The Effect of Programming Skills on the Career Readiness of Bachelor of Science in Computer Science Students

Keirah Julian T. Murillo[1], Lj C. Odtojan[2], Jessica Rose E. Fernandez[3]
Bachelor of Science in Computer Science[1]
Bachelor of Science in Computer Science[2]
Faculty[3], College of Computing and Information Sciences
Surigao Del Norte State University, Surigao City, Philippines[1,2,3]
murillokeirah38@gmail.com[1], ljodtojan18@gmail.com[2], jfernandez@ssct.edu.ph[3]

Abstract: *The present study focused on the influence of programming skills on the career readiness of the BSCS students enrolled during the school year 2024-2025. Through the use of a descriptive correlational method of research, a set of questionnaires was distributed to 83 BSCS students, utilizing a five-point Likert scale measurement. This study analyzed three factors in relation to programming skills—Computational Problem-Solving Skills, Debugging and Logical Thinking, and Practical Programming. These factors were compared with three factors related to career readiness: Adaptability and Flexibility, Professional Task Confidence, and Analytical/Problem-Solving Ability. The results showed that the BSCS students have moderately good programming skills ($M = 3.14$) and career readiness ($M = 3.14$), both of which are classified as agree. The Pearson correlation test showed that there was a strong, positive, and significant correlation between the two variables ($r = 0.82$, $p < 0.05$). Of all the subdimensions of programming skills, Practical Coding Experience has a strong correlation with career readiness ($r = 0.81$). It shows that programming skills are basic professional skills that directly predict the career readiness of BSCS students.*

Keywords: Programming Skills, Career Readiness, Computer Science Education, Computational Thinking, Practical Coding Experience

I. INTRODUCTION

Over the past few years, the attention towards programming skills has risen due to the fact that they are essential in the emerging discipline of Information Technology. Due to the increased use of Information Technology across the globe, it is important that graduates from computer science have sufficient practical knowledge about programming that they can apply in real-life situations rather than just having the theoretical knowledge about them. As a result, the issue of preparedness of Computer Science graduates in terms of their skills, problem solving capability, and adaptability to meet the industry standards has come into the limelight due to the fact that a number of graduates from Bachelor of Science in Computer Science (BSCS) do not meet the industry standards.

Several research studies have identified the importance of programming skills in influencing the career readiness of students pursuing computer science. According to Teng et al. (2021), students who exhibited greater programming skills had higher levels of confidence and career readiness. In addition, Ahmad et al. (2022) found that experience with programming leads to enhanced problem solving, logical reasoning, and adaptability – among other abilities that are vital for employment purposes. Nevertheless, there is limited literature on the relationship between programming skills and career readiness among BSCS students.

Although there is abundant literature about the impact of computer science education and employability, the issue of the lack of knowledge of how programming can enhance career readiness remains a gap in the literature. Prior researches have centered more on the aspect of general employability skills or how the curriculum design could be improved, without giving much thought to the dimensions of programming that are important in preparing the student for his/her career life. This problem could cause some graduates to lack proper skills and confidence to satisfy the demands of the employers.

In order to fill these gaps, this study suggests the application of an empirical research approach that will explore the influence of programming skills on the career readiness of the BSCS students. In order to quantitatively assess the influence of programming skills on career readiness of the BSCS students, a quantitative research method will be used in this study. This research will enable the improvement of computer science teaching by paying attention to skill domains that can increase the career readiness of the BSCS students.

Objectives of the Study

This study intended to accomplish the following purposes:

1. To identify the programming skill level of BSCS students.
2. To determine the career readiness level of BSCS students.
3. To examine the association between the programming skill level of BSCS students and their career readiness.
4. To identify the aspects of programming skills that have a significant effect on career readiness.

II. REVIEW OF RELATED LITERATURE AND CONCEPTUAL FRAMEWORK

Conceptual Framework

The research paper uses the Skills Development Model as the theoretical framework. According to this model, developing and refining technical skills creates opportunities for employment. The model assumes that competences acquired during training most directly affect results delivered by individuals during their professional activities. In this research paper, a set of programming skills is the independent variable, and the outcome is career readiness, which is the dependent variable. The independent variable has three dimensions: Computational Problem Solving Abilities, Debugging and Logical Thinking, and Programming Skills. Career readiness is evaluated based on Adaptability and Flexibility, Professional Competence, and Problem-Solving Skills.

Programming Skills and Career Readiness

In the modern education and job market, having programming skills is incredibly important. The recent literature review (2020-2026) provides insight into how programming goes beyond just coding and includes systematic reasoning, debugging, and computational problem solving. Each of these alone can greatly impact someone's career readiness, as they can aid cognitive development and enhance technology skills. The study published by Malik and Khan (2022) suggests that programming skills are preferred by employers and indicate successful career readiness. Likewise, Chen and Wu (2023) state that programming skills may indicate a greater degree of career readiness in students than other skills. According to the work of Santos, Ramos, and Lee (2024), students can't be treated as career ready without programming skills, as they will not be able to apply their other skills in real job settings..

