Krittawaya Thongkoo
PhD in Science and Technology Education, Assistant Professor
College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand
ORCID ID 0000-0003-2167-658X
krittawaya.t@cmu.ac.th

Kannika Daungcharone
PhD in Science and Technology Education, Assistant Professor
College of Arts, Media and Technology, Chiang Mai University, Chiang Mai, Thailand
ORCID ID 0000-0002-4108-2087
kannika.d@cmu.ac.th

DESIGN AND IMPLEMENTATION OF A UBIQUITOUS CODE ANNOTATION SYSTEM FOR INQUIRY-BASED COLLABORATIVE LEARNING OF PROGRAMMING

Abstract. The growing complexity of computer programming education has intensified the need for digital learning environments that support not only technical proficiency but also collaborative inquiry, reflective practice, and deeper conceptual development. In response to this demand, the present study designed, implemented, and empirically evaluated a ubiquitous inquiry-based learning environment integrated with a collaborative digital coding annotation system, known as UILA-CDCAS. The platform was developed to embed inquiry processes directly into authentic programming tasks. It enables learners to annotate code segments, exchange targeted peer feedback, pose context-specific questions, and engage in real-time collaborative problem-solving across both classroom and distributed learning contexts. This study aims to evaluate the system's effectiveness in improving programming learning outcomes and learner engagement. A quasi-experimental study was conducted with 50 first-year university students. Analysis of covariance was used to control for differences in prior knowledge. The findings demonstrate that students in the experimental condition achieved significantly higher learning gains in programming proficiency and conceptual understanding. In addition, collaborative interaction analytics and learner perception measures revealed heightened engagement and sustained motivation. They also indicated increased peer-supported knowledge construction and more substantive reflective learning practices associated with code annotation activities. Collectively, these results highlight the pedagogical and technological value of integrating collaborative coding annotation into ubiquitous inquiry-based learning designs. They also provide empirical and design-oriented insights for the design of digital platforms that support programming education and technology-enhanced learning.

Keywords: ubiquitous inquiry-based learning; collaborative coding annotation; programming education environments; technology-enhanced collaborative learning; inquiry-driven programming instruction

1. INTRODUCTION

The rapid advancement of computing technologies has transformed not only the content of computer programming education but also the technological infrastructures through which programming knowledge is constructed, practiced, and shared. As contemporary software development increasingly involves collaborative workflows, distributed problem-solving, and iterative code refinement, educational environments are expected to mirror these professional practices through digitally mediated learning systems [1], [2]. However, many programming classrooms continue to rely on conventional instructional models and isolated coding tools that do not sufficiently support collaborative inquiry, real-time feedback, and socially situated knowledge construction. This limitation has resulted in a persistent gap between theoretical instruction and authentic programming practice, particularly in fostering conceptual understanding and higher-order problem-solving competencies [3]. Although the integration of advanced digital learning technologies has demonstrated potential in enabling interactive,

© Krittawaya Thongkoo, Kannika Daungcharone, 2026.

learner-centered programming environments [4], existing systems often fail to support structured inquiry processes and meaningful peer collaboration simultaneously. Among emerging pedagogical approaches, inquiry-based learning (IBL) has gained increasing attention in computer science education due to its emphasis on active knowledge construction through questioning, investigation, and problem exploration [5]. Grounded in constructivist and social constructivist traditions, IBL repositions learners as active investigators who construct understanding through iterative engagement with authentic coding challenges. Nevertheless, despite its theoretical promise, the implementation of inquiry-based programming instruction remains constrained by the limited availability of digital systems capable of integrating inquiry processes with structured collaborative interaction [6].

To address this gap, this study designed and implemented a ubiquitous inquiry-based learning environment integrated with a collaborative digital coding annotation system, referred to as UILA-CDCAS. The system embeds inquiry processes directly into authentic programming tasks by enabling learners to annotate code segments, exchange targeted peer feedback, and engage in real-time collaborative problem-solving across distributed learning contexts [7]. The platform also supports both synchronous and asynchronous knowledge co-construction, extending programming interaction beyond conventional classroom boundaries. This study examines the effects of UILA-CDCAS on first-year undergraduate students' programming proficiency, conceptual understanding, collaborative learning behaviors, motivation, and engagement. It evaluates its effectiveness by comparing it with a ubiquitous inquiry-based learning approach supported by general digital coding tools (UILA-GDCS). Beyond evaluating system effectiveness, this study contributes to the field of computer science education and technology-enhanced learning in three key ways. First, it provides empirical evidence on the impact of integrating collaborative coding annotation within an inquiry-based learning framework. Second, it offers a design-oriented perspective on how digital annotation environments can support collaborative inquiry and conceptual understanding in programming education. Third, it extends existing research by demonstrating how inquiry-driven and collaboration-supported learning processes can be operationalized within a unified ubiquitous learning system.

1.1 Research Goal

The primary aim of this study is to evaluate the effectiveness of a ubiquitous inquiry-based collaborative coding annotation system (UILA-CDCAS) in improving students' programming learning outcomes and engagement. Specifically, the study seeks to examine its impact on programming proficiency, conceptual understanding, motivation, and collaborative learning behaviors in comparison with a traditional inquiry-based learning approach supported by general digital coding tools.

Following this introduction, the study addresses the following research questions:

- To what extent does the UILA-CDCAS approach improve programming proficiency and conceptual understanding compared with a traditional inquiry-based learning approach supported by general digital coding tools (UILA-GDCS)?
- How does the UILA-CDCAS approach influence learners' motivation, engagement, and collaborative practices compared to traditional inquiry-based methods?

2. THEORETICAL BACKGROUND

The integration of pedagogical innovation and technological affordances has become central to advancing computer programming education within digitally mediated learning environments. From a theoretical perspective, the design of technology-enhanced programming

systems is informed by intersecting conceptual traditions that explain how knowledge is constructed, shared, and applied through collaborative digital interaction. This section synthesizes theoretical foundations across four interrelated domains—Inquiry-Based Learning (IBL), collaborative learning supported by digital tools, collaborative coding annotation systems within ubiquitous learning environments, and the systemic challenges and opportunities shaping their implementation. Each domain represents a distinct yet complementary lens through which technology-enhanced programming education can be conceptualized. Collectively, they converge on the development of robust, student-centered digital learning ecosystems that operationalize inquiry, collaboration, and context-embedded problem-solving.

2.1 Inquiry-Based Learning (IBL) for Promoting Programming Skills

Inquiry-Based Learning (IBL), grounded in constructivist and social constructivist learning theories, reframes the role of the learner from a passive recipient to an active investigator [8]. In programming education, this approach prompts students to engage in cycles of questioning, hypothesis generation, experimentation, and reflection. Empirical studies have shown that IBL fosters deeper conceptual understanding and transferable problem-solving skills by allowing learners to confront authentic learning challenges and explore multiple solution paths [9]. More recent research confirms that embedding IBL in coding curricula enhances computational thinking and adaptability to emerging technologies, particularly in AI-assisted programming contexts [10]. In this study, IBL is operationalized through structured inquiry cycles embedded within programming tasks, including problem exploration, hypothesis generation, collaborative code analysis, and iterative refinement supported by peer interaction and instructor facilitation.

