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 3, June 2026
Impact Factor: 8.2

Learning Programming Languages: Experiences of Freshmen Computer Science Students

Terence Jade W, Tan[1], Vhan Lester L, Malacura[2], Jessica Rose E, Fernandez[3]
BSCS Students, College of Computing and Information Sciences[1,2]
Faculty, College of Computing and Information Sciences[3]
Surigao Del Norte State University, Surigao City, Philippines
terencejadewtan@gmail.comvhanlester.malacura@gmail.com jfernandez@ssct.edu.ph

Abstract: *: This study is about the programming language learning experiences of freshman BSCS students. With the use of the quantitative descriptive approach involving 50 respondents chosen through the purposive sampling method, the relationship between learning techniques, resources and tools, and motivation and programming learning outcomes, knowledge, and experience is investigated. Survey questionnaires served as research instrument and were statistically analyzed using the weighted mean and Pearson Product-Moment Correlation. It was found that there are significant correlations among the three independent variables, with Resources and Tools giving the highest value (r=0.955), then Learning Methods (r=0.945), and Motivation (r=0.933), all of which are significant at p < .001 level. The results support the rejection of the null hypothesis that these variables do not significantly influence programming outcomes. There are certain recommendations for laboratory improvements and effective instruction approaches such as motivation maintenance and collaborative work with project-based activities.*

Keywords: programming instruction, learning methods, resources and tools, motivation, programming skills

I. INTRODUCTION

As technology continues to evolve, programming and computing skills have grown more significant in today's educational system. Programming education plays an essential role in the development of problem-solving skills, digital literacy, and computational thinking among students in today's era (Vinueza-Morales et al., 2025). Even though programming education holds immense significance and benefits, many students find themselves facing difficulties in understanding concepts like syntax, logic, and algorithm creation.

From previous studies, the lack of facilities and resources impacts programming educational results significantly, especially within school settings and those with scarce facilities (Li, 2023; Hudin & Adil, 2024). Constraints related to money, which prevent students from getting devices or using technology, can limit programming and interest (Li, 2023). It is essential that programming education be combined with practical application to increase motivation and relevance (Li, 2023; Zhengda, 2023).

From recent studies, it has been shown that students' engagement is an important factor in fostering higher order thinking, which involves critical and innovative problem-solving in the process of programming (Li et al., 2023; Tsai et al., 2023). But engagement can be influenced by classroom goals, self-efficacy, and motivational factors facilitated by educators and curriculum developers (Ozturk et al., 2026). Other studies have pointed out the potential impact of gamification, personalized learning, and adaptive feedback as ways to motivate cognitive engagement in programming learning (Ishaq & Alvi, 2023; Garcia-Villegas & Aguero, 2023).

Nevertheless, there are still gaps in the existing literature. In this regard, although there are numerous works focusing on single strategies, such as gamification or design thinking, there is little knowledge about how various teaching strategies together contribute to the long-term understanding and retention of skills in programming learning (Cheah, 2020; Vinueza-Morales et al., 2025). Besides, many studies are conducted in controlled or specialized conditions rather

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 3, June 2026
Impact Factor: 8.2

than general educational settings. It calls for further research that is focused on everyday classroom learning. Thus, the rationale for conducting the study under consideration is clear.

Objectives of the Study

This study aims to:

(1) Studying the experiences of learners who have studied programming languages; (2) Discovering what the major difficulties are with learning programming languages; (3) Examining the factors that contribute to understanding and developing skills in programming; (4) Examining how learning techniques and available resources influence the effectiveness of programming education; and (5) Offering suggestions for improving programming education programs.

II. REVIEW OF RELATED LITERATURE

Programming is a key aspect of computer science courses. But studying programming does not only mean gaining knowledge in terms of syntax; it also entails having the capacity for cognitive reasoning, problem solving, and emotional control. According to studies, students frequently encounter difficulties when moving from learning about the basics of programming to algorithmic reasoning and the process of program development (Cheah, 2022; Ngadengon et al., 2025). The aforementioned problems can induce cognitive overload, frustration, and decreased self-confidence on the part of the learners (Abu Bakar et al., 2025; Wang et al., 2021).

Problems in learning how to program often stem from the demand for abstract thinking, debugging, and logical reasoning. According to research, knowing students' experiences is fundamental to developing successful pedagogical techniques in overcoming these obstacles (Figueiredo & Garcia Penalvo, 2022; Napalit et al., 2023).

Learning Methods in Programming Education

The use of active learning techniques in programming has proven to be an effective method of instruction. Project learning, peer assessment, collaboration in programming, and live programming are some of the many effective strategies that enhance students' knowledge acquisition more than pure lecture-based teaching methods (Berssanette& de Francisco, 2021; Masegosa et al., 2024).

Learning through peer feedback and collaboration for solving problems also improves engagement and critical thinking skills in programming education (Hsiao et al., 2022; Li et al., 2023; Tsai et al., 2023). The use of gamification in the e-learning environment to motivate students increases their level of engagement (Garcia Villegas & Aguero, 2023; Ishaq & Alvi, 2023). This evidence highlights the significance of learning methods as a crucial variable that affects programming education results.

Resources and Digital Tools

Availability of technological resources is an important aspect for learning how to program. Digital tools, IDEs, and online tools help students in developing their coding abilities and in debugging their codes. Research findings have indicated that when students use technology effectively, they exhibit better engagement and performance levels (Asgari et al., 2024; Elnaffar et al., 2025).

The emergence of AI-powered tutoring systems and the use of generation tools have also become means for supporting learners in programming education, helping them recognize mistakes in their programs and improve their code (Elnaffar et al., 2025; Rojas Galeano et al., 2025). Nevertheless, insufficient availability of resources can become an obstacle to learning, limiting the chances of practicing (Abu Bakar et al., 2025).

Motivation and Psychological Factors

Self-motivation and self-efficacy have a great effect on students' dedication when studying programming classes. There is a strong connection between emotional and cognitive engagement and good academic performance (Zhou & Tsai, 2022). If students understand that what they learn will be useful in the real world, they can maintain their motivation to study programming (Vinueza Morales & Cevallos Ayon, 2025).

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 3, June 2026
Impact Factor: 8.2

Goal structure of the classroom and self-efficacy can also be regarded as significant predictors of programming performance (Ozturk et al., 2026). Those learners who possess higher self-confidence levels usually perform well and exhibit resilience in dealing with challenging coding problems (Yukselturk&Altiok, 2017). These empirical findings support the significance of the motivational factors in enhancing programming skills.

