International Journal of Research Studies in Education
2026 Volume 15 Number 23, 239-252



A preliminary evaluation of CodeChum as a formative assessment platform

Napalit, Arcely
Holy Angel University, Philippines (anapalit@hau.edu.ph)
Rabena, Mary Jane ✉
Holy Angel University, Philippines (mrabena@hau.edu.ph)

ISSN: 2243-7703
Online ISSN: 2243-7711

OPEN ACCESS

Received: 13 October 2026 **Revised**: 13 November 2026 **Accepted**: 18 November 2026
Available Online: 20 November 2026 **DOI**: 10.5861/ijrse.2026.26425

Abstract

This study evaluated CodeChum as a formative assessment platform used in programming courses in the School of Computing. Specifically, it examined students' perceptions of CodeChum in terms of system usability, user experience, learning experiences, and user satisfaction. A descriptive research design was employed involving 52 second-year Bachelor of Science in Computer Science students who had used CodeChum since its implementation during Academic Year 2025–2026. Data were collected through the System Usability Scale (SUS), a researcher-developed questionnaire, and open-ended questions. Descriptive statistics and Cronbach's alpha were used to analyze the quantitative data, while thematic analysis was employed for the qualitative responses. The findings showed that CodeChum obtained a SUS score of 49.7, indicating poor usability. User experience, learning support, engagement and motivation, skill development, and user satisfaction were all interpreted as neutral. Reliability analysis showed acceptable to excellent internal consistency across all constructs. Qualitative responses revealed that students appreciated the platform's accessibility, hands-on programming activities, automated feedback, and gamified features. However, several respondents reported concerns related to strict output checking, case sensitivity, grading inconsistencies, and problematic test cases. The study concludes that CodeChum provides useful support for programming instruction but requires improvements in usability, feedback mechanisms, and assessment features to enhance students' overall experiences with the platform.

Keywords: CodeChum, formative assessment, system usability, user experience, programming education, user satisfaction

© The Author(s) / Attribution CC BY

A preliminary evaluation of CodeChum as a formative assessment platform

1. Introduction

Programming is a fundamental component of Computer Science education because it develops students' problem-solving, analytical, and computational thinking skills. Through programming courses, students learn how to analyze problems, develop algorithms, and create software solutions for real-world situations. Despite its importance, programming remains one of the most challenging subjects for many learners due to the complexity of programming concepts, logical reasoning requirements, and problem-solving demands. Studies have shown that students often struggle with understanding programming concepts, writing code, debugging errors, and developing effective solutions to programming problems (Islam et al., 2019; Napalit et al., 2023). These challenges may negatively affect students' confidence, motivation, and academic performance. Similarly, Alakeel (2015) emphasized that differences in learners' backgrounds and learning environments contribute to the difficulties encountered in programming courses. Recent studies further highlight the challenges faced by novice programmers. Tlale-Mkhize (2025) found that students often struggle with programming concepts and problem-solving, while Singh (2024) reported that novice programmers commonly experience anxiety and disengagement. Similarly, Kalelioğlu and Keskinkılıç (2026) identified misconceptions and weak problem-solving skills as common learning barriers. Pan and Harun (2025) further noted that programming anxiety can negatively affect students' engagement and academic performance.

Educational institutions have increasingly adopted technology-enhanced learning to support programming instruction and create more interactive learning environments. Monserate (2018) noted that technology can enhance students' learning experiences and academic performance when integrated effectively into instruction. Similarly, Li et al. (2025) found that technology-supported programming instruction improved students' learning outcomes, motivation, and self-efficacy. Silva et al. (2020) also highlighted the role of technology in promoting engagement, collaboration, and active participation among programming students. Alongside technology integration, formative assessment has emerged as an important instructional strategy in programming education. Through continuous feedback, students can monitor their progress, identify mistakes, and improve their understanding of programming concepts. Studies have shown that formative assessment contributes to increased engagement, confidence, and programming proficiency among learners (Thangaraj, 2022; Thangaraj et al., 2024). Furthermore, automated formative assessment systems provide immediate feedback that enables students to identify errors and improve their programming solutions (Liebenberg, 2023). In a review of online formative assessment practices in programming courses, Munisamy et al. (2024) highlighted the role of formative assessment platforms in supporting continuous learning, monitoring student progress, and providing timely instructional support. Similarly, Wan (2025) emphasized that online assessment systems can strengthen formative evaluation in programming education by facilitating continuous feedback and helping instructors monitor students' learning development.

Recognizing the potential of technology-enhanced learning and formative assessment, the School of Computing adopted CodeChum as a programming learning and assessment platform. CodeChum has been integrated into several programming courses, including Logic Formulation and Programming, Programming with Data Structures, Object-Oriented Programming, and Data Structures and Algorithms. The platform provides coding exercises, automated code evaluation, instant feedback, progress monitoring, and gamified learning activities that support student learning and skill development. Previous studies have reported positive perceptions of CodeChum's usability and usefulness among both students and instructors, highlighting its potential to improve student engagement and programming performance (Maranga et al., 2019). Likewise, adaptive assessment features integrated into the platform were found to enhance student performance by aligning programming tasks with learners' abilities (Maranga et al., 2024). Arellano and Serato (2023) also reported favorable student perceptions

regarding the use of CodeChum as an online assessment tool.