However, the successful implementation of IBL in programming contexts requires a shift in both curriculum design and instructional practice. Core strategies include integrating open-ended coding problems, adaptive scaffolding to accommodate diverse levels of learner readiness, and deliberate facilitation of peer discourse [11]. Kopeyev, Akrimzhanov, and Bidaibekov [12] highlight that engagement in cloud-based, cognitively activating learning environments can strengthen learners' computational thinking and algorithmic problem-solving capacities, particularly when supported by self-regulated learning and sustained cognitive engagement. Recent case studies also reveal that hybrid IBL models, combining self-directed inquiry with AI-driven feedback, can mitigate common learner frustrations in programming courses [13]. Nonetheless, students accustomed to directive, teacher-centered instruction may experience cognitive and motivational barriers during the transition to IBL environments [14]. Thus, structured onboarding, continuous formative feedback, and explicit modeling of inquiry processes are essential to support this pedagogical shift.

Significantly, while IBL excels at cultivating problem-solving autonomy, its benefits are magnified when paired with collaborative mechanisms that allow learners to negotiate meaning and share cognitive load, as explored in the next section.

2.2 Collaborative Learning and Digital Tools in Computer Education

Collaboration in programming education mirrors professional software development practices, where complex problems are often addressed through distributed expertise and coordinated effort. Prior research has shown that structured collaborative dialogue in pair programming can significantly enhance students' programming self-efficacy and coding performance [15]. In this context, digital tools designed for collaborative learning – such as real-time code editors, version control platforms, and annotation systems – enable students to construct shared solutions and engage in reciprocal feedback processes, including

communication, coordination, and collaborative problem-solving. These practices are consistent with emerging perspectives in software engineering education [16]. Recent large-scale classroom implementations have shown that integrating cloud-based collaborative environments increases both peer-to-peer help-seeking behavior and sustained engagement across the semester [17], [18].

The literature underscores that digital collaboration tools amplify the benefits of active learning by extending the scope and immediacy of peer interaction. Prior research has shown that collaborative annotation tools can promote active engagement and learner-centered participation in technology-enhanced environments [19]. Real-time annotation systems, for instance, allow students to highlight specific code segments, pose context-sensitive questions, and propose modifications. These interactions support formative assessment processes, complementing automated grading and feedback mechanisms that have been shown to enhance programming learning through timely and structured feedback [20]. Recent advances in collaborative platforms have increasingly incorporated AI-driven techniques – such as automated program repair and code generation – to support code review and enhance the quality of peer feedback without diminishing student agency [21]. In parallel, cloud-based collaborative tools have been shown to reduce geographical and temporal constraints, enabling more flexible participation and fostering digital teamwork and engagement in programming learning contexts [22].

Notably, when collaborative tools are integrated into an inquiry-driven framework, they transform individual problem-solving into a collective knowledge-building endeavor. This convergence is exemplified in the use of collaborative coding annotation systems, discussed in the following section.

2.3 Collaborative Coding Annotation Systems in Ubiquitous Learning

Ubiquitous learning environments leverage networked technologies to make educational resources and interactions accessible "anytime, anywhere" [23]. In this study, ubiquitous learning is operationalized through cloud-based access to learning resources, support for both synchronous and asynchronous interaction, and the continuity of collaborative programming activities across classroom and distributed learning contexts. Within these contexts, collaborative coding annotation systems (CCAS) serve as both cognitive scaffolding and social learning spaces. By enabling synchronous and asynchronous code annotation, CCAS facilitates joint problem diagnosis, contextualized feedback, and iterative solution refinement [24]. Empirical evidence from web-based inquiry learning environments further demonstrates that annotation-supported systems can significantly enhance learners' problem-solving performance and information literacy by integrating inquiry processes with collaborative digital interaction [25].

From a pedagogical standpoint, CCAS aligns with Vygotsky's [26] concept of the Zone of Proximal Development (ZPD), wherein learners advance their competencies through guided interaction with more capable peers. Empirical research further indicates that collaborative annotation in programming contexts can enhance code comprehension, facilitate peer feedback processes, and support collaborative problem-solving practices [27]. Recent longitudinal studies further indicate that students who engage regularly in code annotation activities demonstrate sustained improvements in software design patterns and refactoring practices [28]. Furthermore, the persistent, documented nature of annotations supports reflective learning, allowing students to revisit prior discussions and rationales.

However, to fully realize their potential, CCAS must be integrated within instructional designs that actively promote inquiry and collaboration – linking the investigative depth of IBL with the social affordances of collaborative tools.

2.4 Challenges and Opportunities in Implementing Digital Annotation Systems

Despite the pedagogical promise of digital annotation systems, their adoption is mediated by systemic, technical, and pedagogical factors. Teacher preparedness is a recurring challenge; without targeted training, educators may underutilize advanced collaborative features or fail to align them with learning objectives. Accessibility issues, including disparities in device ownership and internet connectivity, threaten to exacerbate the digital divide [29]. Recent policy analyses suggest that institutional support frameworks and professional development programs are decisive in scaling such tools effectively [30]. Moreover, usability considerations, such as intuitive interface design, integration with existing learning management systems, and compliance with data privacy regulations, are critical to sustained adoption.

On the other hand, the opportunities are substantial. Digital annotation systems enable continuous formative assessment, support student agency in the learning process, and create rich datasets for learning analytics. New developments in AI-driven learning analytics and annotation-related technologies enable educators to identify common misconceptions and tailor instructional interventions more effectively [31]. When strategically implemented within an IBL framework and coupled with collaborative practices, these systems can transform programming education from a solitary, syntax-focused activity into a socially mediated, inquiry-rich experience.

These theoretical perspectives can be integrated into a unified conceptual model. This model combines inquiry-based learning, collaborative learning, and ubiquitous learning environments to support programming education. Addressing implementation challenges – particularly those related to teacher readiness, equitable access, and technological integration – remains essential to realizing these theoretical affordances in practice. Within this conceptual framing, the UILA-CDCAS represents a system-level instantiation of these intersecting theories, designed to translate pedagogical principles into a scalable digital learning environment that enhances programming proficiency, conceptual understanding, and collaborative engagement in contemporary computer science education. Despite these advances, existing studies have largely examined inquiry-based learning, collaborative tools, and annotation systems in isolation. Limited research has systematically integrated these elements within a unified framework to support programming education.

3. RESEARCH METHODS

Figure 1 presents the quasi-experimental framework developed to examine the pedagogical and technological effects of UILA-CDCAS on programming learning. The study employed a controlled-comparative design with an experimental and a control group during an eight-week instructional intervention. To strengthen internal validity, both groups were taught by the same instructor and followed the same curriculum, learning objectives, and inquiry-based facilitation strategies. The intervention was operationalized through two contrasting digital learning environments that enacted the same inquiry-based structure but differed in the availability of collaborative annotation affordances. The experimental group used UILA-CDCAS, which integrates real-time collaborative code annotation, contextualized peer feedback, synchronous discussion, and instructor scaffolding within shared code artifacts. This environment enables learners to externalize reasoning and iteratively refine solutions during inquiry cycles. In contrast, the control group used general digital coding tools (UILA-GDCS) that supported standard code editing and submission but did not provide embedded annotation or structured peer interaction within code segments, resulting in inquiry processes being enacted primarily through individual work supplemented by instructor-mediated discussion. Data collection followed a transparent, stepwise sequence aligned with the intervention. A baseline

pre-test was administered prior to Week 1, followed by continuous capture of platform interaction logs throughout the eight weeks. A post-test was conducted at the end of Week 8, along with pre- and post-surveys measuring engagement and motivation. Post-intervention semi-structured interviews were also conducted to capture learners' perceptions.

