IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, May 2026
Impact Factor: 8.2

First-year students' Perception of Different Assessment Styles and Programming Performance in Python: A Descriptive-Correlational Study

Rain A. Solante[1], Joshua A. Oliveros[2], Jessica Rose E. Fernandez[3]
BSCS Student, College of Computing and Information Sciences
Surigao del Norte State University, SurigaoCity, Philippines[1,2]
Faculty, College of Computing and Information Sciences
Surigao del Norte State University, Surigao City, Philippines[3]
rainsolante1@gmail.comoliverosjoshua55@gmail.com jfernandez@ssct.edu.ph

Abstract: *This study examined the perceptions of first-year Bachelor of Computer Science (BSCS) students on different assessment styles and their relationship with Python programming performance. Using a descriptive-correlational design, 41 first-year BSCS students from Surigao Del Norte State University were selected through purposive sampling. Data were gathered through an online structured questionnaire with a 4-point Likert scale measuring students' perceptions of three assessment styles: quiz-based, written exam-based, and laboratory exam-based assessments. Mean and standard deviation were used to analyze the data. The findings revealed that students generally held positive perceptions across all three assessment styles, with responses ranging from "Agree" to "Strongly Agree." Laboratory examinations received the highest level of agreement, followed by quizzes and written examinations. Students showed slightly lower confidence in complex topics such as conditional statements, loops, comparison operators, and parameters. The results suggest that practical and interactive assessments, particularly laboratory exams and quizzes, contribute more to students' perceived programming competence. These findings highlight the importance of incorporating varied assessment strategies that balance theoretical and practical skills in introductory Python programming courses.*

Keywords: BSCS students, assessment styles, programming performance, Python programming, descriptive-correlational study, first-year college students

I. INTRODUCTION

Programming performance has become one of the most critical components of computing degree programs as the demand for skilled technology professionals continues to rise globally. Among the various programming languages taught in higher education, Python has emerged as one of the most widely adopted introductory languages due to its simplicity and versatility (Ranjeeth & Padayachee, 2024). Central to student learning is the role of assessment, the methods used to evaluate student understanding can directly shape how students' study, engage, and perform. Beyond assessment design, how students perceive these methods matters just as much, since attitudes toward assessments can influence motivation and effort. This study seeks to describe first-year Bachelor of Computer Science students' perceptions of different assessment styles and examine their relationship with Python programming performance.

Unger and Lecher (2024) found that allowing students to choose their preferred assessment type increased engagement and improved how they viewed their instructor's teaching. Similarly, Ye et al. (2022) emphasized that students' perceptions of learning are significant predictors of their motivation and academic performance in computing education. Charytanowicz et al. (2024) also found notable differences in student scores depending on the teaching modality used. A study on first-year CS students further found that students preferred practical and written

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

examinations as their most effective evaluation methods (Perceived Learning Behaviors Study, 2024). Additionally, variations in assessment perceptions were observed based on prior programming experience, gender, and technology familiarity (Assessing Engineering Student Perceptions, 2024). Most of these studies were conducted abroad and their findings may not fully reflect the situation of students in the Philippines, making this study relevant and necessary.

Despite the growing body of research on programming education, little attention has been given to how first-year BSCS students in the Philippines perceive different assessment styles and how these relate to their Python programming performance. If left unaddressed, educators may continue designing assessments without fully understanding how their students experience them, which could result in poor engagement and weaker performance. This study aims to bridge that gap by looking into this concern within a single local institution.

To address this, the study used a descriptive-correlational design through survey questionnaires. The findings are intended to help teachers, program coordinators, and curriculum designers make better decisions in structuring assessments in an introductory Python programming class.

Objectives of the Study

This study aims to:

1. Determine the profile of first-year BSCS students in terms of their perceived assessment styles in Python programming.
2. Assess the Python programming performance of first-year BSCS students.
3. Examine the relationship between students' perception of different assessment styles and their Python programming performance.

Conceptual Framework

This section outlines the conceptual framework of the study and presents the impact of differences of assessment style on programming performance. This schematic diagram uses the independent variable anddependent variable of the study.