Hands-On Laboratory Learning

Laboratory experiments have been adopted as the bedrock of computer science teaching, owing to the ability of such exercises to create a link between theory and practice. Through regular lab work and task calibration, first year computer science learners get a chance to learn through trial and error, thus enhancing their technical skills and problem solving efficiency (Abrahams & Millar, 2024; Benin, 2025; Miller & Chen, 2026). Laboratory environments offer a well structured learning process that enables learners to acquire coding skills through regular task execution and interactions within development systems (Astin, 2025).

Additionally, the role of augmented learning via Large Language Models (LLMs) plays an important role as a moderating element in today's educational environment, providing immediate debugging and individualized feedback that influence the academic path of a student (Zdravkova&Ilijoski, 2025; Zhang et al., 2025; Khan & Rahim, 2025). When students consider AI tools to be high-value assets for improving comprehension skills and not merely a shortcut, their sense of self-efficacy and digital literacy will greatly increase (Huang & Wang, 2025; Becker et al., 2024). In conclusion, the combination of practical work in laboratories along with the use of augmented learning tools increases the effectiveness of learning processes for first-year students to meet industry requirements (Oguntibeju, 2026; Zdravkova&Ilijoski, 2025).

Research Gap

While there is a plethora of research on such variables individually, very little research has been conducted with regards to integrating all these variables in a single model where the relationship between them and their collective influence on programming language learning outcomes can be explored (Abu Bakar et al., 2025; Berssanette& de Francisco, 2021; Ngadengon et al., 2025). Current literature mostly focuses on the impact of such intervention strategies on programming learning outcomes individually without any attempt to analyze them altogether for assessing their collective influence. The present research aims to fill in this gap.

Synthesis

Programming is an integral part of the computer science curriculum, whereby the learner must not only master skills in coding and other technical aspects but also must acquire cognitive, analytic, and emotional skills. Past literature reveals that programming entails more than syntax because it requires algorithms, logic, and problem-solving skills. Research done by Cheah (2022) and Ngadengon et al. (2025) has shown that new programmers face challenges such as abstraction, debugging, and logical thinking; these pose significant barriers in moving from theory to practice. Similar findings are reported by Abu Bakar et al. (2025) and Wang et al. (2021), where it is revealed that such problems tend to cause mental fatigue and frustration among programmers.

In order to overcome these challenges, there has been increasing academic evidence of the efficacy of active learning strategies in the instruction of computer programming. According to the literature, project-based learning, peer coding, peer review, and live coding are all more effective in developing conceptual understandings compared to conventional lectures. Bersanatte and de Francisco (2021) argue that active participation in coding tasks encourages learners to achieve deeper cognitive gains in terms of learning and retaining new knowledge since they are engaged in constructing their understandings directly. This conclusion is supported by Hsiao et al. (2022), Li et al. (2023), and Tsai et al. (2023), who show that peer interaction and feedback help learners to remain actively involved in the process and develop advanced thinking skills. Finally, gamification techniques, mentioned in Garcia Villegas and Aguero (2023) and Ishaq & Alvi (2023), also aid in motivating learners through incentives and competition.

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 3, June 2026
Impact Factor: 8.2

Technological resources and digital tools have also played a vital role in teaching programming skills. Availability of software applications such as IDEs, online coding sites, and digital AI-based applications enables the learner to engage in consistent practice. According to studies carried out by Asgari et al. (2024) and Elnaffar et al. (2025), proper usage of digital tools is critical in improving the rate of learning and enhancing students' participation in the class. Specifically, AI-based tutoring services and LLMs are useful digital tools since they help learners with bug fixing, code fixing, and concept clarification. Results from research conducted by Rojas Galeano et al. (2025), Zdravkova&Ilijoski (2025), and Khan & Rahim (2025) reveal that using such digital tools as part of learning is crucial in improving academic performance.

The psychological aspect of motivation also plays an essential part in programming success. It has been demonstrated that the degree of cognitive and emotional engagement correlates directly with perseverance and success in programming classes. As stated by Zhou and Tsai (2022) and Ozturk et al. (2026), self-efficacy and classroom goal structures are major factors which contribute to effective programming results. Students who are more confident and have set achievement-oriented goals usually manage to persist in their attempts to tackle programming tasks. Similarly, Yukselturk and Altiok (2017) argue that confidence facilitates problem-solving skills and persistence in programming.

Laboratory education enhances programming education even further by linking the theories taught to actual practice in the classroom setting. There is no doubt about the importance of laboratory education as an educational method that promotes learning through practice and experimentation. Abrahams and Millar (2024) and Benin (2025) highlight that laboratory education not only improves academic performance but also enables students to gain practical skills through a trial-and-error approach to learning. This is also true according to Miller and Chen (2026) and Oguntibeju (2026). They believe that practical learning settings lead to greater engagement and improved problem-solving skills.

The inclusion of LLMs in programming learning in laboratories is yet another avenue through which these advantages can be expanded through the offering of timely and tailored support services. Research conducted by Huang and Wang (2025) and Becker et al. (2024) highlights that learners utilizing LLMs for coding demonstrate increased computational thinking capabilities, self-efficacy, and code understanding abilities.

Although extensive research on programming education exists, there are still substantial knowledge gaps. Current research is largely centered on individual factors like active learning, gamification, artificial intelligence-based training, and motivation separately. There are few studies conducted that consider all these factors together and assess their impact on programming education outcomes in the domain of computer sciences. The importance of bridging this knowledge gap cannot be overstated since the integration of instructional techniques, technology tools, motivational elements, and laboratory practice is required to foster successful programming education outcomes. In this regard, the present study aims to make a contribution in this direction.