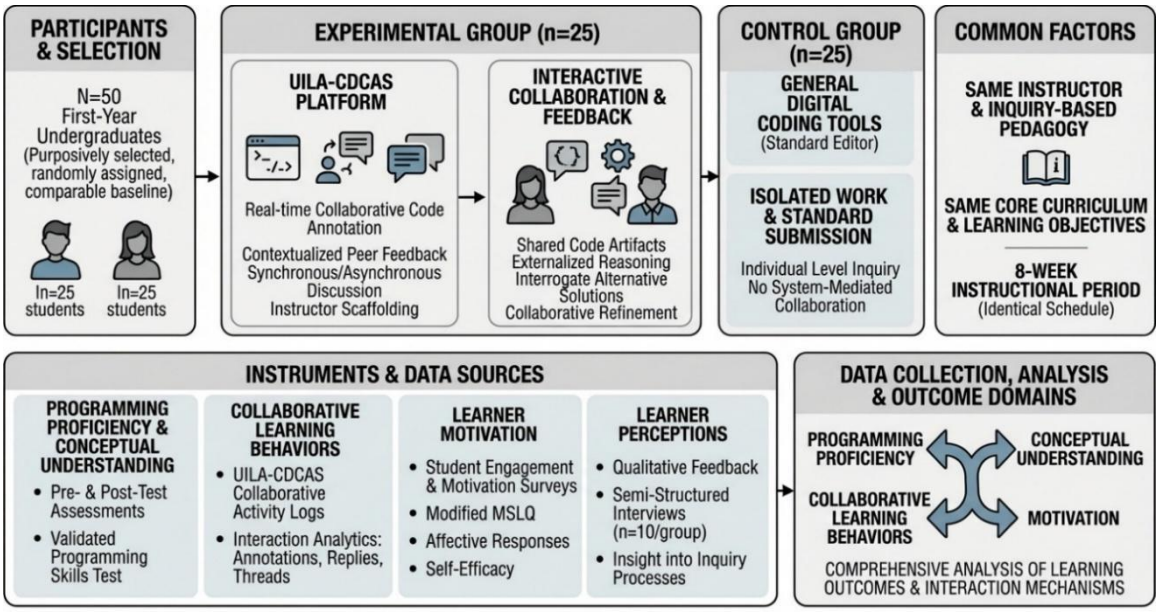

Figure 1. Research framework of the study

3.1 Participants

The study involved 50 first-year undergraduate students enrolled in an introductory computer programming course at a public university. Participants were drawn from two intact classroom sections to preserve the ecological validity of the instructional setting. One class was assigned as the experimental group (n = 25), and the other as the control group (n = 25). To ensure baseline comparability, participants' foundational programming knowledge was assessed using a diagnostic pre-test administered prior to the intervention. The results were examined to confirm that no substantial differences existed between groups before the instructional treatment. As illustrated in Figure 1, the experimental group participated in inquiry-based learning activities through the UILA-CDCAS platform, while the control group followed the same inquiry-based instructional structure using general digital coding tools (UILA-GDCS) without collaborative annotation functionality. Both groups were taught by the same instructor, followed the same instructional schedule, and received equivalent learning content. To control for potential confounding variables, demographic and background information (e.g., age, gender, and prior coding experience) were collected and considered during analysis where appropriate. These controls ensured that differences in learning outcomes could be more confidently attributed to the intervention rather than external factors.

Prior to data collection, the researchers explained the study procedures to all participants. Informed consent was obtained, and all data were handled in accordance with confidentiality and anonymity standards. Participation was voluntary, and students were assured that their academic evaluation would not be affected.

3.2 Research Design and Instruments

A quasi-experimental pre-test/post-test design was employed to compare UILA-CDCAS and UILA-GDCS across cognitive, behavioral, and affective outcome domains. Both groups

followed the same instructional structure, learning objectives, and duration, with the primary difference being the integration of collaborative annotation functionality in the experimental condition. Programming proficiency and conceptual understanding were measured using pre- and post-intervention assessments. These assessments captured both procedural coding competence and higher-order reasoning (e.g., program logic explanation, debugging reasoning, and solution quality). Content validity was established through expert review to ensure alignment with course learning outcomes. To document collaborative learning processes, system-generated UILA-CDCAS interaction logs were collected for the experimental group. These logs included time-stamped records of annotation activity, peer-response exchanges, and refinement-oriented discussion threads. The data were operationalized into quantitative indicators such as annotation frequency, peer-reply density, and refinement patterns. Learner engagement and motivational orientations were assessed using a contextualized adaptation of the Motivated Strategies for Learning Questionnaire (MSLQ) for programming and technology-mediated collaboration, with reliability checks confirming strong internal consistency (Cronbach's $\alpha > .85$). Finally, semi-structured interviews with a purposive subsample ($n = 10$ per group) were conducted after the intervention to capture learner perceptions of platform usability, collaborative inquiry experiences, and perceived impacts on understanding and reflection, thereby triangulating performance outcomes, interaction analytics, survey responses, and qualitative insights to strengthen the interpretive robustness of the study.

3.3 UILA-CDCAS Platform Interface

The UILA-CDCAS platform was designed to operationalize inquiry-based programming pedagogy through an integrated digital environment that supports structured peer interaction and real-time collaborative code annotation. As illustrated in Figure 2, the platform interface provides learners with simultaneous access to inquiry-based learning resources and authentic programming tasks within a unified workspace. Foundational conceptual materials, worked examples, and scaffolded inquiry prompts are embedded within coding activities, enabling students to explore core programming principles while actively engaging in problem-solving. This co-location of instructional resources and coding tasks was intentionally structured to reduce cognitive fragmentation and support just-in-time conceptual reinforcement during programming practice. Unlike general digital coding tools, the UILA-CDCAS platform integrates collaborative annotation functionalities directly into the coding environment, enabling learners to interact with specific code segments through contextualized comments, inquiry prompts, and peer feedback. This design embeds collaborative dialogue and inquiry processes within shared code artifacts, supporting real-time co-construction of knowledge and iterative problem-solving during programming activities.