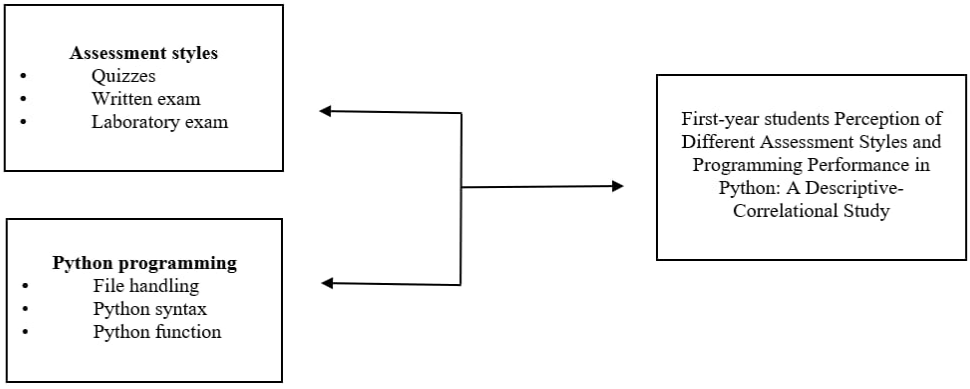

Figure 1. Conceptual Framework

II. REVIEW OF RELATED LITERATURE

Assessment Styles

The assessment style in Python programming for students has gained significant attention from past proponents. Previous studies focused on issues such as teacher education, family background, and socio-economic factors. As emphasized by Shakeel and Peterson (2020), one of the most significant parts of human resource development is education. In Python programming, assessment methods including quizzes, written exams, and lab exams serve as the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

basis of student achievement. The match between teaching methods and student learning style encourages meaningful learning and academic performance. Student learning style, teaching practices, and performance assessment form an educational triangle that fosters excellence (Maya et al., 2021). Quizzes are widely used as formative assessments in programming courses. Ghosh et al. (2024) found that quizzes involving code tracing, debugging, and equivalence tasks improved learning outcomes among 405 elementary students learning visual programming. Hui et al. (2025) reported that unlimited randomized self-assessment quizzes in introductory computer science courses enhanced student study habits and self-efficacy. These findings suggest that well-designed quizzes support programming performance by reinforcing concepts through repeated practice.Written exams measure students' theoretical understanding of programming concepts. According to Otero (2024), multiple-choice questions generated for Python assessments can effectively evaluate knowledge of syntax and logic. Shakeel and Peterson (2020) emphasized that written tests remain a standard measure of academic achievement. However, Maya et al. (2021) noted that written exams should align with teaching methods and learning styles to maximize their effectiveness. In programming education, written exams assess conceptual knowledge but may not capture practical coding skills. Laboratory examinations assess students' hands-on programming abilities. Ibrahim et al. (2025) found that students using a quiz-based mobile app for programming scored significantly higher than non-users, highlighting the value of interactive assessments. Nielsen et al. (2025) reported that practice-based assessments improved practical performance in computing courses. Laboratory exams allow students to demonstrate competencies in areas such as file handling, functions, and syntax application, bridging the gap between theory and practice.

Python Programming Performance

Learning programming syntax is particularly challenging for novice programmers. Tchamabe et al. (2024) introduced PyLe, an interactive learning environment for introductory Python courses. Their study with first-year students found that PyLe significantly improved students' ability to master Python syntax with high usability rates. However, they found no significant relationship between time spent on tasks and solution quality. Xie et al. (2025) conducted a randomized controlled trial examining syntax exercises in Python courses. Controlling for prior CS experience, they found no significant differences between students assigned to syntax exercises versus those assigned to practice programming problems across multiple performance measures. Sunderraman (2025) presented an approach requiring students to complete programming assignments using a purely functional subset of Python. This method engages learners with functional concepts including immutability, pure functions, and stateless programming. The proponents demonstrated that functional concepts can be effectively taught in introductory Python courses. The LuCE Research Lab (2025) developed the Toolbox of Functions approach for teaching code reuse to beginner Python programmers. Over 800 users executed more than 30,000 programs using this approach, effectively instilling code reuse principles among high school students learning Python. Zaripov and Hasanov (2023) highlighted that files serve as collections of data in Python, allowing storage of objects including data, functions, classes, and modules. Their study covered opening, modifying, editing, deleting, and closing files as essential Python operations. The University of Kansas (2025) outlined practical file input/output operations, emphasizing built-in functions for reading and writing files using modes such as "r" and "w". Middlebury College (2024) demonstrated practical applications including loading text from files, using context managers, and reading multiple lines into lists for processing.