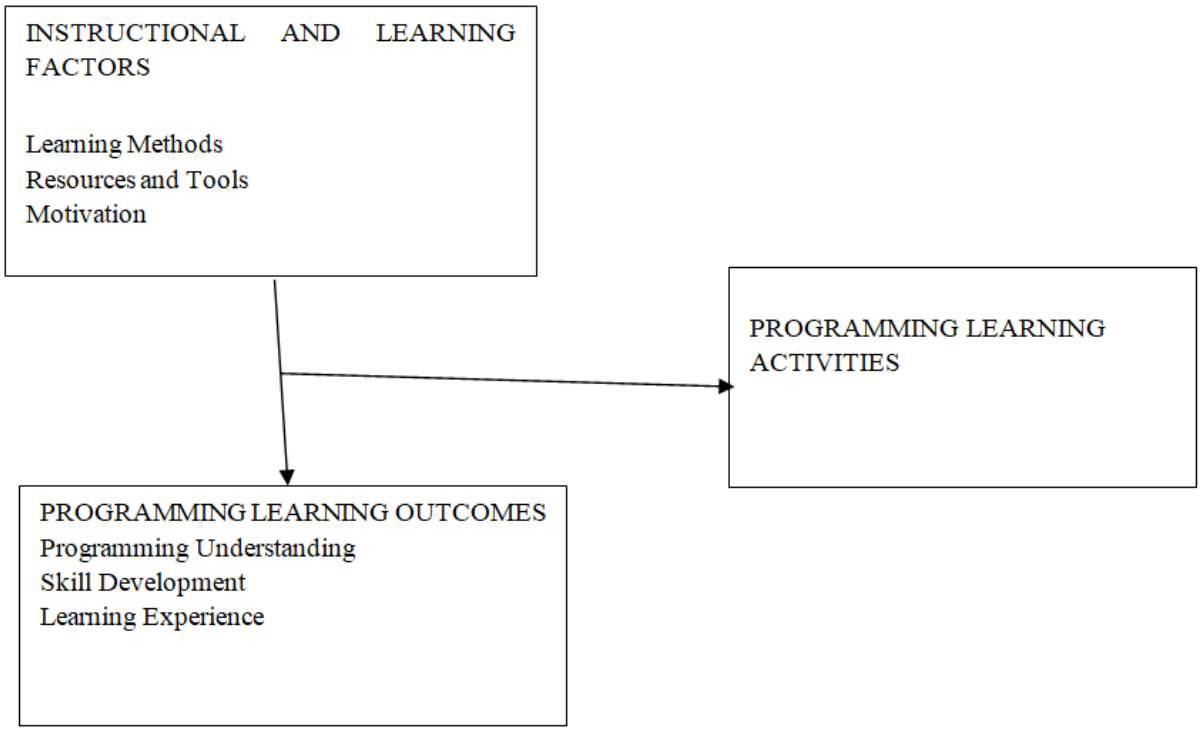
III. RESEARCH METHODOLOGY

This chapter outlines the research design, data collection techniques, and analysis methods used in the study. It is written with enough detail to allow for replication of the research process.

Conceptual Framework

For this study, there are many things that may influence individuals' learning experience with respect to the programming language learning process. Among the many factors that will influence this type of learning experience are the learning style, learning materials, learners' motivation, and practice among others. Independent variables include: (1) Learning Style – Pedagogical strategy employed in teaching programming, including strategies aimed at fostering problem-solving skills, gamification and project-based learning, and influence on learner motivation; (2) Resources and Tools – availability of computing tools and technologies such as IDEs, which support practice and coding skills; and (3) Motivation – Students' motivation due to instructional strategy and relevance. Dependent variables include: (1) Programming understanding; (2) Skill Development – Improvement in programming skills; and (3) Learning experience.

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 3, June 2026
Impact Factor: 8.2



3.1 Research Design

The study will employ a quantitative-descriptive type of research design to investigate the learning experiences of BSCS freshmen students regarding programming language. Quantitative research is considered the best choice since it will enable the researchers to obtain quantifiable information on the experiences, difficulties, and factors affecting the ability of the students to learn programming languages. Descriptive research will help in identifying and describing the present condition, experience, and problems of learning programming languages by the students.

The independent variables for this research will include learning approach, learning materials and tools, and motivation. These will be likely to impact the dependent variables including understanding of programming, programming skills acquired, and learning experiences. The data collection in this case will involve surveys using questionnaires, while data analysis will be quantitative in nature. This type of research design will prove effective since it offers numerical information on the effects of instruction and learning on programming.

3.2 Population and Sampling

For this research study, the participants to be involved are 50 freshmen students currently studying in the BSCS course program and undertaking introductory courses related to programming. Such selection is made based on the fact that these freshmen are at an early level of acquiring knowledge about programming language concepts, hence will face difficulties in their programming education journey.

Purposive sampling will be used by the researchers in determining the participants in the study. Purposive sampling is ideal for this kind of study since the participants involved were chosen based on their involvement in programming-related subjects. In choosing participants for the study, there should be at least three qualities that must be met, which are: (1) Participants should be freshman BSCS students; (2) They should be enrolled in any programming subjects; and (3) Participants should possess experience with programming languages. It is vital to choose freshmen students as participants in this study since the researchers could pinpoint the problems freshmen encounter with their programming subjects.

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 3, June 2026
Impact Factor: 8.2

3.3 Data Collection

A structured survey questionnaire will serve as the main tool used to gather data during this study. The questionnaire will be conducted using Google forms to make it more convenient for respondents. This questionnaire will have two major parts. Part I will contain information about the independent variables, which include learning techniques, resources, tools, and motivation, and identify the effect of these variables on students' experiences with programming. Part II will have questions concerning the dependent variables, including programming knowledge, skills, and experiences.

Participants will be made aware of the objective of the experiment before completing the survey. Participation will be voluntary, and confidentiality will be guaranteed. The survey will be distributed via various online media outlets, such as Messenger, class chat groups, or emails.

3.3.1 Data Analysis

The collected data will be analyzed using the following statistical tools. Frequency and Percentage Distribution will be used to describe the profile of respondents and summarize their responses regarding programming language learning experiences. Weighted Mean will be used to measure the average responses of participants regarding the factors affecting programming learning and the level of programming understanding and skill development. The weighted mean will be interpreted using the scale presented in Table I.

TABLE I. SCALE OF INTERPRETATION FOR LIKERT RESPONSES

Scale	Range	Verbal Description	Interpretation
4	3.50 – 4.00	Strongly Agree	Very High
3	2.50 – 3.49	Agree	High / Moderate
2	1.50 – 2.49	Disagree	Low
1	1.00 – 1.49	Strongly Disagree	Very Low

Pearson Product-Moment Correlation Coefficient (Pearson r) will be used to test for the existence of any significant relationship between the independent variables (learning approaches, learning tools, and motivational factors) and the dependent variables (programming knowledge and skills). Pearson r will reveal the existence of a relationship as well as its strength and direction. For purposes of this research, a level of significance value of 0.05 will be adopted; a p-value of less than 0.05 will reveal a significant relationship. The data collected will then be entered and analyzed through a statistical package for ease and efficiency in analysis.