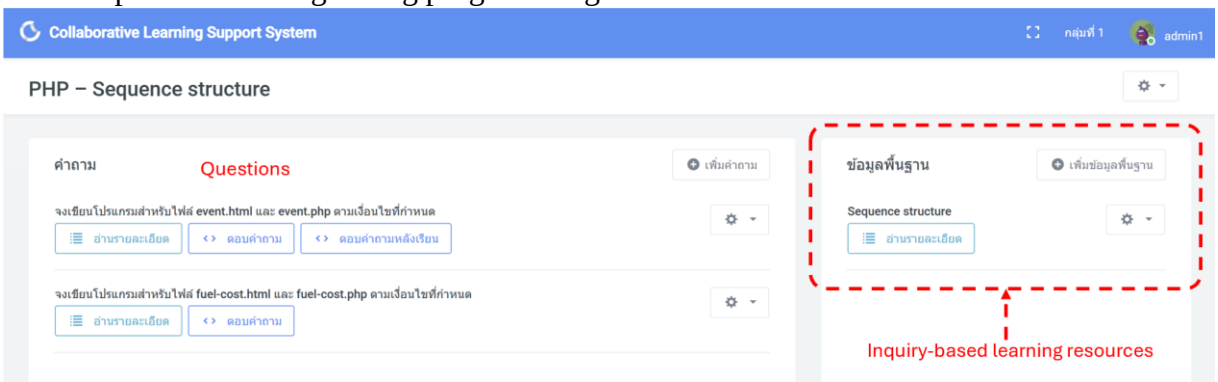

Figure 2. Inquiry-based learning resources in the UILA-CDCAS platform

During collaborative group sessions, each learner could view and interact with the programming solutions produced by peers in the same working group. As depicted in Figure 3, the platform integrates real-time annotation tools that allow students to interrogate code segments, provide immediate recommendations, and exchange targeted feedback directly within shared programming artifacts. Annotation functionalities include definitional notes, evaluative comments, inquiry prompts, and associative links, all designed to scaffold dialogic inquiry and reflective reasoning. Through these mechanisms, learners engage in clarification processes, comparison of alternative coding strategies, and justification of programming decisions, thereby transforming code review into an inquiry-driven collaborative learning activity rather than a purely evaluative exercise.

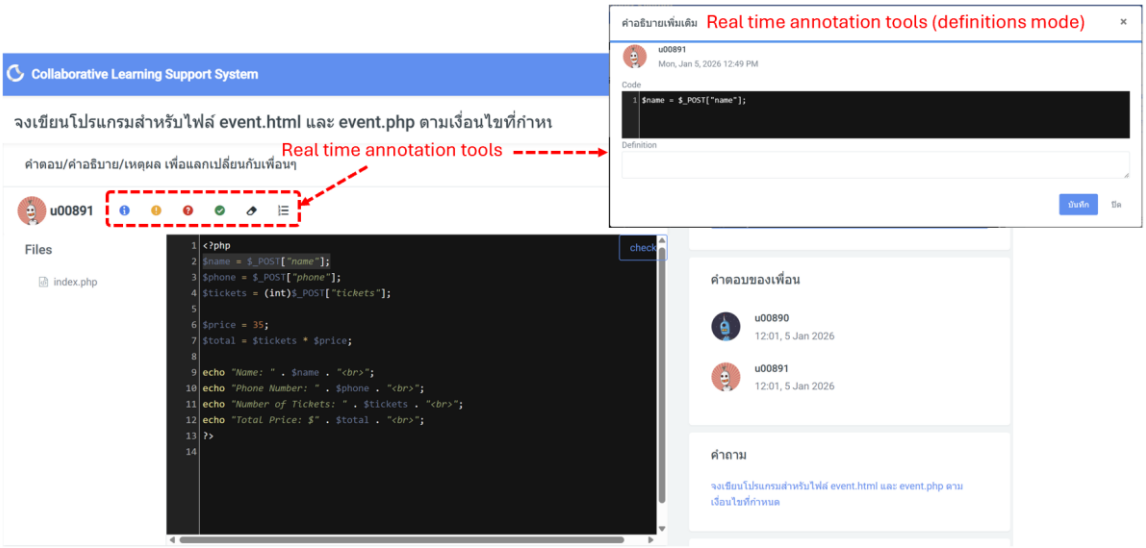

Figure 3. Real-time collaborative annotation interface of the UILA-CDCAS platform

In addition to annotation-centered interaction, the UILA-CDCAS incorporates an integrated individual-and-group coding editor coupled with a synchronous group communication interface, as shown in Figure 4. This environment enables learners to discuss solution approaches, negotiate debugging strategies, and collectively refine programming outputs through real-time textual dialogue. The integration of shared code workspaces and communication channels supports peer scaffolding, collaborative problem diagnosis, and sustained reflective engagement throughout programming tasks. The platform was deployed exclusively within the experimental condition during the eight-week instructional intervention, with its interface functionalities systematically embedded into instructional procedures to support inquiry enactment, collaborative knowledge construction, and distributed coding practice.

3.4 Procedures

The instructional intervention was implemented over eight weeks. During this period, both the experimental and control groups followed the same core curriculum, learning objectives, and programming task sequence delivered through an inquiry-based pedagogy. This design ensured comparable content exposure while isolating the technological affordances of the learning environment as the principal experimental variable. Across the eight weeks, instruction was organized into recurring inquiry cycles in which students engaged with programming problems aligned with the weekly topic, generated and tested solution approaches, articulated reasoning and debugging decisions, and revised solutions through

feedback and reflection. Within this shared instructional structure, the primary procedural distinction lay in how peer interaction and feedback were embedded in the learning workflow. In the experimental condition, students used UILA-CDCAS to annotate code, respond to peer inquiries, and engage in collaborative debugging within shared code artifacts. These interactions supported reasoning externalization and iterative refinement during inquiry cycles. In contrast, students in the control condition completed equivalent inquiry-based tasks using general digital coding tools that supported code development and submission but did not provide embedded collaborative annotation or structured peer interaction around code segments, resulting in inquiry processes being enacted primarily through individual work supplemented by instructor-mediated discussion. To maintain instructional fidelity, both groups were taught by the same instructor, who consistently applied inquiry-oriented facilitation strategies, including problem framing, guided questioning, and reflective debriefing, and implementation consistency across conditions was reinforced through the use of the same learning materials, time allocation, and assessment schedule, with the presence or absence of collaborative digital annotation functionality constituting the only systematic procedural difference between groups.