Previous studies have examined assessment methods and programming performance in computer programming courses, but most focused only on specific approaches such as feedback systems, peer assessment, or automated evaluation rather than students' perceptions of different assessment styles. Çakıroğlu et al. (2020) found that feedback strategies improved students' programming performance, while Hsiao et al. (2023) emphasized the effectiveness of peer assessment in enhancing computational thinking and learning outcomes in Python programming. However, these studies did not compare multiple assessment styles such as quizzes, written examinations, and computer laboratory examinations, nor did they focus on the perceptions of first-year BSCS students using a descriptive-correlational

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

design. Furthermore, limited studies have been conducted within the Philippine context, creating a gap in understanding how different assessment styles relate to students' programming performance in Python.

III. METHODOLOGY

Research Design

A descriptive-correlational study is designed to describe relationships among variables, instead of attempting to determine cause and effect relationships (Lappe, 2020). This study examined the relationship between BSCS first-year students' perceptions of assessment styles and their programming performance in Python. A descriptive-correlational approach was appropriate because the study aimed to determine whether a significant relationship exists between students' perceived assessment styles and their programming performance, without manipulating any variables.

Participants and sampling method

The participants of the study were BSCS first-year students from the Surigao Del Norte State University. The population of first-year students who enrolled in the BSCS program consisted of 93 students to complete the sample. The sample size was 41 that were computed using the 50% rule of thumb. They were chosen as the respondents of this study because they were newly exposed to Python programming making them the most appropriate to examine in this study. Purposive sampling was used in population size, where each participant was selected based upon a variety of criteria which may include significant knowledge of the research issue toparticipate in the research (Rai et al., 2015). The inclusion criteria require that participants:

1. Were enrolled as first-year BSCS students at Surigao Del Norte State University;
2. Had taken a Python programming subject;
3. Had experienced different assessment styles (quizzes, written exam, lab exam).

Data collection

Data were collected using an online structured questionnaire distributed to the participants through Google forms. The use of Google forms enabled convenient distribution and collection of responses, as participants could answer the survey in their own time. It also ensured automatic recording and organization of responses in a spreadsheet format, minimizing data entry errors (Simanjuntak & Limbong, 2018).

The questionnaire included measuring students' perception of different assessment styles such as quizzes, written exam, and laboratory exam. Responses were measured using a Likert scale allowing for efficient data collection and analysis.

The responses of the students were picked through these checklist numbers with a 4-point Likert scale: (4) Strongly Agree, (3) Agree, (2) Disagree, (1) Strongly Disagree. This was done to provide more decisive data for analyzing students' perceptions of assessment styles.

TABLE I: THE STUDENTS' PERCEPTIONS OF QUIZ-BASED ASSESSMENT AND ITS INFLUENCE IN PYTHON PROGRAMMING PERFORMANCE.

After taking the quiz, I can…	1	2	3	4
1) determine the default value for reading in file handling.				
2) determine a file for appending in file handling.				
3) determine a file for writing in file handling.				
4) identify an 'f string' in Python syntax.				
5) identify if, elif, and else in operations.				
6) identify which is the comparison operators used in conditional statements.				
7) examine functions defined with the keyword "def".				
8) locate the "return" statements in code.				
9) identify the functions used in a code.				

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

In this section, the respondents were asked to select among the 4-point Likert scale by picking the following number: (4) Strongly Agree, (3) Agree, (2) Disagree, and (1) Strongly Disagree in the table. This was done to know the perception of Quiz-Based assessment and its correlation in Python programming performance.

TABLE II: THE STUDENTS' PERCEPTIONS OF WRITTEN EXAM-BASED ASSESSMENT AND ITS CORRELATION IN PYTHON PROGRAMMING PERFORMANCE.

After taking the written exam, I can…	1	2	3	4
1) explain how to use 'r' (read).				
2) explain how to use 'a' (append).				
3) describe the 'w' (write) in handling a file.				
4) recall the correct syntax for writing Python statements.				
5) apply conditional statements and while loop and for loop correctly.				
6) distinguish the data store in a variable if it's a str or an int.				
7) explain the proper use of parenthesis () in a determining a function.				
8) identify the parameters inside a code in Python.				
9) Analyse function codes to insight the output correctly in Python.				

Furthermore, another Likert scale was done to examine the perception of Written Exam-Based assessment and its relation in Python programming performance of students.

TABLE III: THE STUDENTS' PERCEPTIONS OF LABORATORY EXAM-BASED ASSESSMENT AND ITS CORRELATION IN PYTHON PROGRAMMING PERFORMANCE.