IV. RESULTS AND DISCUSSION

This chapter presents the results of the data gathered from the survey on the programming language learning experiences of Computer Science students. The analysis includes the demographic profile of the respondents, the level of instructional and learning factors, the level of programming outcomes, and the relationship between these variables.

4.1 Profile of the Respondents

The following tables describe the demographic characteristics of the 50 respondents involved in the study.

4.1.1 Age Distribution

TABLE 1. FREQUENCY AND PERCENTAGE DISTRIBUTION OF RESPONDENTS BY AGE

Age Group	Frequency (f)	Percentage (%)
17–18 years old	8	16.0%
18 years old and above	42	84.0%

As indicated by Table 1 below, the respondents were distributed according to their age groups. From the data presented, it is evident that most of the freshmen surveyed (84%) fall under the “18 years and above” age group. This implies that the learners are mature and have adapted to the college environment while being focused on their studies.

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 3, June 2026
Impact Factor: 8.2

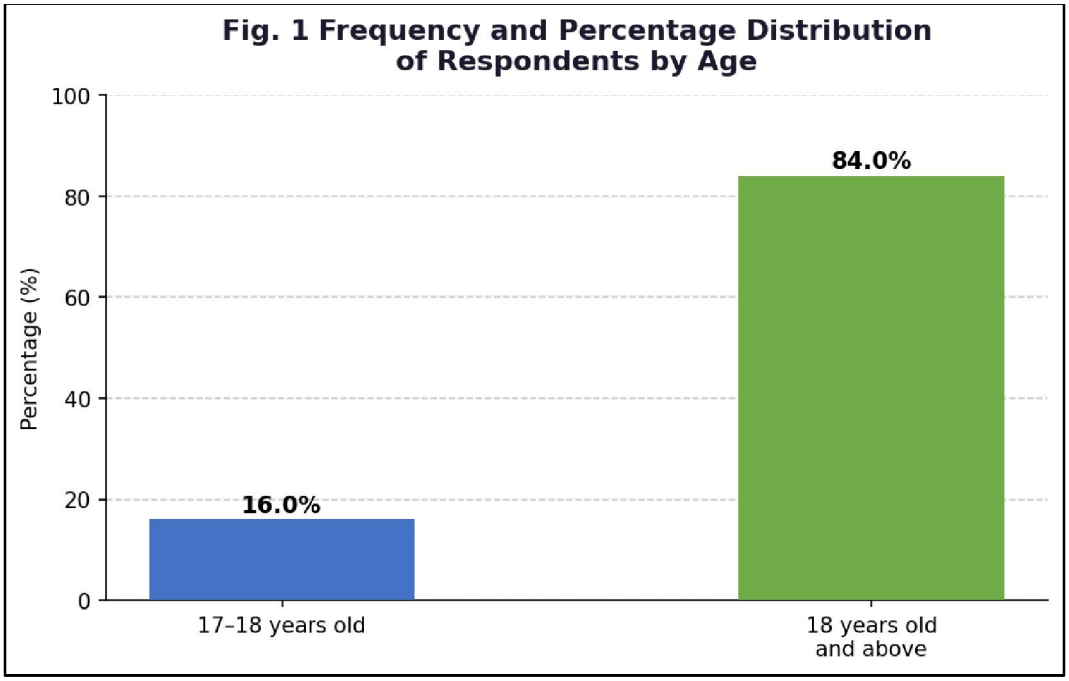


Fig. 1 Frequency and Percentage Distribution of Respondents by Age

Figure 1 shows the frequency and percentage distribution of the respondents based on their age range. From Figure 1, it is evident that the highest number of respondents, 84%, fall under the "18 years old and above" age range, whereas 16% of them fall under the "17-18 years old" age range. This distribution indicates that most of the BSCS freshmen enrolling in the program are already adults, indicating an adult learner population. Based on the age range of the respondents, one can conclude that most learners enrolling in programming lessons have a certain amount of preparedness and readiness to tackle the challenge of learning programming languages.

4.1.2 Sex Distribution

TABLE 2. FREQUENCY AND PERCENTAGE DISTRIBUTION OF RESPONDENTS BY SEX

Sex	Frequency (f)	Percentage (%)
Male	31	62.0%
Female	19	38.0%

Table 2 shows the categorization of the respondents in accordance with their sex. With regard to sex, 62 percent of the total respondents are males, whereas 38 percent are females. Although the number of males is greater compared to that of females, it should be noted that there is a considerable number of females as well.

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 3, June 2026
Impact Factor: 8.2