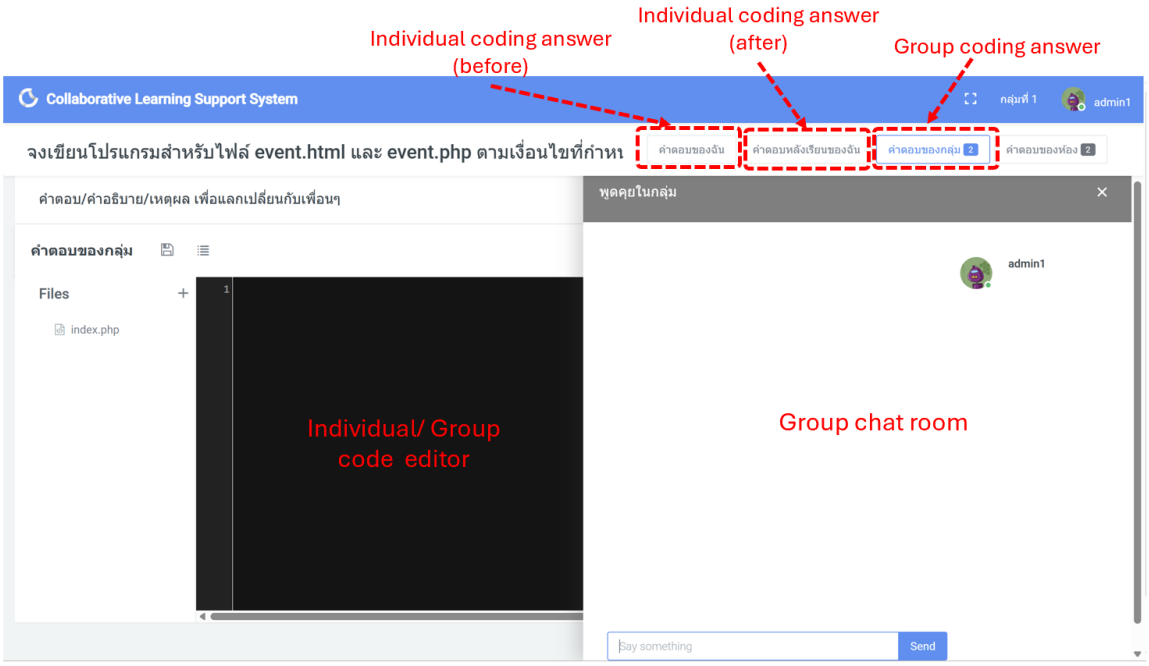

Figure 4. Integrated collaborative coding and group communication environment

3.5 Data Analysis

Quantitative learning outcome data from the pre-test and post-test were analyzed using one-way ANCOVA to adjust for baseline differences in prior programming knowledge. For each outcome (programming proficiency and conceptual understanding), post-test scores served as the dependent variable, group (UILA-CDCAS vs. UILA-GDCS) was specified as the fixed factor, and pre-test scores were included as the covariate. Prior to conducting ANCOVA, key statistical assumptions were examined, including normality of residuals, homogeneity of variance, and homogeneity of regression slopes. All assumptions were satisfied, supporting the validity of the subsequent inferential analysis. Adjusted mean post-test scores were reported to support interpretable between-group comparisons while controlling initial performance variability. To examine collaboration-related behaviors in the UILA-CDCAS condition, system-generated interaction logs were analyzed to derive behavioral indicators such as annotation frequency, peer response volume, and resolution/refinement rates of annotation

threads. These indicators were used as proxies for collaborative engagement and knowledge co-construction within the annotation-supported learning environment. Because these log-based indicators were generated by the UILA-CDCAS platform and were not equivalently available in the UILA-GDCS condition, interaction analytics were primarily analyzed using descriptive statistics to characterize participation patterns and to clarify how collaborative inquiry was enacted in practice within the experimental group. Survey responses measuring learner engagement and motivation were analyzed using descriptive statistics and paired-samples t-tests to examine within-group changes from pre to post, supplemented by between-group comparative analyses where relevant to determine whether changes observed in the experimental group exceeded those in the control group; reliability coefficients were confirmed prior to inferential testing to ensure measurement consistency. Qualitative interview data were analyzed thematically following Braun and Clarke's six-phase framework [32], involving iterative coding, theme development, and refinement to identify recurring patterns related to conceptual understanding, collaborative inquiry experiences, and the perceived pedagogical value of the annotation-supported environment. Finally, findings from ANCOVA results, interaction analytics, survey analyses, and thematic interpretation were triangulated to strengthen explanatory robustness and clarify the mechanisms linking UILA-CDCAS features to observed learning outcomes.

3.6 Preliminary Findings from Pilot Study

A preliminary pilot implementation of the UILA-CDCAS platform was conducted with a subset of participants (n = 12) prior to the full-scale study. The pilot followed the same instructional structure, task flow, and assessment schedule planned for the main intervention. Its purpose was to evaluate system feasibility, usability, and procedural readiness. The results indicated an average improvement of 18% from pre-test to post-test performance, suggesting initial learning gains under the UILA-CDCAS-supported inquiry workflow. Interaction log data further demonstrated active peer engagement, with learners producing an average of 6.4 substantive annotations per session and more than 70% of annotations receiving at least one peer response. These indicators reflect sustained dialogic exchange and reciprocal feedback, as summarized in Table 1, providing preliminary evidence of system readiness prior to the quasi-experimental deployment. Qualitative feedback from semi-structured interviews further supported these findings by revealing three key learning mechanisms: (1) clarification through peer explanations, which required learners to articulate and justify their reasoning; (2) exposure to diverse solution strategies, which expanded learners' problem-solving approaches; and (3) reflective learning, supported by the revisitable nature of annotation threads that enabled learners to track conceptual development over time. The pilot also identified areas for refinement, including the need for clearer onboarding tutorials and more structured inquiry prompts to support analytically focused discussions. Overall, these findings support the technical feasibility, pedagogical viability, and procedural readiness of UILA-CDCAS for implementation in the main study.

Table 1

Pilot Feasibility and Interaction Indicators (n = 12)

Indicator	Result	Interpretation
Mean post-test gain relative to pre-test	+18%	Preliminary evidence of learning improvement
Substantive annotations per session (mean)	6.4	Active annotation-based participation
Annotations receiving ≥1 peer response	>70%	Sustained peer-to-peer dialogue around code

4. RESEARCH RESULTS

This section presents the empirical findings from the quasi-experimental study, integrating both quantitative and qualitative data to provide a comprehensive view of the impact of the UILA-CDCAS on students' programming proficiency, conceptual understanding, collaborative engagement, and motivation. Statistical analyses are supplemented with descriptive observations to capture the nuances of learner experience.

4.1 Improvements in Programming Proficiency and Conceptual Understanding

At the pre-test stage, no significant differences were observed between the experimental and control groups in programming proficiency and conceptual understanding, indicating baseline equivalence. To evaluate learning gains while controlling for baseline differences, post-test outcomes were analyzed using ANCOVA with pre-test scores as the covariate. As shown in Table 2, both groups demonstrated improvement from pre-test to post-test; however, the experimental group achieved higher adjusted post-test means than the control group in both programming proficiency and conceptual understanding. For programming proficiency, the experimental group outperformed the control group by an adjusted mean difference of 8.3 points (82.4 vs. 74.1), with a statistically significant group effect, $F_{(1, 47)} = 15.42$, $p < .001$, and a substantial effect size (partial $\eta^2 = .247$). A similar pattern was observed for conceptual understanding, where the experimental group again demonstrated a significantly higher adjusted mean (84.7 vs. 76.5), $F_{(1, 47)} = 18.11$, $p < .001$, with a moderate-to-strong effect size (partial $\eta^2 = .278$). Collectively, these findings suggest that the UILA-CDCAS provided added instructional value in enhancing both procedural performance and conceptual understanding compared to inquiry-based instruction supported by general-purpose digital coding tools.

Table 2

Adjusted Post-Test scores by group

Outcome Measure	Group	n	Adjusted Mean	SD	$F_{(1,47)}$	p-value	Partial η^2
Programming Proficiency	Experimental (UILA-CDCAS)	25	82.4	6.8	15.42	< .001	.247
	Control (UILA-GDCS)	25	74.1	7.5			
Conceptual Understanding	Experimental (UILA-CDCAS)	25	84.7	5.9	18.11	< .001	.278
	Control (UILA-GDCS)	25	76.5	6.3			

Note. Adjusted means were obtained using ANCOVA with pre-test scores as the covariate. SD = standard deviation. Partial η^2 represents effect size.