Since the adoption of CodeChum in the School of Computing during Academic Year 2025–2026, the platform has been integrated into programming courses beginning in the first year of the Computer Science program. The second-year Computer Science students represent the first cohort of learners who have used CodeChum across multiple programming courses since its implementation. Having utilized the platform in subjects such as Logic Formulation and Programming, Programming with Data Structures, Object-Oriented Programming, and Data Structures and Algorithms, these students have gained substantial experience with its features, assessment mechanisms, and learning activities. As a result, they are in a suitable position to provide meaningful insights regarding the platform's usability, user experience, learning support, and overall satisfaction.

Despite the positive findings reported in previous studies, limited research has comprehensively examined students' perceptions of CodeChum in terms of system usability, user experience, learning experiences, and user satisfaction. While previous studies have primarily discussed CodeChum as an online assessment tool, few have evaluated the platform using multiple dimensions of student experience. Moreover, limited local evidence exists regarding the experiences of students who have used CodeChum across several programming courses. This study addresses these gaps by providing a preliminary evaluation of CodeChum as a formative assessment platform using both quantitative and qualitative data. Specifically, it examines system usability, user experience, learning support, engagement and motivation, skill development, and user satisfaction among second-year Computer Science students who have extensive experience using the platform. The findings of the study may provide baseline information for instructors, administrators, and platform developers in improving the implementation and effectiveness of CodeChum in programming courses. Furthermore, the study contributes to the growing body of knowledge on formative assessment technologies and may serve as a reference for future research on technology-enhanced learning and programming education.

2. Review of Related Literature

Programming Education and Student Learning Challenges - Programming education has become increasingly important as digital technologies continue to influence modern education and professional practice. However, learning programming remains a challenging experience for many students because it requires logical reasoning, analytical thinking, and problem-solving skills. Studies have shown that students frequently struggle with programming concepts, algorithm development, debugging, and code implementation. Ahmad and Ghazali (2020) noted that introductory programming is particularly difficult for beginners because of the challenges associated with understanding abstract concepts and applying them in practical situations. Similarly, Almdahem (2024) found that many students perceive programming as a difficult subject due to limited understanding and insufficient learning support. Abu Bakar et al. (2025) further identified pedagogical, cognitive, and technological factors as major challenges affecting programming education. Likewise, Ou et al. (2023) reported that although students generally show interest in programming, weak foundational knowledge often hinders learning and skill development.

Programming learning challenges extend beyond conceptual difficulties and often affect students' confidence, engagement, and academic success. Students frequently encounter problems related to programming environments, debugging, error interpretation, and problem-solving tasks. Wang et al. (2023) highlighted that factors such as problem difficulty and sequencing significantly influence student performance and cognitive load when using online programming systems. These findings suggest that learners require adequate instructional support, appropriate learning resources, and effective technological tools to develop programming competence successfully. As programming continues to play a central role in Computer Science education, addressing these challenges remains essential to improving student learning experiences and outcomes.

Technology-Enhanced Programming Education - Technology-enhanced learning has become an important approach in programming education as institutions seek to create more interactive and student-centered learning

environments. Rather than relying solely on traditional lectures, technology-enabled platforms provide opportunities for collaboration, practice, feedback, and active participation. Ivanović et al. (2017) noted that technology-enhanced learning supports interactive learning experiences and helps address limitations commonly associated with direct instruction. Similarly, Fanchamps and Van der Wijk (2023) found that technology-enhanced programming environments contribute positively to the development of computational thinking skills, particularly when combined with effective teacher guidance and scaffolding. Jiang et al. (2020) further reported that technology-enhanced instructional strategies improved student interaction, increased participation in programming activities, enhanced problem-solving abilities, and reduced programming anxiety. These findings highlight the potential of technology to create learning environments that are more engaging, supportive, and conducive to programming skill development.

Recent innovations have further expanded the role of technology in programming education by introducing immersive, adaptive, and intelligent learning experiences. Lu and Hu (2025) demonstrated that digital twin-based programming environments significantly improved student engagement and coding performance, particularly among novice learners. Likewise, Mposula and Oyetade (2025) emphasized the value of game-based learning in increasing student engagement, conceptual understanding, and skill acquisition in programming courses. Advances in artificial intelligence have also transformed programming education by supporting instructional design, assessment, feedback generation, and performance monitoring. Manorat et al. (2025) found that artificial intelligence and machine learning technologies are increasingly being used to enhance programming instruction and support learners throughout the learning process. Collectively, these studies suggest that technology-enhanced learning environments can improve engagement, motivation, computational thinking, and programming skills, making them valuable tools in programming education.

Formative Assessment in Programming Education - Formative assessment has become an important approach in programming education because it provides continuous feedback that helps students monitor their learning progress and improve their understanding of programming concepts. Unlike summative assessments, which evaluate learning at the end of instruction, formative assessments support learning throughout the process. Jha et al. (2023) emphasized that continuous feedback helps students improve their learning experiences and promotes a culture of ongoing learning in programming courses. Similarly, Thangaraj et al. (2025) found that adaptive formative assessment can improve student motivation, confidence, and understanding of programming by helping learners recognize and address common programming errors. Stanja et al. (2023) further highlighted that formative assessment enables educators to better understand students' learning needs and provide appropriate instructional support.