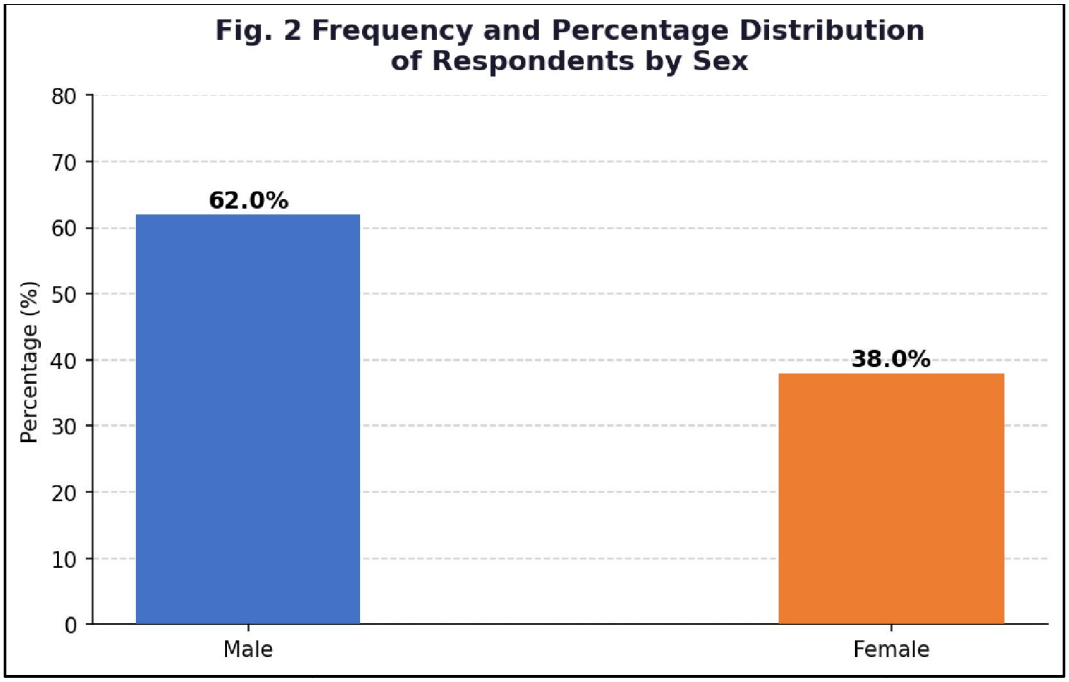


Fig. 2 Frequency and Percentage Distribution of Respondents by Sex

Figure 2 illustrates the frequency and percentage distributions of the respondents in terms of their sexes. As illustrated in the graph above, the male students make up the larger share of respondents at 62%, compared to only 38% for female students. The gender distribution observed in this case follows trends witnessed in other cases where male computer science students outnumber female computer science students. This is mainly because of sociocultural factors. Regardless of this common trend, the presence of female respondents in this research is significant. This is because of the increasingly high participation of female computer science students. Such gender diversity makes it possible to understand programming better.

4.2 Analysis of Learning Factors and Outcomes

4.2.1 Learning Methods

TABLE 3. MEAN SCORES FOR LEARNING METHODS

Dependent Variable	Indicators: The learning methods help me to...	Mean	Verbal Description
Programming Understanding	1. understand programming concepts better.	3.06	Agree
	2. improve my understanding of programming lessons.	3.04	Agree
	3. grasp coding concepts more easily.	2.92	Agree
Subscale Mean		**3.01**	**Agree**
Skill Development	4. develop my coding skills effectively.	2.98	Agree
	5. improve my ability to write correct code.	2.92	Agree
	6. practice and enhance my programming abilities.	3.08	Agree
Subscale Mean		**2.99**	**Agree**
Learning Experience	7. enjoy my overall programming learning experience.	2.94	Agree
	8. feel more confident while learning programming.	2.96	Agree
	9. stay interested in programming lessons.	3.02	Agree
Subscale Mean		**2.97**	**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 3, June 2026
Impact Factor: 8.2

According to the analysis, Learning Methods are seen as having a strong correlation with Programming Understanding (M = 3.01), especially in facilitating concept comprehension (M = 3.06). The students also acknowledged that the methods were effective in fostering Skill Development (M = 2.99) and providing them with a positive Learning Experience (M = 2.97). Essentially, the students acknowledge that the learning methods used in their courses positively influence them.

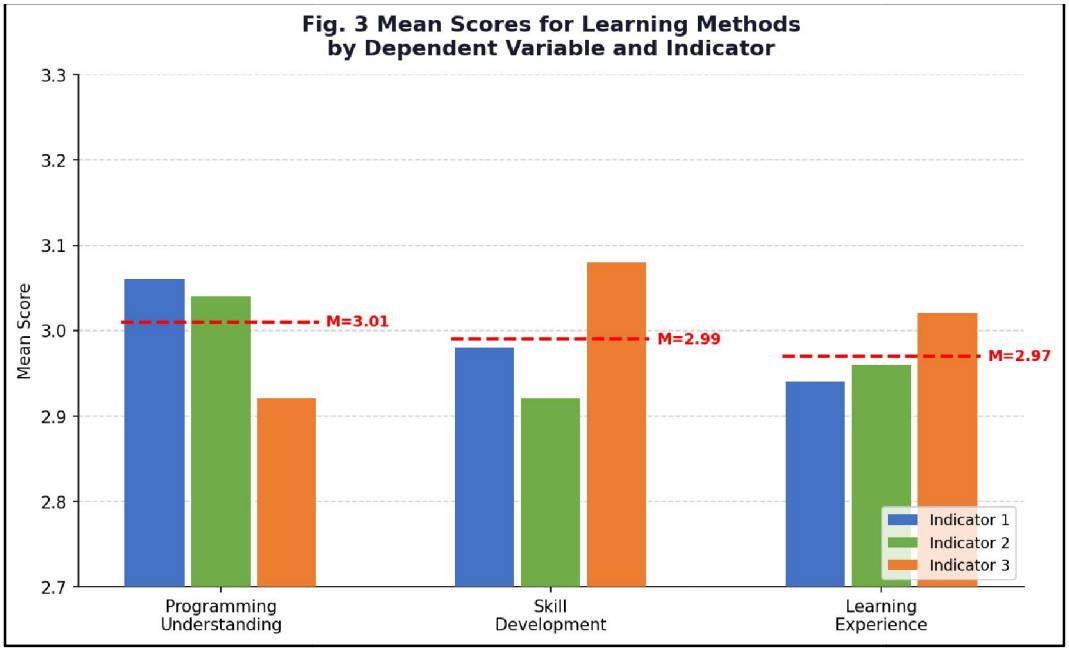


Fig. 3 Mean Scores for Learning Methods

Figure 3 presents the mean responses of the participants on the three dependent variables, namely Programming Understanding, Skill Development, and Learning Experience, under the influence of Learning Methods. Based on Figure 3, it can be seen that the mean responses for all the subscales fall under the category "Agree," where the highest subscale mean was obtained in Programming Understanding with a score of 3.01, while those for Skill Development and Learning Experience were slightly lower at 2.99 and 2.97, respectively. It is evident that the use of learning methods in the instruction of programming classes was viewed by the students as highly effective in helping improve the understanding of concepts, and at the same time, contribute to their skill development and satisfaction.

4.2.2 Resources and Tools

TABLE 4. MEAN SCORES FOR RESOURCES AND TOOLS

Dependent Variable	Indicators: The resources and tools help me to...	Mean	Verbal Description
Programming Understanding	10. understand programming concepts more clearly.	2.88	Agree
	11. visualize how codes and programs work.	3.06	Agree
	12. follow lessons through examples and demonstrations.	2.94	Agree
Subscale Mean		**2.96**	**Agree**
Skill Development	13. improve my coding skills through practice.	3.02	Agree
	14. develop my ability to write and test programs.	2.92	Agree
	15. enhance my problem-solving skills in programming.	2.80	Agree
Subscale Mean		**2.91**	**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 3, June 2026
Impact Factor: 8.2

Learning Experience	16. have a more engaging programming learning experience.	2.90	Agree
	17. stay interested during programming lessons.	2.92	Agree
	18. feel more comfortable while learning coding.	2.90	Agree
Subscale Mean		**2.91**	**Agree**

Resources & Tools are most useful in terms of aiding students in terms of their learning of programming (M=2.96). Although Skill Development (M=2.91) and Learning Experience (M=2.91) both scored high, the data clearly indicates that technological support is mainly used as an aid in making abstract logical reasoning clear through demonstrations. It can be seen from the data that having tools and resources always aids the students in learning.