4.2 Qualitative Insights into Conceptual Understanding

The thematic analysis identified three interrelated mechanisms through which UILA-CDCAS supported students' conceptual development during programming tasks (Table 3). First, clarification through peer explanations emerged as learners were repeatedly required to organize their thinking and articulate code logic in concise annotation notes for peers; this "learning by teaching" process strengthened their own understanding by forcing explicit reasoning, justification, and error explanation rather than tacit trial-and-error. Second, exposure to diverse solution strategies showed that encountering peers' contrasting implementations expanded learners' mental models of what constitutes an effective program, encouraging them to compare structures, evaluate trade-offs between efficiency and readability, and adopt alternative approaches when refining their own code. Third, reflective learning is enabled by

the persistent and revisitable nature of annotation threads, which allow learners to return to earlier notes, trace how misconceptions were resolved, and monitor progress across tasks; this supports metacognitive regulation by making prior mistakes and conceptual growth visible over time. Taken together, the findings suggest that collaborative annotation in UILA-CDCAS functions not only as immediate feedback but also as a sustained cognitive-social scaffold that strengthens conceptual comprehension through explanation, strategic comparison, and reflective metacognition within authentic programming practice.

Table 3

Summary of Qualitative Themes Related to Conceptual Understanding

Theme	Description	Illustrative Evidence
Clarification through Peer Explanations	Writing annotations to explain code to peers required learners to externalize and structure their reasoning (learning by teaching), leading to more apparent justification of logic, debugging decisions, and conceptual repair.	Accounts describing how composing explanations and responding to peers' questions clarified misconceptions and strengthened understanding of code logic.
Exposure to Diverse Solution Strategies	Viewing alternative peer solutions expanded learners' mental models of practical programming, supporting comparative evaluation of code structure, efficiency, and readability, and informing strategy transfer to their own solutions.	Reports of comparing different implementations and adopting or adapting peers' approaches to improve solution quality.
Reflective Learning	Persistent annotation threads enabled learners to revisit prior notes, observe error patterns, and track conceptual development over time, supporting metacognitive monitoring and consolidation beyond single-task completion.	References to returning to earlier annotations to review progress, recognize recurring mistakes, and refine understanding across sessions.

4.3 Collaborative Interactions and Peer Learning Behaviors

Interaction log data extracted from the UILA-CDCAS platform indicated a high level of collaborative engagement among learners in the experimental group. As summarized in Table 4, students generated an average of 5.8 annotations per learner per week, with 72% of annotations receiving at least one peer response. Peer replies averaged 3.1 per learner per week, while 68% of annotation threads resulted in code resolution or refinement. These indicators suggest that collaborative exchanges are not merely participatory but functionally oriented toward problem diagnosis and solution development.

A qualitative examination of the discussion content further revealed that peer interactions extended beyond surface-level commentary. Approximately half of the annotation threads centered on error identification and debugging, while the remaining discussions addressed optimization strategies, code readability, and alternative algorithmic approaches. Threads involving multiple contributors produced more sophisticated refinements, reflecting the cumulative value of distributed expertise within collaborative programming processes.

Collectively, these interaction patterns demonstrate that annotation-supported dialogue fostered active peer learning, enhancing both the frequency and the functional depth of collaborative engagement during programming activities.

Table 4

Summary of Collaborative Interaction Metrics in the Experimental Group

Interaction Metric	Mean / Percentage
Annotations per learner per week	5.8
Annotations receiving peer comments	72%
Peer replies per learner per week	3.1
Annotation threads leading to resolution/refinement	68%

4.4 Engagement and Motivation

Survey analyses using paired-samples t-tests revealed a significant increase in self-reported engagement within the experimental group (pre: M = 3.48; post: M = 4.21 on a 5-point Likert scale; $t(24) = -6.15$, $p < .001$), whereas no significant change was observed in the control group. Motivation scores demonstrated a similar upward trend among experimental participants (pre: M = 3.55; post: M = 4.12; $t(24) = -4.92$, $p < .001$). As summarized in Table 5, these results indicate that participation in the UILA-CDCAS environment is associated with meaningful improvements in learners' affective engagement with programming tasks.

Qualitative interview data further contextualized these findings. Participants consistently identified real-time peer feedback, the visibility of collaborative contributions, and a heightened sense of shared accountability as key motivational drivers. Several learners noted that the social dimension introduced through collaborative annotation—largely absent in conventional individual coding activities—renders the learning process more engaging, interactive, and personally meaningful.

The convergence of statistical outcomes, interaction evidence, and learner perceptions provides robust support for the motivational affordances of the UILA-CDCAS. By enabling immediate dialogic exchange around code, the platform fosters sustained participation, reinforces intrinsic motivation, and enhances learners' sense of belonging within the programming learning community. Collectively, these findings further substantiate the pedagogical value of embedding collaborative digital annotation into inquiry-based programming instruction, highlighting its potential as a promising approach for enhancing engagement in programming education within similar instructional contexts.

Table 5

Changes in Engagement and Motivation Scores within the Experimental Group

Measure	n	Pre-test Mean	Post-test Mean	t(df)	p-value
Engagement	25	3.48	4.21	$-6.15\ (24)$	< .001
Motivation	25	3.55	4.12	$-4.92\ (24)$	< .001

Note. Scores were measured using a 5-point Likert scale. Results are based on paired-samples t-tests comparing pre-test and post-test scores within the experimental group.

5. DISCUSSION

The observed learning gains should not be interpreted solely as evidence of system effectiveness. Instead, they provide design-oriented insights into how collaborative digital annotation can support inquiry-based programming education. Significant improvements in programming proficiency and conceptual understanding suggest that embedding inquiry processes within shared coding artifacts strengthens learners' capacity to articulate reasoning, negotiate meaning, and refine solutions through situated problem-solving. Interaction log and interview analyses further illuminate three learning-supportive design mechanisms: structured

peer annotation fostered explicit articulation aligned with dialogic learning; exposure to alternative coding strategies broadened problem-solving repertoires; and the revisitable nature of annotations enabled longitudinal reflection and conceptual consolidation. These findings align with prior research indicating that collaborative coding platforms are associated with improvements in computational thinking and problem-solving practices [3]. In addition, annotation-supported peer interaction has been shown to promote active engagement and knowledge construction in programming contexts [19]. However, existing studies also highlight important considerations related to digital access and implementation constraints [29]. Similarly, studies on technology-supported inquiry and collaborative learning environments report that embedding investigative dialogue within authentic disciplinary tasks promotes deeper reasoning and shared knowledge construction [6], [26]. From a pedagogical design perspective, the UILA-CDCAS illustrates how collaborative inquiry can be operationalized in a structured manner within similar instructional contexts: real-time feedback supported timely identification of misconceptions, while visible peer contributions fostered accountability and mirrored professional collaborative software development practices. Nevertheless, limitations include the quasi-experimental design, single-institution context, potential novelty effects, and reliance on self-reported engagement measures. Future research should pursue longitudinal, multi-institutional validation, examine how specific annotation behaviors predict learning trajectories, and explore AI-supported analytics to enhance adaptive feedback and peer coordination in inquiry-based collaborative programming environments.