After taking the laboratory exam, I can…	1	2	3	4
1) apply 'r' (read) to access a file without editing it.				
2) create a new content to the existing file using 'a' (append).				
3) replace the new existing file using 'w' (write).				
4) use comments to label a specific line using hashtag (#).				
5) use triple single quotes ("') to comment in a 2 or more lines.				
6) demonstrate the proper calculation using the Python operators. Such as arithmetic (*, /, +, % and others) and comparison (==, =! <, > and others).				
7) establish a function for not repeatedly write the same calculation code with def ().				
8) apply where to place the "return" in the functions.				
9) use the "return" statements to send back data in the functions.				

Lastly, the ethical guidelines for research consideration were followed in this study. The BSCS first-year students participated in the study were fully informed how their information is used, including any potential risks or benefits. Their answer remained private and confidential unless they explicitly agreed to share them with others. The personal information of participants was protected, ensuring that process is not linked to individual identities unless they gave permission to do so. A clear and straightforward questionnaire related to the different assessment styles and their Python programming performance was provided to minimize confusion. The process was conducted in a respectful and supportive manner, prioritizing psychological well-being of all the participants.

Data Analysis

The proponents addressed the research questions of the study through the use of descriptive analysis specifically mean and standard deviation. Mean is the average of collected data, it was used to determine the overall perception of

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, May 2026
Impact Factor: 8.2

students toward assessment styles, as it provided average response based on the Likert scale, while standard deviation measured the consistency of students' responses. All the analysis were conducted using Spreadsheet Software (Excel). The table show below was used to interpret the level of agreement and the consistency of different assessment styles (Quizzes, Written-Exam, and Laboratory-Exam) and their programming performance in Python.

TABLE IV: SCALE INTERVAL FOR THE QUIZ-BASED ASSESSMENT STYLE AND ITS RELATION ON PROGRAMMING PERFORMANCE IN PYTHON.

Scale Interval	Interpretation
3.26 – 4.00	Strongly Agree
2.51 – 3.25	Agree
1.76 – 2.50	Disagree
1.00 – 1.75	Strongly Disagree

TABLE V: SCALE INTERVAL FOR THE WRITTEN EXAM-BASED ASSESSMENT STYLE AND ITS RELATION ON PROGRAMMING PERFORMANCE IN PYTHON.

Scale Interval	Interpretation
3.26 – 4.00	Strongly Agree
2.51 – 3.25	Agree
1.76 – 2.50	Disagree
1.00 – 1.75	Strongly Disagree

TABLE VI: SCALE INTERVAL FOR THE LABORATORY EXAM-BASED ASSESSMENT STYLE AND ITS RELATION ON PROGRAMMING PERFORMANCE IN PYTHON.

Scale Interval	Interpretation
3.26 – 4.00	Strongly Agree
2.51 – 3.25	Agree
1.76 – 2.50	Disagree
1.00 – 1.75	Strongly Disagree

The scale interval was used to interpret the level of students' agreement on different assessment styles.

IV. RESULTS AND DISCUSSION

This part of the research focused on providing in-depth analysis of the findings, interpretation of the data, and the results of the study. The study utilized a descriptive-correlational research design to examine the perceptions of BSCS first-year students on different assessment styles and the relationship between these perceptions in their programming performance in Python. The proponents used Microsoft Excel to get the mean and standard deviation for the gathered data, with the help of this software, proponents greatly managed their time effectively.

Quiz-Based

Table 7 aimed to analyze and categorize programming performance in Python based on Quiz-Based assessment style, using mean scores and standard deviation to determine the level of agreement based on the assessment style.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, May 2026
Impact Factor: 8.2

TABLE VII: QUIZ-BASED PERCEPTIONS AMONG BSCS FIRST-YEAR STUDENTS.

Statement	Mean	Standard Deviation	Interpretation
After taking the quiz, I can…			
1) determine the default value for reading in file handling.	3.3	0.57	Strongly agree
2) determine a file for appending in file handling.	3.2	0.56	Agree
3) determine a file for writing in file handling.	3.2	0.53	Agree
4) identify an 'f string' in Python syntax.	3.3	0.55	Strongly Agree
5) identify if, elif, and else in operations.	3.3	0.53	Strongly Agree
6) identify which is the comparison operators used in conditional statements.	3.0	0.64	Agree
7) examine functions defined with the keyword “def”.	3.4	0.59	Strongly Agree
8) locate the “return” statements in code.	3.1	0.65	Agree
9) identify the functions used in a code.	3.3	0.57	Strongly Agree