The integration of technology has further strengthened the effectiveness of formative assessment in programming education. Sun et al. (2019) found that formative assessment through peer code review contributed to improvements in students' programming abilities and encouraged active participation. Likewise, Wan (2025) reported that online assessment systems provide immediate feedback and positively influence student learning performance. Araujo et al. (2025) also demonstrated that automated formative feedback supports self-assessment and helps students improve their understanding of algorithms and data structures. These findings suggest that formative assessment, particularly when supported by technology, plays a significant role in enhancing engagement, confidence, learning outcomes, and programming proficiency.

CodeChum as a Programming Learning and Assessment Platform - Students' perceptions play an important role in the successful adoption of educational technologies. Learners are more likely to use and benefit from technological tools when these are perceived as useful, easy to use, and supportive of learning. Ubaidillah et al. (2020) found that perceived ease of use and usefulness positively influenced students' acceptance of technology in higher education. Similarly, Correia and Good (2023) reported that educational technology utilization was associated with positive learning experiences, progress toward learning objectives, and overall course satisfaction. Studies have also shown that students generally view educational technologies positively when they enhance engagement, support learning, and provide meaningful learning experiences (Minga & Ghosh, 2024; Ragab et al.,

2025).

These findings are relevant to programming learning platforms such as CodeChum, which combine instructional support, assessment, and feedback within a single environment. Developed as an online learning and monitoring platform for programming education, CodeChum enables students to complete coding activities, receive automated feedback, monitor their progress, and engage with gamified learning features (Maranga et al., 2019). The platform was designed to address common challenges in programming education while supporting continuous learning through formative assessment. More recently, adaptive assessment capabilities integrated into CodeChum were found to improve student engagement and performance by providing programming activities that match learners' skill levels (Maranga et al., 2024). Likewise, Arellano and Serato (2023) reported that Computer Science students generally perceived CodeChum as a useful online assessment tool that supports their learning experiences. These studies suggest that CodeChum has the potential to support programming instruction while promoting positive learning experiences and student engagement.

Objectives of the Study - This study aims to conduct a preliminary evaluation of CodeChum as a formative assessment platform used in programming courses within the School of Computing. Specifically, the study seeks to examine how students perceive the platform in terms of its usability, user experience, learning support, and overall satisfaction. Specifically, it aims to:

- Determine the level of system usability of CodeChum using the System Usability Scale (SUS).
- Examine the user experience of Computer Science students while using CodeChum.
- Assess students' learning experiences with CodeChum in terms of Learning Support, Engagement and Motivation, and Skill Development
- Determine the level of user satisfaction with CodeChum as a programming learning and assessment platform.
- Identify the perceived strengths, challenges, and areas for improvement of CodeChum based on students' responses to the open-ended questions.

Scope and Delimitation of the Study - This study focused on the preliminary evaluation of CodeChum as a formative assessment platform used in programming courses within the School of Computing. It examined students' perceptions of CodeChum in terms of system usability, user experience, learning experiences, and user satisfaction. The respondents were 52 second-year Bachelor of Science in Computer Science students who had used CodeChum since its adoption during Academic Year 2025–2026 and had experienced the platform across several programming courses. Data were gathered through an online questionnaire consisting of the System Usability Scale (SUS), researcher-developed items on user experience, learning experiences, and user satisfaction, as well as open-ended questions. The study was limited to students' perceptions and experiences with CodeChum and did not examine programming competency, academic performance, or learning outcomes. Likewise, the findings were limited to second-year Computer Science students from the School of Computing and may not be generalizable to other populations or institutions.

3. Methods

Participants of the Study - The study involved second-year Bachelor of Science in Computer Science students from the School of Computing. These students represented the first cohort who had used CodeChum since its adoption during Academic Year 2025–2026. They had utilized the platform in several programming courses, including Logic Formulation and Programming, Programming with Data Structures, Object-Oriented Programming, and Data Structures and Algorithms. A total of 63 students comprised the target population of the study. Since the population was relatively small and accessible, total enumeration was employed. Of the 63 students invited to participate, 52 completed the questionnaire, resulting in a response rate of 82.54%.

Research Instrument - The study used a structured questionnaire to gather data on students' perceptions of CodeChum as a formative assessment platform. The questionnaire consisted of five sections that measured system usability, user experience, learning experiences, user satisfaction, and participants' qualitative feedback regarding their experiences with the platform. The first section employed the System Usability Scale (SUS) developed by Brooke (1996), a widely used instrument for assessing the usability of software systems and digital platforms. The second section measured user experience, focusing on students' perceptions of the platform's interface, navigation, responsiveness, accessibility, and overall interaction. The third section assessed learning experiences in terms of learning support, engagement and motivation, and skill development. The fourth section focused on user satisfaction, which examined students' overall satisfaction with CodeChum as a programming learning and assessment platform. The last section included open-ended questions that allowed students to share their experiences, identify challenges they encountered, and provide suggestions for improving the platform. All closed-ended items were rated using a five-point Likert scale, where 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree.