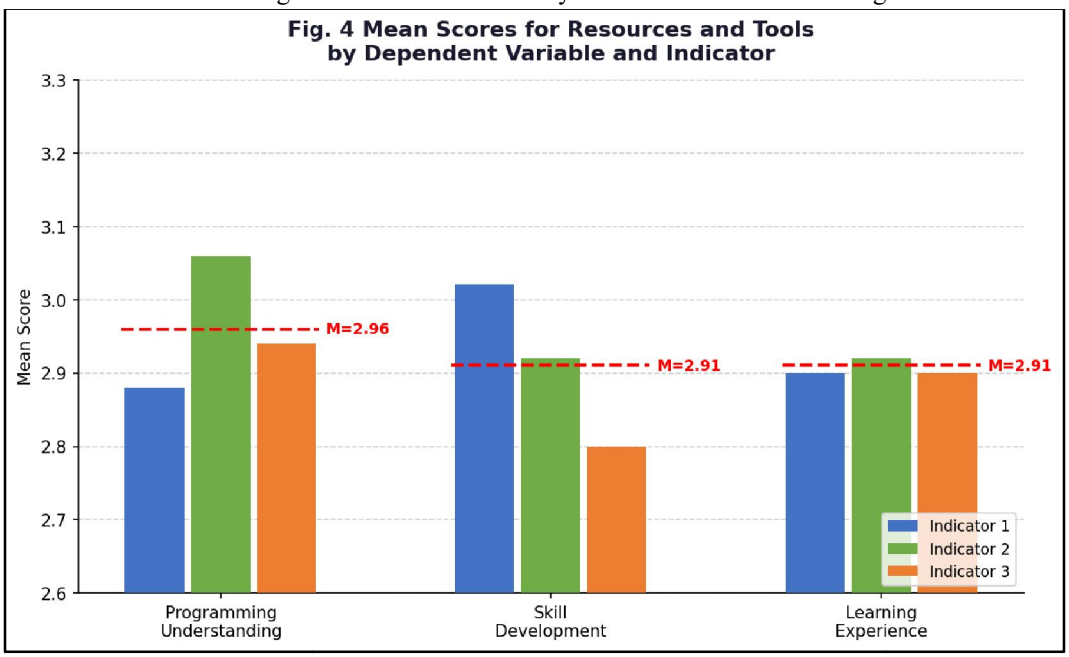


Fig. 4 Mean Scores for Resources and Tools

Figure 4 shows the mean score of Resources and Tools for each of the three programming learning outcomes subscales. As seen from Figure 4 below, the Programming Understanding subscale has achieved the highest mean of 2.96, with a particular influence of the visualizing code/program indicator (M=3.06). Skill Development and Learning Experience had subscale means of 2.91. This implies that though resources and tools play a major role in helping students understand concepts related to programming, they still have a considerable impact on skill acquisition and the whole process of learning. From these findings, it is evident that the students place great importance on having access to tools such as IDEs and demonstration sites which would help them grasp concepts better.

4.2.3 Motivation

TABLE 5. MEAN SCORES FOR MOTIVATION

Dependent Variable	Indicators: My motivation in learning helps me to...	Mean	Verbal Description
Programming Understanding	19. understand programming concepts more clearly.	2.90	Agree
	20. focus better during coding lessons.	2.92	Agree
	21. put more effort into understanding difficult topics.	3.08	Agree
Subscale Mean		**2.97**	**Agree**
Skill Development	22. improve my coding skills through continuous	3.08	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 3, June 2026
Impact Factor: 8.2

	practice.		
	23. develop my ability to solve programming problems.	2.96	Agree
	24. practice programming even outside class hours.	3.08	Agree
Subscale Mean		**3.04**	**Agree**
Learning Experience	25. enjoy learning programming more.	3.10	Agree
	26. stay interested during programming lessons.	3.04	Agree
	27. feel more positive about coding activities.	3.00	Agree
Subscale Mean		**3.05**	**Agree**

Motivation produced the greatest subscale means in Learning Experience (Mean = 3.05) and Skill Development (Mean = 3.04). In other words, motivation enables students to feel the enjoyment associated with learning programming (Mean = 3.10) and motivates them to practice on their own time beyond the classroom environment (Mean = 3.08).

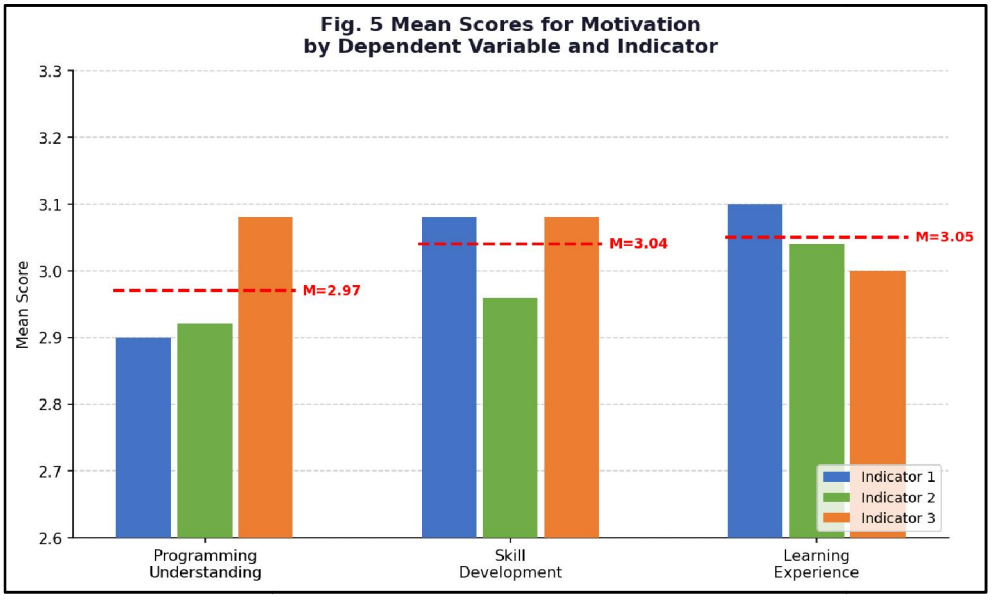


Fig. 5 Mean Scores for Motivation

The average mean scores of Motivation are displayed for all three categories of programming learning outcomes in Fig. 5. From Fig. 5, it can be seen that Learning Experience gained the highest mean subscale score of 3.05, Skill Development secured the second place with a mean subscale score of 3.04, and Programming Understanding came third with a mean subscale score of 2.97. This implies that motivation is the strongest among all factors contributing to the learners' satisfaction with their programming learning process especially in relation to instilling positive attitudes towards programming activities and maintaining students' interests during the lessons (M=3.10 and M=3.04 respectively). Moreover, a higher subscale mean score of Skill Development implies that motivated students tend to program more often without any supervision from instructors, which helps them improve their skills faster.