6. CONCLUSION

This study contributes to the field of information and education technology by demonstrating how a collaborative digital coding annotation system can be purposefully designed and enacted to support inquiry-based learning in programming education. Beyond documenting performance gains, the findings provide design-level evidence that embedding inquiry processes within shared coding artifacts transforms individual programming tasks into socially mediated learning experiences that foster conceptual understanding, reflective practice, and collaborative competence. By making peer reasoning visible and persistent, the UILA-CDCAS operationalizes inquiry through technology affordances that connect dialogue directly to authentic coding activity. From a pedagogical design perspective, the results underscore the importance of aligning collaborative technological tools with disciplinary practices and inquiry goals; when collaborative functions are tightly integrated with problem-solving workflows, they extend learning beyond task completion toward collective knowledge construction and sustained reflection. Such design principles position collaborative annotation systems not merely as supplementary communication tools but as core infrastructures supporting inquiry-rich programming environments. More broadly, the study suggests the potential applicability of collaborative annotation as a design pattern for related domains characterized by iterative problem-solving and peer review, including engineering design, data science, and research-based writing. As educational technologies continue to evolve – particularly through the integration of AI-driven analytics – design-oriented research of this kind provides a foundation for developing adaptive, inquiry-centered learning ecosystems that bridge theory, practice, and professional relevance.

REFERENCES (TRANSLATED AND TRASLITERATED)

[1] A. S. Elshall and A. Badir, “Balancing AI-assisted learning and traditional assessment: the FACT assessment in environmental data science education,” *Frontiers in Education*, vol. 10, Art. no. 1596462, Jun. 2025, doi: 10.3389/feduc.2025.1596462. (in English)

[2] T. Lee, K. Pham, A. Crosby, and J. F. Peterson, "Digital collaboration in design education: how online collaborative software changes the practices and places of learning," *Pedagogy, Culture & Society*, vol. 29, no. 2, pp. 231–245, 2020, doi: 10.1080/14681366.2020.1714700. (in English)
[3] V. Dolgopolovas and V. Dagienė, "Competency-based TPACK approaches to computational thinking and integrated STEM: A conceptual exploration," *Computer Applications in Engineering Education*, vol. 32, 2024, doi: 10.1002/cae.22788. (in English)
[4] Y. G. Tedla and H.-L. Chen, "The impacts of computer-supported collaborative learning on students' critical thinking: a meta-analysis," *Education and Information Technologies*, vol. 30, pp. 1487–1516, 2024, doi: 10.1007/s10639-024-12857-y. (in English)
[5] M. B. Thomas, A. Muscat, A. Zuccolo, C. N. Luguetti, and A. Watt, "Navigating pedagogical innovation in higher education: Education academics' experiences with active and inquiry-based learning in intensive teaching," Innovative Higher Education, vol. 50, pp. 1917–1943, 2025, doi: 10.1007/s10755-025-09807-y. (in English)
[6] J. E. Hinostroza, S. Armstrong-Gallegos, and M. Villafañe, "Roles of digital technologies in the implementation of inquiry-based learning (IBL): A systematic literature review," *Social Sciences & Humanities Open*, 2024, doi: 10.1016/j.ssaho.2024.100874. (in English)
[7] P. Beldar, R. Rakhade, M. Bahiram, and S. Kadbhane, "Innovative coding teaching methodologies: A comprehensive approach for diverse learners," *Journal of Engineering Education Transformations*, vol. 39, no. 2, 2025, doi: 10.16920/jeet/2025/v39i2/25141. (in English)
[8] C. D. Moemeke, J. N. Chukwunenye, and N. A. Malik, "Understanding the theoretical foundations of inquiry-based learning: Pivot for productive science education in Africa," *Educational Considerations*, 2025, doi: 10.4148/0146-9282.2446. (in English)
[9] S. K. Sata, "Advancing analytical and problem-solving skills for STEM-related challenges: Exploring innovative educational strategies and practical applications," *World Journal of Arts, Education and Literature*, vol. 1, no. 2, pp. 32–38, Oct. 2024. (in English)
[10] 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 & E-Learning: An International Journal*, 2019, doi: 10.34105/j.kmel.2019.11.016. (in English)
[11] C. Halkiopoulos and E. Gkintoni, "Leveraging AI in e-learning: Personalized learning and adaptive assessment through cognitive neuropsychology – A systematic analysis," *Electronics*, vol. 13, no. 18, Art. no. 3762, 2024, doi: 10.3390/electronics13183762. (in English)
[12] A. Amirbek, Y. Torebek, M. Abdualiyeva, S. Altynbekov, A. Tursynbayev, R. Abdraimov, and G. Omashova,"Cloud-based learning and cognitive development among lower-secondary students (Grades 8–9): Evidence from schools in Kazakhstan," *Frontiers in Education*, vol. 11, Art. no. 1749929, Feb. 2026, doi: 10.3389/feduc.2026.1749929. (in English)
[13] C. B. Omeh and M. A. Ayanwale, "Artificial intelligence meets PBL: Transforming computer-robotics programming motivation and engagement," *Frontiers in Education*, vol. 10, Art. no. 1674320, Nov. 2025, doi: 10.3389/feduc.2025.1674320. (in English)
[14] K. Lu, T. Ji, F. Lu, and R. Shadiev, "Bridging the digital divide: The mediating role of learning engagement between technology usage approaches and higher order thinking skills in a technology-enhanced inquiry-based learning environment," *Educational Technology Research and Development*, vol. 73, pp. 2929–2949, 2025, doi: 10.1007/s11423-025-10533-7. (in English)
[15] J. Tan, L. Wu, and S. Ma, "Collaborative dialogue patterns of pair programming and their impact on programming self-efficacy and coding performance," *British Journal of Educational Technology*, vol. 55, pp. 1060–1081, 2023, doi: 10.1111/bjet.13412. (in English)
[16] R. Hoda, "Toward agentic software engineering beyond code: Framing vision, values, and vocabulary," arXiv preprint arXiv:2510.19692, 2025, doi: 10.48550/arXiv.2510.19692. (in English)
[17] S. Ljubić, A. Skoki, F. Hržić, and A. Salkanović, "Automated online assessment and cloud-based programming: Advancing computer engineering education," in Proc. International Workshop on Higher Education Learning Methodologies and Technologies Online, 2023, doi: 10.1007/978-3-031-67351-1_6. (in English)
[18] N. S. Baanqud, H. Al-Samarraie, A. I. Alzahrani, and O. Alfarraj, "Engagement in cloud-supported collaborative learning and student knowledge construction: A modeling study," *International Journal of Educational Technology in Higher Education*, vol. 17, Art. no. 23, Dec. 2020, doi: 10.1186/s41239-020-00232-z. (in English)
[19] S. H. Tuiloma, K. T. Arnesen, Q. Guo, R. West, and C. R. Graham, "Promoting active learning through collaborative annotation tools," *The Journal of Applied Instructional Design*, 2025, doi: 10.59668/2224.22361. (in English)