Before the administration of the questionnaire, the instrument was reviewed by a licensed psychologist to assess its content, clarity, and overall tone. Based on the recommendations provided, several statements were revised to improve wording and remove double-barreled questions. However, no items were removed from the instrument. The revised questionnaire was subsequently administered to the target respondents of the study. For data organization and statistical analysis, each questionnaire item was assigned to a corresponding code as presented in Table 1. These codes were used during data processing, reliability analysis, and presentation of results.

To determine the reliability of the instrument, Cronbach's alpha was computed using the responses of the 52 participants, as presented in Table 1. The System Usability Scale obtained a reliability coefficient of 0.739, indicating acceptable internal consistency. The User Experience scale yielded a coefficient of 0.959, while the Learning Experience subscales obtained coefficients of 0.932 for Learning Support, 0.932 for Engagement and Motivation, and 0.944 for Skill Development. The User Satisfaction scale obtained a Cronbach's alpha of 0.971. These results indicate that the questionnaire demonstrated acceptable to excellent reliability and was suitable for assessing students' perceptions of CodeChum.

Table 1
Reliability Coefficients of the Research Instrument

Construct	Item Codes	Cronbach's Alpha	Interpretation
System Usability Scale (SUS)	SUS1-SUS10	0.739	Acceptable
User Experience	UE1-UE10	0.959	Excellent
Learning Experience - Learning Support	LELS1-LELS5	0.932	Excellent
Learning Experience – Engagement and Motivation	LEEM1-LEEM5	0.932	Excellent
Learning Experience – Skills Development	LESD1-LESD5	0.944	Excellent
User Experience	US1-US8	0.971	Excellent

Data Collection - Data was collected through an online survey questionnaire administered using Google Forms. Before completing the questionnaire, participants were informed about the purpose of the study and the voluntary nature of their participation. An informed consent statement was provided at the beginning of the survey, and only those who agreed to participate proceeded with the questionnaire. The survey link was distributed to the target respondents, and responses were collected during the designated data collection period. After the survey was closed, the responses were retrieved, organized, and prepared for statistical analysis.

Data Analysis - The quantitative data gathered from the questionnaire were analyzed using descriptive statistical techniques with the aid of Jamovi Version 2.7 (The jamovi Project, 2024). Frequency and percentage were used to summarize responses where applicable. Mean and standard deviation were computed to determine the level of system usability, user experience, learning experiences, and user satisfaction with CodeChum as presented in Table 2.

Table 2
Likert Scale for Interpretation

Scale	Mean Range	Interpretation
5	4.21-5.00	Strongly Agree
4	3.41 – 4.20	Agree
3	2.61 – 3.40	Neutral
2	1.81 – 2.60	Disagree
1	1.00 – 1.80	Strongly Disagree

The System Usability Scale (SUS) was scored using the standard SUS scoring procedure to obtain the overall usability score of CodeChum. Responses to the open-ended questions were analyzed through thematic analysis. Similar responses were grouped according to recurring ideas and patterns to identify common strengths, challenges, and suggestions related to the use of CodeChum.

Ethical Considerations - The study was conducted in accordance with accepted ethical research practices. Participation was voluntary, and the respondents were informed about the purpose of the study before completing the questionnaire. They were also informed that they could choose not to participate or discontinue their participation at any point without any negative consequences. The confidentiality and anonymity of the respondents were maintained throughout the study. No personally identifiable information was collected, and all responses were used solely for academic and research purposes. The data gathered were stored securely and were accessible only to the researcher. To protect the privacy of the participants, the findings were reported in summarized form, ensuring that individual responses could not be traced to any specific respondent. The researcher also ensured that the data were presented accurately and honestly, and that the welfare and rights of all participants were respected throughout the conduct of the study.

4. Results and Discussion

Table 3 presents the respondents' assessment of CodeChum based on the individual System Usability Scale (SUS) items. The highest mean scores were observed in indicating that students generally agreed that the platform was easy to use. Most of the remaining items received neutral ratings, suggesting that students had mixed opinions regarding different aspects of the platform's usability. Table 4 shows the overall System Usability Scale score of CodeChum. The platform obtained a mean SUS score of 49.7, which falls below the commonly accepted benchmark score of 68 and is interpreted as poor usability. This finding suggests that while students recognized some useful features of CodeChum, several usability concerns remain. Responses from the open-ended questions further revealed common issues related to strict output checking, case sensitivity, grading inconsistencies, and problematic test cases. These concerns may have contributed to the relatively low usability score reported by the respondents.

Table 3
Item-Level System Usability Results

Items	Mean	SD	Interpretation
SUS1	2.98	1.36	Neutral
SUS2	3.44	1.18	Agree
SUS3	3.42	1.14	Agree
SUS4	2.69	1.21	Neutral
SUS5	3.12	1.18	Neutral
SUS6	3.40	1.16	Neutral
SUS7	3.17	1.17	Neutral
SUS8	3.08	1.27	Neutral
SUS9	2.98	1.28	Neutral
SUS10	3.17	1.22	Neutral

Table 4
Overall System Usability Score

Measure	Value
Mean SUS Score	49.70
Median SUS Score	50.00
Interpretation	Poor Usability

Table 5 presents the respondents' assessment of CodeChum in terms of user experience. The overall mean score of 3.28 indicates a neutral perception of the platform. This suggests that while students recognized several useful features of CodeChum, they also encountered challenges that affected their overall experience in using the platform. Among the indicators, students agreed that CodeChum is easy to navigate, visually organized, and allows them to locate tools and features efficiently. However, neutral ratings were observed in areas related to platform responsiveness, coding environment, automated feedback, error messages, comfort of use, gamification, and overall experience. These findings imply that although CodeChum offers a functional environment for programming activities, improvements in system performance, feedback mechanisms, and user interaction may help provide a more positive user experience.