4.3 Relationship Between Variables

TABLE 6. CORRELATION MATRIX: INSTRUCTIONAL FACTORS VS. LEARNING OUTCOMES

Independent Variable	Pearson r	p-value	Interpretation	Decision
Learning Methods	0.945	< .001	Significant	Reject H_0
Resources and Tools	0.955	< .001	Significant	Reject H_0
Motivation	0.933	< .001	Significant	Reject H_0

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 3, June 2026
Impact Factor: 8.2

According to the results from the use of Pearson Product-Moment Correlation, the p-value is below the 0.05 level of significance for all factors considered. Thus, the Null Hypothesis (H0) is REJECTED in each case. In learning Methods, we get r=0.945 ($p < .001$); in Resources and Tools, we have r=0.955 ($p < .001$); and in Motivation, r=0.933 ($p < .001$). This means that a significant relationship exists between instructional/learning factors and the programming outcomes of BSCS students. The very high r-values indicate that the better the methods, tools, and motivation, the higher the level of performance.

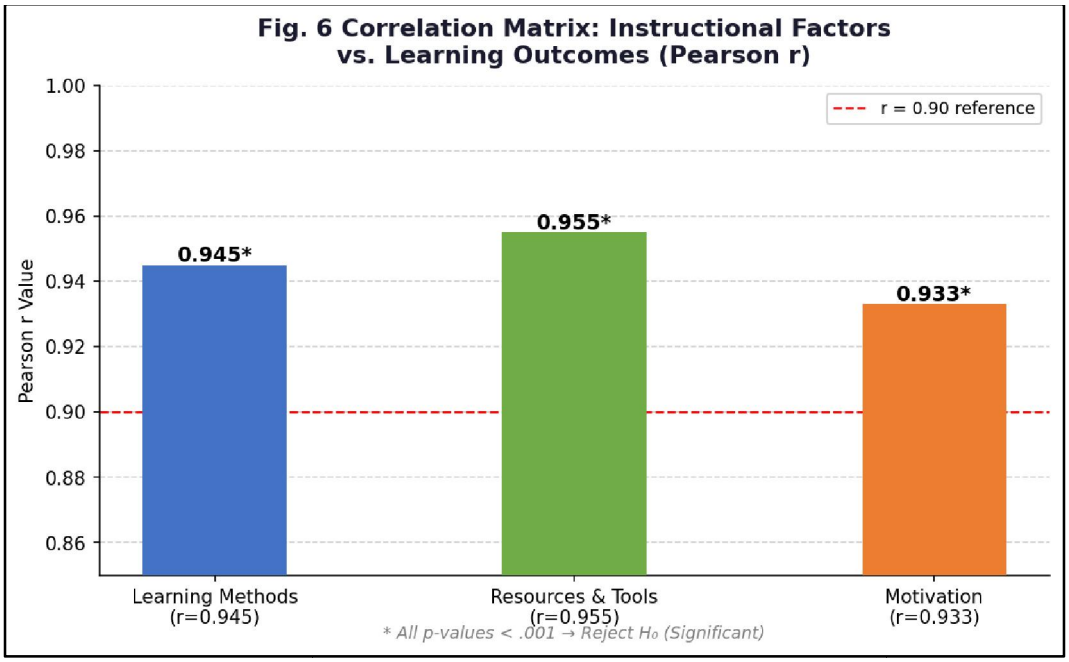


Fig. 6 Correlation Matrix: Instructional Factors vs. Learning Outcomes

Figure 6 displays the correlation matrix of the strength and direction of the relationship of each of the instructional factors with the total programming learning outcomes of the respondents. According to Figure 6, all the three independent variables produced very strong positive Pearson r correlation coefficients: Resources and Tools (r=0.955), Learning Methods (r=0.945), and Motivation (r=0.933), all with significance levels below .001. This means that the more effective the learning methods, the more available the resources and tools, and the greater the motivation, the more proficient the freshman BSCS students will become in their programming knowledge and skills. The null hypotheses were thus rejected regarding all the three factors since there is indeed a significant relationship between them and programming learning outcomes. It should be noted, however, that the greatest correlation coefficient obtained was from the variable Resources and Tools, which implies that access to and use of technology is the most important factor.

V. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

5.1 Summary of Findings

This research looked at the correlation between the factors that affect the instructional/learning process and the output of the BSCS freshmen programmers. With regard to the demographic data about the respondents, most of them were already 18 years old or more (84%) while 62% of them are males. As for the instructional factors, the one with the highest mean was Motivation (mean of 3.02) followed by Learning Methods (mean of 2.99) and lastly, Resources and Tools with a mean score of 2.93. As for the programming results, the area where the students scored the highest in terms of their level of agreement was related to the enjoyment during the learning process and the visualization of code using demonstration techniques. According to the statistical tests involving the use of Pearson r, the factors were positively correlated, with Resources and Tools having the highest coefficient (r=0.955), followed by Learning Methods (r=0.945) and Motivation (r=0.933). Since the p-value is less than 0.001, the null hypothesis was rejected.

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 3, June 2026
Impact Factor: 8.2

5.2 Conclusions

It is clear from the results obtained that instructional variables such as the teaching approach, technical assistance and student motivation have a great role to play in helping students master programming languages. Although the students have very high internal motivation levels, the technical assistance and equipment available are the best predictors of whether they will be able to acquire the skills, implying that it is through technical assistance that motivation leads to skill acquisition. In addition, the teaching techniques used by the teachers help to make the link between theoretical concepts and practical application. The rejection of the null hypothesis in all the categories implies that there is a need for a complete integration of modern teaching approaches, technical assistance and student motivation for the success of Computer Science freshmen.