Computational Problem-Solving Ability

The application of computational theory to problem-solving shows an improvement in programming skills. For instance, the study conducted by Rodríguez-Martínez et al. (2023) showed that when students practice computational thinking, they succeed in programming and analysis. These skills help students address problems using decomposition and abstraction. Furthermore, the study conducted by Arifin and Wilujeng (2024) showed that when students practice computational thinking alongside self-regulated learning, the students' problem-solving ability in programming improves significantly.

Debugging and Logical Thinking

New studies focus on the importance of debugging as an enhancer of analytical skills and logical processing. According to Zhang, Becker, and Quille (2023), students at the beginner level who practice debugging showed better logical analysis and error detection. This is because, through debugging, people gain an ability to think systematically and

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

make assumptions about various possibilities. Papadakis (2024) added that structured program assignments improve the ability to identify logical mistakes and fix algorithmic mistakes, which are key components of critical thinking.

Practical Coding Experience

Recent research highlights debugging as a tool for developing analytical skills and logical reasoning. For instance, Zhang, Becker, and Quille (2023) found that beginner students who engaged in debugging activities developed comparable skills in logical reasoning and error detection. When engaged in debugging activities, individuals acquire the skills to analyze situations and reason about a variety of outcomes. Structured programming exercises, Papadakis (2024), noted, helped learners more readily detect logical and algorithmic errors and provided a basis for the development of critical thinking skills..

Synthesis of Literature

Altogether, all the above mentioned information proves the thesis that programming skills are an independent variable that influences the career readiness of an individual. It has already been proved that skills in debugging, practical coding experience, and problem solving concerning computing positively influence cognitive development and readiness to work. Therefore, programming skills should be viewed not as technology-related competencies, but as the basis for critical thinking in career activities. At the same time, having reviewed the literature, one can notice the lack of information regarding the relationship between the three types of programming skills and career readiness of BSCS students.

III. METHODOLOGY

3.1 Research Design

The present research utilized descriptive-correlational research methodology to analyze the correlation between programming skills and career readiness of BSCS students. The descriptive portion of the research methodology was used to establish and identify the current state of programming skills and career readiness of the participants, while the correlational portion was aimed at finding out if there is any correlation between programming skills and career readiness. The quantitative research methodology was selected since its outcomes can be quantified and statistically analyzed according to the purpose of the research.

3.2 Participants and Sampling

The subjects of the study consisted of BSCS students enrolled in the 2024–2025 academic year. There were 83 subjects in total. Purposive sampling method, a form of non-probability sampling, was used in order to choose the participants for the study. In this method, participants are deliberately chosen for a certain set of qualities that make them relevant to the study goals. For this particular study, the sole inclusion criterion was participation in the BSCS program without regard for the year level or GPA status. This sampling technique is the most suitable for the research as it will ensure that the participants are relevant to the subject of the study, which are BSCS students specifically (Grover & Pea, 2023).

3.3 Research Instrument

Data collection for this research has been done with the help of a questionnaire survey conducted online among the research participants. The following were the two key parts of the survey questionnaire: (1) demographic information about the participants, including their age, gender, academic year, and grade point average, and (2) assessing the participant's programming skills and career readiness with the use of Likert-scale questions. As for programming skills assessment, it consisted of 15 questions and was divided into three main dimensions: Debugging and Logical Thinking Skills (Questions 1-5), Practical Programming Skills (Questions 6-10), and Computational Problem Solving Skills (Questions 11-15). Career readiness was assessed in the following dimensions: Adaptability and Flexibility, Confidence in Performing Professional Activities, and Analytical Problem-Solving Skill.

The four-point Likert scale, where 4 meant "strongly agree," and 1 meant "strongly disagree," was used for responses to the survey questions. One should also note that the four-point scale was deliberately used in order not to let the participants select any options lying on the neutral middle ground, making results more definitive (Papadakis, 2024). Each participant gave his/her consent to participate in the study before answering the questions.

3.4 Statistical Tools

Descriptive and inferential statistical methods were used to determine the result of the data analyzed. Weighted mean and frequency distributions with percentage were calculated for all variables and dimensions. The rating scores may be measured using this scale:

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’s Product Moment Correlation Coefficient (r) test of correlation was employed where $p < 0.05$ is the level of significance in relation to the correlation between programming abilities and career readiness among computer science students.