Table 5
User Experience

Items	Mean	SD	Interpretation
UE1	3.54	1.18	Agree
UE2	3.58	1.27	Agree
UE3	3.42	1.21	Agree
UE4	3.25	1.22	Neutral
UE5	3.21	1.33	Neutral
UE6	3.17	1.26	Neutral
UE7	3.25	1.28	Neutral
UE8	3.06	1.30	Neutral
UE9	3.23	1.21	Neutral
UE10	3.10	1.32	Neutral
Mean	**3.28**	**1.26**	**Neutral**

Table 6 presents the respondents' assessment of CodeChum in terms of learning support. The overall mean score of 3.21 indicates a neutral perception of the learning support provided by the platform. This suggests that students viewed CodeChum as moderately helpful in supporting their understanding of programming concepts and reinforcing programming skills. Although the respondents recognized the value of the programming exercises and problem-based activities, they expressed neutral perceptions regarding the platform's overall contribution to learning. These findings imply that while CodeChum provides useful opportunities for practice and skill reinforcement, improvements in instructional guidance, feedback quality, and learning resources may further strengthen its role as a learning support tool.

Table 6
Learning Experience in terms of Learning Support

Items	Mean	SD	Interpretation
LELS1	3.17	1.20	Neutral
LELS2	3.35	1.23	Neutral
LELS3	3.08	1.22	Neutral
LELS4	3.08	1.25	Neutral
LELS5	3.35	1.17	Neutral
Mean	**3.21**	**1.21**	**Neutral**

Table 7, presents the respondents' assessment of CodeChum in terms of engagement and motivation. The overall mean score of 3.10 indicates a neutral perception of the platform's ability to engage and motivate students in programming activities. This suggests that while CodeChum provided opportunities for participation and exploration, its overall influence on students' motivation toward learning programming was moderate. The findings show that students were somewhat encouraged to participate in programming activities and explore different approaches to solving problems. However, the neutral ratings across all indicators suggest that the platform's current features may not be sufficient to strongly motivate learners or significantly increase their interest in programming. Additional interactive and motivational features may help improve students' engagement with the platform.

Table 8 presents the respondents' assessment of CodeChum in terms of skill development. The overall mean score of 3.19 indicates a neutral perception of the platform's contribution to developing programming-related skills.

This suggests that students viewed CodeChum as somewhat helpful in improving their skills, although its impact was not strongly evident across all indicators. Among the indicators, students rated the platform more positively in terms of improving problem-solving abilities and strengthening logical thinking skills. However, neutral ratings across all items suggest that respondents were divided in their views regarding the extent to which CodeChum enhanced their confidence, independence, and overall development in programming. These findings imply that while the platform provides opportunities for practice and skill application, additional learning support and enhanced features may further strengthen its contribution to students' programming skill development.

Table 7
Learning Experience in terms of Engagement and Motivation

Items	Mean	SD	Interpretation
LEEM1	2.92	1.13	Neutral
LEEM2	3.13	1.09	Neutral
LEEM3	3.15	1.19	Neutral
LEEM4	3.19	1.17	Neutral
LEEM5	3.12	1.17	Neutral
Mean	3.10	1.15	Neutral

Table 8
Learning Experience in terms of Skills Development

Items	Mean	SD	Interpretation
LESD1	3.31	1.18	Neutral
LESD2	3.27	1.09	Neutral
LESD3	3.10	1.19	Neutral
LESD4	3.15	1.24	Neutral
LESD5	3.13	1.17	Neutral
Mean	**3.19**	**1.17**	**Neutral**

Table 9, presents the respondents' assessment of CodeChum in terms of user satisfaction. The overall mean score of 3.09 indicates a neutral level of satisfaction with the platform. This suggests that students generally had mixed views regarding their overall experience with CodeChum as a programming learning and assessment platform. Although some respondents recognized the platform's usefulness in supporting programming activities and enhancing their learning experience, their ratings remained neither strongly positive nor negative. The findings imply that while CodeChum provides features that assist students in learning programming, concerns related to feedback quality, assessment processes, and platform functionality may have influenced their level of satisfaction. Overall, the results suggest that CodeChum has potential as a formative assessment platform, but improvements in its features and user support mechanisms may help increase student satisfaction.