The section presents the Quiz-Based Perception among BSCS first-year students. The ‘After taking quiz, I can examine functions defined with keyword def’ has the highest mean of 3.4, this indicated that studentsStrongly Agree with this statement, while the standard deviation of 0.59 showed that responses are consistent. Despite the differences of their standard deviation with of 0.57, 0.55, and 0.53 it still indicates responses are consistent, the statement ‘1, 4, 5, 9’ comes in second with ranging similar mean average of 3.3 that indicates Strongly Agree with their perceptions. Moreover, a 0.64 standard deviation suggested the consistency in answer, while the lowest mean of 3.0 for ‘Identify which is the comparison operators used in conditional statements.’. Lastly, all things are considered, the overall "Agree" to "Strongly Agree" interpretations, paired with relatively low standard deviations (0.53 to 0.65), suggest that the quiz successfully reinforced a consistent and positive grasp of fundamental file handling, syntax, and logic concepts across the cohort.

Written Exam-Based

Table 8 presented the examination of programming performance in Python based on written exam assessment style. This was also categorized by Strongly Agree, Agree, Disagree, Strongly Disagree.

TABLE VIII: WRITTEN EXAM-BASED PERCEPTIONS AMONG BSCS FIRST-YEAR STUDENTS.

Statement	Mean	Standard Deviation	Interpretation
After taking the written exam, I can…			
1) explain how to use 'r' (read).	3.2	0.56	Agree
2) explain how to use 'a' (append).	3.2	0.68	Agree
3) describe the 'w' (write) in handling a file.	3.2	0.59	Agree
4) recall the correct syntax for writing Python statements.	3.3	0.58	Strongly Agree
5) apply conditional statements and while loop and for loop correctly.	3.1	0.63	Agree
6) distinguish the data store in a variable if it’s a str or an int.	3.3	0.57	Strongly Agree
7) explain the proper use of parenthesis () in a determining a function.	3.3	0.56	Strongly Agree
8) identify the parameters inside a code in Python.	3.1	0.70	Agree
9) analyze function codes to insight the output correctly in Python.	3.2	0.49	Strongly Agree

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

The table indicates that students generally have positive perceptions of written exam-based assessment, with responses ranging from "Agree" to "Strongly Agree." The highest mean of 3.3 is observed in recalling correct Python syntax, distinguishing data types, and explaining the proper use of parentheses in functions, showing that students are most confident in these fundamental concepts, while the standard deviations for these items around 0.56–0.58 are low, which means the students' responses are consistent and closely grouped, indicating similar levels of agreement. Overall, all standard deviation values in the table that ranging from 0.49 to 0.70 are low, suggesting that there is little variation in responses and that most students share similar perceptions. In contrast, the lowest mean score 3.1 appears in applying conditional statements and loops and identifying parameters in code; although still interpreted as "Agree," these slightly lower means suggest that students are less confident in these more complex skills, with slightly higher standard deviations (up to 0.70) indicating a bit more variation in their responses.

Laboratory Exam-Based

TABLE IX: LABORATORY EXAM-BASED PERCEPTIONS AMONG BSCS FIRST-YEAR STUDENTS.

Statement	Mean	Standard Deviation	Interpretation
After taking the laboratory exam, I can…			
1) apply 'r' (read) to access a file without editing it.	3.3	0.52	Strongly Agree
2) create a new content to the existing file using 'a' (append).	3.3	0.50	Strongly Agree
3) replace the new existing file using 'w' (write).	3.3	0.45	Strongly Agree
4) use comments to label a specific line using hashtag (#).	3.3	0.48	Strongly Agree
5) use triple single quotes (''') to comment in a 2 or more lines.	3.2	0.55	Agree
6) demonstrate the proper calculation using the Python operators. Such as arithmetic (*, /, +, % and others) and comparison (==, =, <,> and others).	3.1	0.59	Agree
7) establish a function for not repeatedly write the same calculation code with def ().	3.2	0.62	Agree
8) apply where to place the "return" in the functions.	3.2	0.59	Agree
9) use the "return" statements to send back data in the functions.	3.3	0.53	Strongly Agree

The table presents the laboratory exam-based perceptions of BSCS first-year students regarding Python programming concepts. The highest mean of 3.3 are observed in several items, indicating that students Strongly Agree that they can apply file handling functions such as reading ('r'), appending ('a'), writing ('w'), using comments (#), and returning values in functions after taking the laboratory exam, while the standard deviation ranging from 0.45-0.53 are considered that their responses are consistent. The second highest mean of 3.2 are seen in using triple quotes ('''), establishing not repeated functions, and applying where to place return indicating they are agree in their perceptions, while the standard deviation of 0.55, 0.59, and 0.62 are relatively consistent across all items. However, the demonstrations of proper Python calculation received the lowest mean of 3.1 with standard deviation of 0.59 that indicates agree and consistent throughout their perception. Overall, the results show that students generally have positive perceptions, as most of the statements fall under "Strongly Agree" and "Agree," with mean scores ranging from 3.1 to 3.3 and the responses are all consistent with a standard deviation ranging from 0.45-0.62 respectively.