[20] M. R. Jamhari and A. Juhana, “Enhancing asynchronous code review in programming education: A systematic literature review of visual-contextual feedback systems,” *Jurnal Teknologi Informasi dan Pendidikan*, vol. 19, no. 1, 2026, doi: 10.24036/jtip.v19i1.1091. (in English)
[21] A. Anand, A. Gupta, N. Yadav, and S. Bajaj, “A comprehensive survey of AI-driven advancements and techniques in automated program repair and code generation,” arXiv preprint arXiv:2411.07586, 2024, doi: 10.48550/arXiv.2411.07586. (in English)
[22] Z. Zulpykhar, A. Sadvakassova, Z. Zhalgasbekova, Z. Zhantassova, and A. Kadyrova, “The role of cloud-based collaborative tools in fostering digital teamwork and engagement in undergraduate computer science courses,” *International Journal of Information and Education Technology*, vol. 16, no. 2, pp. 2514–2523, 2026, doi: 10.18178/ijiet.2026.16.2.2514. (in English)
[23] M. M. Fakhri, A. S. Ahmar, R. Rosidah, D. Fadhilatunisa, and M. I. Tabash, “Barriers to effective learning: Examining the influence of delayed feedback on student engagement and problem solving skills in ubiquitous learning programming,” *Journal of Applied Science, Engineering, Technology, and Education*, vol. 4, no. 5, pp. 2825–2835, 2024, doi: 10.35877/454ri.asci2825. (in English)
[24] M. Kyambade, A. Namatovu, and A. Male Ssentumbwe, “Exploring the evolution of artificial intelligence in education: From AI-guided learning to learner-personalized paradigms,” *Cogent Education*, vol. 12, Art. no. 2505297, 2025, doi: 10.1080/2331186X.2025.2505297. (in English)
[25] C.-M. Chen, M.-C. Li, and Y.-T. Chen, “The effects of web-based inquiry learning mode with the support of collaborative digital reading annotation system on information literacy instruction,” *Computers & Education*, vol. 179, Art. no. 104428, 2022, doi: 10.1016/j.compedu.2021.104428. (in English)
[26] L. S. Vygotsky and M. Cole, Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA, USA: Harvard University Press, 1978. (in English)
[27] F. G. Neto and F. Dobslaw, “Building collaborative learning: Exploring social annotation in introductory programming,” in Proc. IEEE/ACM 46th Int. Conf. Software Engineering: Software Engineering Education and Training (ICSE-SEET), 2024, pp. 12–21, doi: 10.1145/3639474.3640063. (in English)
[28] [28] R. Koitz-Hristov, “Peer code review methods: An experience report from a data structures and algorithms course,” in Proc. 56th ACM Technical Symposium on Computer Science Education (SIGCSE), 2025, doi: 10.1145/3641554.3701799. (in English)
[29] A. Kalita and P. Baishya, “Technology-mediated learning and the digital divide: Challenges faced by scheduled tribe students in higher education in India,” *International Journal for Multidisciplinary Research*, vol. 7, no. 4, 2025, doi: 10.36948/ijfmr.2025.v07i04.54371. (in English)
[30] H. Murillo-Jiménez, M. Centeno-Alarcón, J. Buele, and F. Yumbla, “Analyzing barriers to the effective implementation of technological tools in inclusive education: A scoping review,” *Frontiers in Education*, 2025, doi: 10.3389/feduc.2025.1687664. (in English)
[31] I. Purnama, R. F. Wijaya, A. Harahap, and F. Edi, “Artificial intelligence as a science teacher assistant: An analysis of machine learning utilization in diagnosing student misconceptions: A review,” *Jurnal Penelitian Pendidikan IPA*, vol. 11, no. 12, 2025, doi: 10.29303/jppipa.v11i12.13089. (in English)
[32] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006, doi: 10.1191/1478088706qp063oa. (in English)

Text of the article was accepted by Editorial Team 17.02.2026

ПРОЄКТУВАННЯ ТА ВПРОВАДЖЕННЯ ПОВСЮДНО ДОСТУПНОЇ СИСТЕМИ АНОТУВАННЯ ПРОГРАМНОГО КОДУ ДЛЯ ДОСЛІДНИЦЬКО ОРІЄНТОВАНОГО СПІЛЬНОГО НАВЧАННЯ ПРОГРАМУВАННЯ

Кріттавая Тонгку
доктор філософії з освіти в галузі науки та технологій, доцент
Коледж мистецтв, медіа та технологій, Університет Чіангмая, м. Чіангмай, Таїланд
ORCID ID 0000-0003-2167-658X
krittawaya.t@cmu.ac.th
Канніка Даунгчароне
доктор філософії з освіти в галузі науки та технологій, доцент
Коледж мистецтв, медіа та технологій, Університет Чіангмая, м. Чіангмай, Таїланд
ORCID ID 0000-0002-4108-2087
kannika.d@cmu.ac.th

Анотація. Зростаюча складність навчання комп’ютерному програмуванню посилила потребу в цифрових навчальних середовищах, які сприяють не лише розвитку технічних

навичок, а й спільним дослідницьким пошукам, рефлексивній практиці та глибшому осмисленню концептуальних засад. У відповідь на цю потребу в межах даного дослідження було розроблено, впроваджено та емпірично оцінено універсальне навчальне середовище, засноване на дослідницькому підході та інтегроване з системою спільного цифрового анотування коду, відоме як UILA-CDCAS. Платформа була розроблена з метою вбудовування дослідницьких процесів безпосередньо в реальні завдання з програмування. Вона дозволяє учням анотувати сегменти коду, обмінюватись цілеспрямованими відгуками однолітків, ставити питання, що стосуються конкретного контексту, та брати участь у спільній роботі над вирішенням проблем у режимі реального часу як у класі, так і в умовах дистанційного навчання. Це дослідження має на меті оцінити ефективність системи в покращенні результатів навчання програмуванню та залученості учнів. Було проведено квазіекспериментальне дослідження за участю 50 студентів першого курсу університету. Для контролю відмінностей у попередніх знаннях було використано аналіз коваріації. Результати дослідження свідчать про те, що студенти в експериментальній групі продемонстрували значно вищі показники успішності в освоєнні навичок програмування та концептуального розуміння. Крім того, аналіз взаємодії в межах спільного навчання та оцінка сприйняття учнів виявили підвищену залученість та стійку мотивацію. Вони також вказали на посилення процесу формування знань за підтримки однолітків та більш змістовні практики рефлексивного навчання, пов’язані з діяльністю з анотування коду. У сукупності ці результати підкреслюють педагогічну та технологічну цінність інтеграції спільного анотування коду в різні моделі навчання, засновані на дослідженні. Вони також надають емпіричні та орієнтовані на дизайн ідеї для розробки цифрових платформ, що підтримують навчання програмуванню та навчання з використанням технологій.

Ключові слова: навчання, засноване на дослідженні; спільне анотування коду; середовища навчання програмуванню; спільне навчання з використанням технологій; навчання програмуванню, орієнтоване на дослідження

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.