Table 9
User Experience

Items	Mean	SD	Interpretation
US1	3.10	1.16	Neutral
US2	3.12	1.22	Neutral
US3	3.13	1.21	Neutral
US4	3.12	1.08	Neutral
US5	3.00	1.24	Neutral
US6	3.15	1.11	Neutral
US7	3.15	1.19	Neutral
US8	2.92	1.33	Neutral
Mean	**3.09**	**1.19**	**Neutral**

Thematic Analysis of Open-Ended Responses - The responses to the open-ended questions were analyzed and grouped into common themes based on the students' experiences with CodeChum. The findings revealed three major themes: perceived strengths of the platform, challenges encountered during use, and suggested improvements.

Strengths of CodeChum - Many respondents viewed CodeChum as a useful learning platform that supports programming education. Students frequently mentioned its accessibility, organized lessons, immediate feedback, coding practice opportunities, and gamified features. Several respondents appreciated the platform's ability to

identify errors, provide debugging assistance, and allow them to apply programming concepts through hands-on activities. Others highlighted the convenience of having lessons, activities, and coding exercises available in a single environment. Overall, these responses suggest that CodeChum contributes positively to students' learning experiences by providing opportunities for practice, feedback, and skill development.

Challenges Encountered While Using CodeChum - Despite the positive feedback, respondents reported several issues while using the platform. The most frequently mentioned concern involved the strict checking mechanism, particularly regarding spacing, formatting, and case sensitivity. Many students noted that their solutions were marked incorrect even when the output was logically correct. Others reported encountering buggy test cases, grading inconsistencies, system errors, and activities with abrupt difficulty increases. Several respondents also expressed frustration with the lack of notifications and the complexity of some programming tasks. These concerns reflect the issues identified in the quantitative findings related to system usability, user experience, and user satisfaction.

Suggested Improvements - Students offered several recommendations for improving CodeChum. The most common suggestions included improving the grading and output-checking mechanism, reducing excessive sensitivity to formatting requirements, and providing more detailed feedback when errors occur. Respondents also recommended adding hints, notifications for new activities, better error explanations, improved test-case validation, and enhanced system stability. Some students suggested developing a mobile application, incorporating more examples and learning materials, and implementing features that allow users to communicate concerns directly through the platform. These recommendations indicate that while students recognize the benefits of CodeChum, they believe further improvements could enhance its effectiveness as a programming learning and formative assessment platform.

Educational Implications of the Study - The results of this study highlight the importance of regularly evaluating educational technologies used in programming courses. Although students acknowledged the benefits of CodeChum in providing coding practice, immediate feedback, and accessible learning resources, the overall findings indicate that there is still room for improvement in terms of usability, learning support, engagement, and user satisfaction. These results suggest that simply providing a digital platform is not enough; students also need clear feedback, reliable assessment mechanisms, and a learning environment that supports their needs throughout the learning process. The study also emphasizes the value of considering students' experiences when adopting technology in programming education. The concerns raised regarding grading accuracy, output checking, and system functionality show how platform design can influence students' learning experiences. For educators and administrators, the findings can serve as a basis for improving the implementation of CodeChum and similar learning technologies. For developers, the results provide practical areas for enhancement that may contribute to a more effective and student-centered programming learning environment.

5. Conclusion

This preliminary study evaluated CodeChum as a formative assessment platform based on the perceptions of second-year Computer Science students who have used the platform since its adoption in the School of Computing. The findings revealed that CodeChum obtained a System Usability Scale (SUS) score that falls under the poor usability category, indicating that students experienced several usability concerns while using the platform. Similarly, user experience, learning support, engagement and motivation, skill development, and user satisfaction were all rated at a neutral level, suggesting that students recognized both the strengths and limitations of the platform. Despite the neutral ratings, the qualitative responses showed that students appreciated CodeChum's accessibility, organized learning materials, automated feedback, gamified features, and opportunities for hands-on programming practice. However, concerns regarding strict output checking, case sensitivity, grading inconsistencies, buggy test cases, and limited feedback were repeatedly raised by the respondents. These findings suggest that CodeChum remains a useful tool for supporting programming instruction, but several aspects of the platform require improvement to enhance its usability, learning support, and overall user experience. Overall, the

study highlights the importance of continuously evaluating educational technologies from the students' perspective. As the first cohort to use CodeChum across multiple programming courses, the respondents provided valuable insights that may serve as baseline information for improving the platform and strengthening its role as a formative assessment tool in programming education.

Recommendations for future study - Based on the findings of the study, the following recommendations are proposed:

- The developers and administrators of CodeChum may consider improving the platform's usability by addressing concerns related to grading consistency, output checking, case sensitivity, and test-case validation. Enhancements in these areas may help provide a better user experience for students.
- Programming instructors may maximize the use of CodeChum's learning and assessment features by providing additional guidance on interpreting feedback, debugging errors, and completing programming activities. This may help students utilize the platform more effectively.
- Additional support features, such as clearer error explanations, activity notifications, hints, and improved feedback mechanisms, may be incorporated to strengthen student engagement, motivation, and satisfaction while using the platform.
- The School of Computing may use the findings as baseline information in evaluating the continued implementation of CodeChum in programming courses and in identifying areas for improvement that can enhance student learning experiences.
- Future researchers may conduct similar studies involving students from other academic programs, year levels, or institutions to determine whether the findings are consistent across different populations.
- Future research may also employ mixed-methods or longitudinal approaches to obtain a deeper understanding of students' experiences with CodeChum and its long-term impact on programming education.