Interpretation of the findings

Based on the results, students generally showed positive perceptions of the quiz-based assessment, with most responses falling between agree and Strongly Agree. This indicates that the quiz helped strengthen their understanding of important programming concepts such as functions, syntax, and logic. The responses were also fairly consistent across participants, suggesting similar learning experiences among the students. However, some areas, particularly

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, May 2026
Impact Factor: 8.2

those involving comparison operators, were less strongly perceived, implying that these concepts may not have been as fully understood. Overall, the findings suggest that the quiz was effective in reinforcing learning, though a few topics may still require further emphasis. According to Sanusi et al. (2025), students generally show positive perceptions toward quiz-based assessments, noting that such tools enhance understanding and support engagement in learning. This supports the findings of the present study, where students also agreed to Strongly Agree that quizzes helped reinforce their knowledge of programming concepts. However, while this study highlights overall positive learning outcomes and engagement, the present study differs by identifying specific areas, such as comparison operators, where students demonstrate relatively lower confidence. This indicates that although quiz-based assessments are effective and well-received, certain concepts may still require additional instructional emphasis. For table 8, Students generally showed positive perceptions of written exam-based assessment, expressing agreement to strong agreement that it supports their understanding of Python concepts. They appeared most confident in fundamental topics such as syntax, data types, and function structure, with consistent responses across the group. However, slightly lower confidence was observed in more complex areas like conditional statements, loops, and parameters, where responses showed a bit more variation. Overall, the results suggest that written exams are effective in reinforcing basic knowledge, though advanced skills may require additional support. According to Öqvist and Nouri (2018), different assessment methods, including written or hand-coded exams, help students develop understanding in programming, particularly in foundational concepts, but students often face more difficulty when dealing with complex problem-solving tasks. This supports the findings of the present study, where students showed strong confidence in basic areas such as syntax, data types, and function structure. However, both studies also reveal a similar challenge—students tend to have lower confidence and more varied responses when applying advanced concepts like conditional statements, loops, and parameters. Lastly, for table 9, Students generally showed positive perceptions of the laboratory exam-based assessment, with responses ranging from agree to Strongly Agree. They appeared most confident in applying practical programming skills such as file handling, using comments, and returning values in functions, with consistent responses across students. Slightly lower agreement was observed in areas like using triple quotes, structuring functions, and properly placing return statements, although these were still positively perceived. The lowest perception was seen in performing Python calculations, but responses remained consistent. Overall, the findings showed that laboratory exams contributed to students' hands-on understanding of programming, though some procedural and computational aspects may still need further reinforcement. According to Rubio-Sánchez et al. (2013), students generally have positive perceptions of programming assessment tools, as these help them test their understanding and improve their coding skills, although students also noted that feedback needs to be more detailed to fully support learning. This is similar to the present study, where students showed agreement to strong agreement that laboratory exams enhanced their ability to apply programming concepts such as file handling and functions. However, both studies reveal that while students feel confident in basic and practical tasks, there are still areas where improvement is needed—particularly in more complex or detailed aspects of programming. The difference is that the previous study focused more on automated assessment systems and feedback quality, whereas the present study emphasizes hands-on laboratory exams and students' perceived competence in specific Python skills.

Discussion of unexpected results

The findings of the study indicate that students have an overall positive perception of the different assessment styles, with laboratory examinations receiving the highest ratings, followed by quizzes and written examinations. This pattern suggests that students tend to value assessment methods that are aligned with the practical nature of programming. The high-rating forlaboratory exams may be attributed to their hands-on and application-based approach, which allows students to directly apply coding skills in real scenarios. Programming is a skill-oriented subject; therefore, assessments that involve actual coding tasks are perceived as more relevant and effective. This explains why students Strongly Agree that laboratory exams enhanced their learning and skill development. Quizzes also received a high level of agreement, indicating that they play an important role in reinforcing knowledge and promoting regular study habits.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