5.3 Recommendations

(1) To the School Administrators: Stress the need for continued improvement of computer labs and provision of up-to-date Integrated Development Environments (IDEs) since data shows that technological availability is the biggest predictor of success. (2) To the Programming Teachers: Keep on applying collaborative learning methods while increasing live demonstrations and visual aids so that students can understand more complicated syntax. (3) To the BSCS Students: Remain highly motivated through practice in coding during times other than lessons as well as engaging in individual projects of interest. (4) To the Future Researchers: Carry out a similar study including the Practice Opportunities variable separately or extend research to include higher year levels to test whether these learning variables will be equally important as programming languages evolve.

VI. ACKNOWLEDGEMENT

The researchers would like to express their heartfelt gratitude to their research adviser for the invaluable guidance and support throughout the course of this study. The researchers also wish to thank the College of Computer and Information Sciences of Surigao Del Norte State University for the support and resources extended during the conduct of this research. Finally, the researchers are deeply grateful to the fifty (50) Bachelor of Science in Computer Science freshmen students who willingly gave their time and honest responses as respondents of this study.

REFERENCES

1. Abrahams, I., & Millar, R. (2008). Does practical work really work? A study of the effectiveness of practical work as a teaching and learning method in school science. International Journal of Science Education, 30(14), 1945–1969.
2. 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.
3. Asgari, M., Tsai, F. C., Mannila, L., & Sadique, K. M. (2024). Students' perspectives on using digital tools in programming courses: A cross-country case study. Discover Education, 3, Article 57.
4. Astin, A. W. (1999). Student involvement: A developmental theory for higher education. Journal of College Student Development, 40(5), 518–529.
5. Becker, B. A., Prather, J., Reeves, B. N., Denny, P., Finnie-Ansley, J., Hellas, A., Leinonen, J., Loksa, D., Pieterse, V., & Santos, E. A. (2024). An empirical study on usage and perceptions of large language models in academic software engineering projects. arXiv.
6. Benin, E. S. (2025). Impact of computer science laboratories on students' academic performance and personal development. UniProjects Repository.
7. Berssanette, J. H., & de Francisco, A. C. (2021). Active learning in the teaching/learning of computer programming: A systematic review. Journal of Information Technology Education: Research, 20, 201–220.
8. Cheah, C. S. (2022). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. Contemporary Educational Technology, 12(2), ep272.

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 3, June 2026
Impact Factor: 8.2

9. Elnaffar, S., Rashidi, F., &Abualkishik, A. Z. (2025). Teaching with AI: A systematic review of chatbots, generative tools, and tutoring systems in programming education. arXiv.
10. Figueiredo, J., & García Peñalvo, F. J. (2022). Design science research applied to difficulties of teaching and learning initial programming. Universal Access in the Information Society.
11. Garcia Villegas, C., & Aguero, N. L. (2023). The gamification of e-learning environments for learning programming. JOIV: International Journal on Informatics Visualization, 7(2), Article 1602.
12. Hsiao, C. H., Chang, J. Y., & Tseng, K. H. (2022). Peer review and feedback in programming courses: Impacts on student performance and engagement. Journal of Educational Computing Research, 60(5), 1334–1355.
13. Huang, J., & Wang, S. (2025). LLM-based collaborative programming: Impact on students' computational thinking and self-efficacy. Humanities and Social Sciences Communications, 12(1).
14. Ishaq, K., & Alvi, A. (2023). Personalization, cognition, and gamification-based programming language learning: A state-of-the-art systematic literature review. arXiv.
15. Khan, M. A., & Rahim, A. (2025). Exploring the impact of LLM prompting on students' learning and technical task completion. Computers, 4(3), Article 31.
16. Li, Z. (2023). Research on the application of programming in high school teaching. Lecture Notes in Education Psychology and Public Media, 19, 186–189.
17. Masegosa, A. R., Cabañas, R., Maldonado, A. D., & Morales, M. (2024). Learning styles impact students' perceptions on active learning methodologies: Live coding and programming exercises. Education Sciences, 14(3), 250.
18. Miller, K., & Chen, R. (2026). Assessing the impact of virtual and physical laboratories on learning engagement and academic performance among university students. ResearchGate.
19. Napalit, F., Tanyag, B., So, C. L., Sy, C., & San Pedro, J. R. (2023). Examining student experiences, challenges, and perceptions in computer programming. International Journal of Research Studies in Education, 12(8), 101–112.
20. Ngadengon, Z., Subramaniam, T. S., Yasak, Z., Ideris, N. A., &Ramly, Z. M. (2025). Evaluating the difficulties in programming learning: Insights from polytechnic students. International Journal of Academic Research in Progressive Education and Development, 14(1), 1051–1065.
21. Oguntibeju, O. O. (2026). Harnessing the power of virtual and hands-on labs: Learning by doing and problem-solving. Public Health Genomics.
22. Öztürk, Ö., et al. (2026). A path model for K–8 students' programming performance: The role of classroom goal structure and self-efficacy. Education and Information Technologies.
23. Rojas Galeano, S., Tejada, J., & Marmolejo-Ramos, F. (2025). Student attitudes toward AI in programming education: Benefits and challenges. arXiv.
24. Tsai, F. C., Mannila, L., & Sadique, K. M. (2023). Learning engagement and higher-order thinking through programming activities. Procedia Computer Science, 223.
25. Vinueza Morales, M., & Cevallos Ayón, E. (2025). Teaching programming in higher education: A bibliometric analysis of trends, technologies, and pedagogical approaches. Frontiers in Education, 10.
26. Wang, W., Kwatra, A., Skripchuk, J., Gomes, N., Milliken, A., Martens, C., & Price, T. (2021). Novices' learning barriers when using code examples in open-ended programming. arXiv.
27. Yukselturk, E., &Altiok, S. (2017). Factors affecting the success of novice programmers in introductory programming courses. Journal of Computer Assisted Learning, 33(3), 233–245.
28. Zdravkova, K., &Ilijoski, B. (2025). The impact of large language models on computer science student writing and task performance. International Journal of Educational Technology in Higher Education, 22, Article 32.
29. Zhang, L., et al. (2025). Enhancing programming performance, learning interest, and self-efficacy: The role of large language models in digital education. Applied Sciences, 13(7), Article 555.
30. Zhou, M., & Tsai, C. C. (2022). Emotional and cognitive engagement in programming learning. Journal of Educational Computing Research, 60(3), 840–865