These recommendations may contribute to the continuous improvement of CodeChum as a formative assessment platform and support efforts to enhance programming instruction and learning experiences among students.

AI Use Disclosure - Microsoft Copilot (GPT-5 chat model, 2026) was used for language editing and manuscript refinement. All outputs were reviewed and revised by the author, who takes full responsibility for the content of the manuscript. No confidential or personally identifiable data were entered into the AI tool.

6. References:

Abu Bakar, E. E., Abd Halim, N. D., Abdul Hanid, M. F., & Inderawati, R. (2025). The challenges of teaching and learning programming in schools: Insights from a systematic literature review. *Karya Journal of Emerging Technologies in Human Services, 1*(1), 48–63. https://doi.org/10.37934/kjeths.1.1.4863

Ahmad, S. N., & Ghazali, J. (2020). *Programming teaching and learning: Issues and challenges*. In *Proceedings of the 6th International Conference on Information Technology & Society (ICITS 2020)* (pp. 1–10). Kolej Universiti Islam Antarabangsa Selangor.

Alakeel, A. M. (2015). Investigating difficulties of learning computer programming in Saudi Arabia. *Universal Journal of Educational Research, 3*(9), 567–577. https://doi.org/10.13189/ujer.2015.030901

Araujo, L., Lopez-Ostenero, F., Plaza, L., & Martinez-Romo, J. (2025). Automated formative feedback for algorithm and data structure self-assessment. *Electronics, 14*(5), 1034. https://doi.org/10.3390/electronics14051034

Arellano, C., & Serato, J. V. (2023). *Profile of BSCS students and their perceptions in the use of CodeChum as an online assessment tool*.

Brooke, J. (1996). SUS: A quick and dirty usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. A. Weerdmeester (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.

Correia, A., & Good, K. (2023). Exploring the relationships between student perceptions and educational technology utilization in higher education. *Journal of Educational Technology Development and Exchange, 16*(1), 92–107. https://doi.org/10.18785/jetde.1601.05

Fanchamps, N., & Van der Wijk, M. (2023). The influence of technology-enhanced programming environments and the impact of teacher guidance and interventions on computational thinking development. In *INTED2023 Proceedings: 17th International Technology, Education and Development Conference* (pp. 2039–2049). IATED. https://doi.org/10.21125/inted.2023.0568

Gallucci, M. (2020). *GAMLj: General analyses for linear models using jamovi* [jamovi module]. https://gamlj.github.io

Irinco, R., & Irinco-Salinas, R. (2025). Student perception of learning, engagement, and confidence in research skills using mobile devices as an active learning strategy in a science course. *International Journal on Open and Distance E-Learning, 10*(2). https://doi.org/10.58887/ijodel.v10i2.291

Islam, N., Sheikh, G., Fatima, R., & Alvi, F. (2019). A study of difficulties of students in learning programming. *Journal of Education & Social Sciences, 7*, 38–46. https://doi.org/10.20547/jess0721907203

Ivanović, M., Xinogalos, S., Pitner, T., & Savić, M. (2017). Technology enhanced learning in programming courses: International perspective. *Education and Information Technologies, 22*, 2981–3003.

Jha, M., Shivshankar, S., Thakur, S., & Jha, S. (2023). Changes formative assessments for programming units. *Universal Journal of Educational Research, 11*(8), 148–158. https://doi.org/10.13189/ujer.2023.110802

Jiang, Y., Zhao, Z., Wang, L., & Hu, S. (2020). Research on the influence of technology-enhanced interactive strategies on programming learning. In *2020 15th International Conference on Computer Science & Education (ICCSE)* (pp. 693–697). IEEE. https://doi.org/10.1109/ICCSE49874.2020.9201627

Kalelioğlu, F., & Keskinkılıç, F. (2026). Difficulties of learning programming in higher education: A systematic literature review of recent research. *Ankara University Journal of Faculty of Educational Sciences (JFES), 59*(2), 771–835. https://doi.org/10.30964/auebfd.1703365

Khanlari, A. (n.d.). *Students perception of the effects of educational technology on their learning and characteristics.*

Li, H.-J., Huang, Q.-R., Wen, L.-P., Chen, W., & Xu, Z.-Z. (2025). Generative artificial intelligence supported programming learning: Learning effectiveness and core competence. *SAGE Open, 15*(3). https://doi.org/10.1177/21582440251377986

Liebenberg, J. (2023). *Automatic formative assessment of programming tasks*. In *The Paris Conference on Education 2023 Official Conference Proceedings*.

Manorat, P., Tuarob, S., & Pongpaichet, S. (2025). Artificial intelligence in computer programming education: A systematic literature review. *Computers and Education: Artificial Intelligence, 8*, Article 100403. https://doi.org/10.1016/j.caeai.2025.100403

Maranga, J. J. A., Matugas, L. K. J., Lim, J. F. W., & Sta. Romana, C. L. C. (2019). CodeChum: An online learning and monitoring platform for C programming. In P. Kommers, T. Issa, P. Isaías, & W. Hui (Eds.), *Proceedings of the IADIS International Conference Educational Technologies 2019* (pp. 109–118). IADIS Press.