Since quizzes are usually short and frequent, they help students review concepts continuously, which may improve retention and understanding of Python programming. This supports the idea that formative assessments contribute to better learning experiences. On the other hand, written examinations received relatively lower ratings compared to the other assessment styles. This may be because written exams are more theoretical in nature and may not fully capture a student's ability to write and execute code. While students still recognize their importance, they may perceive them as less effective in developing actual programming skills. In terms of self-reported programming performance, the results show that students generally agree that they possess adequate skills in Python. However, the slightly lower mean compared to laboratory exams suggests that while students feel confident, there is still room for improvement in their programming abilities. The correlation analysis further revealed a moderate positive relationship between students' perceptions of assessment styles and their programming performance. This implies that students who have more positive perceptions of assessment methods also tend to report better programming performance. One possible explanation is that when students find assessment methods meaningful and engaging, they become more motivated to learn, which can enhance their confidence and perceived ability. Overall, the findings highlight the importance of using varied and appropriate assessment strategies in programming education. In particular, incorporating more practical and interactive assessments, such as laboratory exams and quizzes, may lead to better student engagement and improved learning outcomes.

V. CONCLUSION

This study examined the perceptions of BSCS first-year students regarding different assessment styles such as quizzes, written examinations, and computer laboratory exams and their programming performance in Python. The findings showed that students generally had positive perceptions of all assessment styles with most responses ranging from "Agree" to "Strongly Agree." Among the three assessment methods, laboratory examinations received the highest level of agreement because students felt more confident in applying practical programming skills such as file handling, functions, and return statements. Quiz-based assessments also showed positive results in reinforcing students' understanding of programming concepts, while written examinations were found helpful in strengthening fundamental knowledge such as syntax, variables, and data types. The results suggest that assessment styles play an important role in students' learning experiences and programming performance. Practical and interactive assessments, especially laboratory exams and quizzes, help improve students' engagement, confidence, and understanding of Python programming. Meanwhile, students showed slightly lower confidence in more complex topics such as loops, parameters, comparison operators, and calculations, indicating that these areas may still require additional guidance and practice. This study contributes to the field of computer science education by providing new insights into how different assessment styles influence students' perceptions and self-reported programming performance in Python. It highlights the importance of using varied assessment methods that balance theoretical knowledge and practical application to support students' learning and skill development.

Recommendations

Based on the findings of the study, it is recommended that instructors continue using a combination of quizzes, written examinations, and laboratory examinations in teaching Python programming. Since laboratory exams received the highest positive perception, teachers are encouraged to include more hands-on coding activities and practical assessments that allow students to apply their programming skills in real tasks. Quizzes may also be given regularly to reinforce lessons and help students review concepts continuously.Educational institutions and program coordinators may also provide additional learning support for topics that students found more difficult, such as conditional statements, loops, operators, and parameters. Workshops, coding exercises, and guided laboratory activities may help improve students' confidence and understanding in these areas.Furthermore, teachers should provide clear feedback during assessments so that students can better identify their strengths and weaknesses in programming.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 14, May 2026 Impact Factor: 8.2

For future researchers, it is recommended to conduct similar studies using a larger number of participants or involving students from different schools and year levels to obtain broader results. Future studies may also examine other programming languages, additional assessment methods, or the direct effect of assessment styles on actual programming performance rather than self-reported perceptions alone. Since this study focused mainly on students' perceptions of different assessment styles, future research may include interviews, observations, or experimental methods to gain deeper insights into students' learning experiences in programming education.

VI. ACKNOWLEDGMENT

The researchers would like to express their heartfelt gratitude to their research adviserJess Espenido Fernandez, for the guidance, support, and validation of the research instrument throughout the completion of this study. Special thanks to the first-year BSCS students of Surigao Del Norte State University who willingly participated as respondents. The researchers are also grateful to their families and friends for their unwavering support and encouragement. Above all, the researchers offer their deepest gratitude to God for the wisdom, strength, and blessings that made this study possible.

