, she specializes in fostering creativity and critical thinking within modern learning environments (E-mail: ycyeh@nccu.edu.tw; ycyeh@mail2.nccu.tw).

Melissa T. A. Simarmata was a postdoctoral researcher at National Chengchi University, where she contributed to projects related to AI ethics, motivation, and digital learning systems. Her research interests include personal innovativeness, knowledge management, and educational technology (E-mail: melissasimarmata@gmail.com).