Maranga, X. D., Pausanos, M. V., Sinoy, S., Sta. Romana, C., & Arellano, C. (2024). Integrating adaptive assessment in CodeChum for personalized programming education. In *Proceedings of the 2024 10th International Conference on Education and Training Technologies (ICETT '24)* (pp. 89–95). Association for Computing Machinery. https://doi.org/10.1145/3661904.3661910

Minga, C., & Ghosh, S. (2024). Student perceptions on ICT use in teaching and learning in public secondary schools in Mbeya District, Tanzania. *Journal of Education, Society and Behavioural Science, 37*(6), 26–39. https://doi.org/10.9734/jesbs/2024/v37i61325

Monserate, C. A. (2018). Impact of technology on the academic performance of students and teaching effectiveness. *International Journal of Interdisciplinary Research and Innovations, 6*(1), 47–87.

Mposula, F., & Oyetade, K. (2025). Enhancing programming education through game-based learning for

improved engagement and learning outcomes. *Educational Process: International Journal, 19*, Article e2025558. https://doi.org/10.22521/edupij.2025.19.558

Munisamy, M., Md Osman, S. Z., & Sanmugam, M. (2024). Exploring the landscape of online formative assessment practices in programming courses: A scoping review. In *Integrating cutting-edge technology into the classroom* (pp. 1–26). IGI Global. https://doi.org/10.4018/979-8-3693-3124-8.ch005

Napalit, F., Tanyag, B., So, C. L., Sy, C., & San Pedro, J. R. (2023). Examining student experiences: Challenges and perception in computer programming. *International Journal of Research Studies in Education, 12*(8), 101–112. https://doi.org/10.5861/ijrse.2023.71

Ou, Q., Liang, W., He, Z., Liu, X., Yang, R., & Wu, X. (2023). Investigation and analysis of the current situation of programming education in primary and secondary schools. *Heliyon, 9*(4), Article e15530. https://doi.org/10.1016/j.heliyon.2023.e15530

Pan, Y., & Harun, J. (2025). Conquering coding fears: A systematic review of programming anxiety in higher education. *Journal of Information Technology Education: Research, 24*, Article 20. https://doi.org/10.28945/5543

Ragab, K., Martínez-Jiménez, E., & Fernandez-Ahumada, E. (2025). Students' perceptions of the impact of interactive technology on engagement in STEM classes. *Contemporary Educational Technology, 17*(4), Article ep602. https://doi.org/10.30935/cedtech/17408

Silva, L., Mendes, A. J., & Gomes, A. (2020). Computer-supported collaborative learning in programming education: A systematic literature review. In *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1086–1095). IEEE. https://doi.org/10.1109/EDUCON45650.2020.9125237

Singh, S. (2024). *Addressing challenges in computer programming education: An action research approach for effective student support* [Doctoral dissertation, Northeastern University]. Northeastern University Digital Repository. https://repository.library.northeastern.edu/files/neu:ms397g413/fulltext.pdf

Slimi, Z., Benayoune, A., & Alemu, A. E. (2025). Students' perceptions of artificial intelligence integration in higher education. *European Journal of Educational Research, 14*(2), 471–484. https://doi.org/10.12973/eu-jer.14.2.471

Stanja, J., Gritz, W., & Krugel, J. (2023). Formative assessment strategies for students' conceptions: The potential of learning analytics. *British Journal of Educational Technology, 54*(1), 58–74. https://doi.org/10.1111/bjet.13288

Thangaraj, J. (2022). Formative assessment as a learning method for introductory programming. In *Proceedings of the 2022 Conference on United Kingdom & Ireland Computing Education Research* (UKICER '22, Article 22). Association for Computing Machinery. https://doi.org/10.1145/3555009.3555033

Thangaraj, J., Ward, M., & O'Riordan, F. (2024). The impact of using formative assessment in introductory programming on teaching and learning. In *Proceedings of the 10th International Conference on Higher Education Advances (HEAd'24)*.

Thangaraj, J., Ward, M., & O'Riordan, F. (2025). Exploring the holistic impact of adaptive formative assessment for novice programmers. In *Proceedings of the 33rd International Conference on Computers in Education (ICCE 2025)*. https://doi.org/10.58459/icce.2025.6102

The jamovi project. (2024). *jamovi (Version 2.7* https://www.jamovi.org

Tlale-Mkhize, M. (2025). Challenges experienced by first-year students in the programming module: A literature review. *Journal of Education and Learning Technology, 6*(12), 1555–1573. https://doi.org/10.38159/jelt.202561224

Ubaidillah, N. Z., Baharuddin, N. N., Kasil, N., & Ismail, F. (2020). Students' perception of the use of technology in education. *Environment-Behaviour Proceedings Journal, 5*(15), 117–122. https://doi.org/10.21834/ebpj.v5i15.2374

Wan, H. (2025). Role of online assessment system in formative evaluation of programming education. *Computers and Education: Artificial Intelligence, 9*, 100515. https://doi.org/10.1016/j.caeai.2025.100515

Wang, J., Lin, P., Tang, Z., & Chen, S. (2023). How problem difficulty and order influence programming education outcomes in online judge systems. *Heliyon, 9*(11), Article e20947.

https://doi.org/10.1016/j.heliyon.2023.e20947