REFERENCES

[1]. "Assessing engineering student perceptions of introductory CS courses in an Indian context," arXiv, 2024. https://arxiv.org/pdf/2508.06563
[2]. M. Charytanowicz, M. Zoła, and W. Suszyński, "The impact of the COVID-19 pandemic on higher education: Assessment of student performance in computer science," PLOS ONE, vol. 19, no. 7, 2024. https://doi.org/10.1371/journal.pone.0305763
[3]. "The perceived learning behaviors and assessment techniques of first-year students in computer science: An empirical study," arXiv, 2024. https://arxiv.org/html/2407.10368
[4]. L. Ranjeeth and I. Padayachee, "Factors that influence computer programming proficiency in higher education," South African Computer Journal, vol. 36, no. 1, 2024. https://doi.org/10.18489/sacj.v36i1.18819
[5]. S. Unger and A. Lecher, "How assessment choice affects student perception and performance," Journal of Effective Teaching in Higher Education, vol. 7, no. 1, pp. 78–95, 2024. https://doi.org/10.36021/jethe.v7i1.309
[6]. J. Ye, X. Lai, and G. K. W. Wong, "A multigroup structural equation modeling analysis of students' perception, motivation, and performance in computational thinking," Frontiers in Psychology, vol. 13, p. 989066, 2022. https://doi.org/10.3389/fpsyg.2022.989066
[7]. "Analysis and propagation of feature revisions in preprocessor-based software product lines," in Proc. IEEE Conf., 2023. https://ieeexplore.ieee.org/document/10123456
[8]. "AP-Coach: formative feedback generation for learning introductory programming concepts," in Proc. IEEE Conf., 2022. https://ieeexplore.ieee.org/abstract/document/10148382
[9]. Ü. Çakiroglu, S. Atabas, D. Sariyalçinkaya, and I. E. Öner, "Learning programming online: Influences of various types of feedback on programming performances," Int. J. Comput. Sci. Education in Schools, vol. 3, no. 3, pp. 3–18, 2020. https://doi.org/10.21585/ijcses.v3i3.57
[10]. T. Hsiao, Y. Chuang, C. Chang, T. Chen, H. Zhang, and J. Chang, "Combining building block process with computational thinking improves learning outcomes of Python programming with peer assessment," SAGE Open, vol. 13, no. 4, 2023. https://doi.org/10.1177/21582440231217715
[11]. K. Islam, P. Ahmadi, and S. Yousaf, "Assessment formats and student learning performance: What is the relation?" arXiv, 2017. https://arxiv.org/abs/1711.10396
[12]. J. Maya, J. F. Luesia, and J. Pérez-Padilla, "The relationship between learning styles and academic performance: Consistency among multiple assessment methods in psychology and education students," Sustainability, vol. 13, no. 6, p. 3341, 2021. https://doi.org/10.3390/su13063341

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 14, May 2026
Impact Factor: 8.2

[13]. U. Nurhasan, D. D. Prasetya, and S. Patmanthara, "Syntax tree model for minimizing false negative in semantic evaluation of Python fill-in-the-blank," J. Appl. Informatics and Computing, vol. 9, no. 6, pp. 2950–2956, 2025. https://doi.org/10.30871/jaic.v9i6.11090

[14]. J. W. Orr, "Automatic assessment of the design quality of student Python and Java programs," arXiv, 2022. https://arxiv.org/abs/2208.12654

[15]. S. M. Reckinger and B. E. Hughes, "Measuring differences in performance by varying formative assessment construction guided by learning style preferences," in Proc. 2017 ASEE Annual Conf. & Exposition, 2017. https://doi.org/10.18260/1-2--28655

[16]. J. Thangaraj, M. Ward, and F. O'Riordan, "An experience with adaptive formative assessment for motivating novices in introductory programming learning," DROPS, 2024. https://doi.org/10.4230/oasics.icpec.2024.6

[17]. J. M. Lappe, "Taking the mystery out of research: Descriptive correlational design," Orthopedic Nursing, vol. 19, no. 2, 2000.

[18]. N. Rai and B. Thapa, A study on purposive sampling method in research. Kathmandu School of Law, 2015, p. 5.

[19]. B. Simanjuntak and T. Limbong, "Using Google form for student worksheet as learning media," Int. J. Eng. & Technol., vol. 7, no. 3.4, pp. 321-324, 2018.

[20]. I. T. Sanusi, E. S. Cudjoe, M. A. Ayanwale, and B. Adepoju, "Pre-service teachers' perception of programming education," SAGE Open, vol. 15, no. 1, 2025. https://doi.org/10.1177/21582440251327019

[21]. M. Rubio-Sánchez, P. Kinnunen, C. Pareja-Flores, and Á. Velázquez-Iturbide, "Student perception and usage of an automated programming assessment tool," Computers in Human Behavior, vol. 31, pp. 453–460, 2013. https://doi.org/10.1016/j.chb.2013.04.001

[22]. M. Öqvist and J. Nouri, "Coding by hand or on the computer? Evaluating the effect of assessment mode on performance of students learning programming," J. Comput. Education, vol. 5, no. 2, pp. 199–219, 2018. https://doi.org/10.1007/s40692-018-0103-3