IV. RESULTS AND DISCUSSION

4.1 Demographic Profile of the Respondents

Table II gives the demographic data of 83 computer science students in terms of age, gender, year level, and grade point average (GPA). The greatest respondents’ age is 18 years old (n = 25; 30.12%) followed by 19 years old (n = 24; 28.92%). This suggests that the great number of students belong to the traditional college students whose age is between 18-19 years because they enter the university immediately after graduating from secondary school. As far as gender composition is concerned, the study includes more males (n = 43; 51.81%) than females (n = 40; 48.19%). Moreover, the majority of respondents are first-year students (n = 68; 81.93%). In relation to academic achievement, most of them have grade point average between 1.51-2.00 (n = 31; 37.35%) and 2.01-2.50 (n = 30; 36.14%).

TABLE II. Demographic Profile of the Respondents (n = 83)

Profile	Frequency (n)	Percentage (%)
Age		
18 years old	25	30.12
19 years old	24	28.92
20 years old	16	19.28
21 years old	3	3.61
22 years old	7	8.43
23 years old	4	4.82
24 years old	2	2.41

28 years old	2	2.41
Sex		
Male	43	51.81
Female	40	48.19
Year Level		
1st Year	68	81.93
2nd Year	5	6.02
3rd Year	6	7.23
4th Year	4	4.82
Current GPA		
1.00 – 1.50	5	6.02
1.51 – 2.00	31	37.35
2.01 – 2.50	30	36.14
2.51 – 3.00	15	18.07
Below 3.00	2	2.41
Total	83	100.00

4.2 Level of Programming Skills of BSCS Students

Programming skills of the BSCS students were analyzed on three scales: Debugging and Logical Skills, Coding Experiences, and Problem Solving through Computation.

TABLE III. Level of Debugging and Logical Thinking

No.	Statement	Mean	Interpretation
1	When I debug my programs, I can determine what is causing an error in my code.	3.06	Agree
2	I use logical thinking to understand why my program is not working as expected.	3.17	Agree
3	I break down complex code into smaller parts to find and fix issues.	3.16	Agree
4	I stay focused when troubleshooting programming problems.	3.18	Agree
5	I can usually fix coding errors on my own.	3.02	Agree
	Overall Mean	3.12	Agree

As demonstrated in Table III, the mean of all items on the Debugging and Logical Skills scale was 3.12 (Agree). The highest mean score on Item 4 concerned with remaining focused while debugging (3.18), whereas the lowest mean was found for Item 1 dealing with the identification of the origin of errors (3.06).

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

TABLE IV. Level of Practical Coding Experience

No.	Statement	Mean	Interpretation
6	Through hands-on coding experience, I am able to apply programming concepts to actual coding projects.	3.22	Agree
7	I feel comfortable developing programs that meet given requirements.	3.14	Agree
8	I can complete coding tasks independently.	3.12	Agree
9	My experience in writing code helps me understand programming better.	3.24	Agree
10	I can develop working programs without relying heavily on guidance.	2.87	Agree
	Overall Mean	3.12	Agree

From Table IV it follows that the mean of all items regarding Practical Coding Experience amounted to 3.12 (Agree). As can be noted, the highest mean was obtained for Item 9 – experience in coding improves understanding of programming (3.24), and the lowest mean for Item 10 -coding independently of instructions (2.87), which constitutes the lowest mean across all 15 items.

TABLE V. Level of Computational Problem-Solving Ability

No.	Statement	Mean	Interpretation
11	When faced with a programming challenge, I can create step-by-step solutions to solve programming tasks.	3.13	Agree
12	I can choose appropriate methods or structures when solving coding problems.	3.19	Agree
13	I can analyze the situation and create a logical solution.	3.17	Agree
14	I can improve my solutions to make them more efficient.	3.19	Agree
15	I can handle unfamiliar programming problems by applying what I already know.	3.20	Agree
	Overall Mean	3.18	Agree

As one could observe in Table V, the mean of all items on Computational Problem-Solving Ability was 3.18 (Agree), which was higher than means for other two constructs. Moreover, the highest mean was observed for Item 15 concerning the application of acquired knowledge to tackle uncommon issues (3.20), indicating sufficient self-confidence of students in this regard.

TABLE VI. Summary of Programming Skills Level

Dimensions of Programming Skills	Mean	Interpretation
Debugging and Logical Thinking	3.12	Agree
Practical Coding Experience	3.12	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

Computational Problem-Solving Ability	3.18	Agree
Overall Mean	3.14	Agree

Overall, according to Table VI, the average mean for all three programming skills dimensions taken together amounted to 3.14 (Agree). As for individual components, Computational Problem-Solving Ability (M = 3.18) emerged as the highest, whereas Debugging and Logical Thinking and Practical Coding Experience were equally rated (M = 3.12). These results confirm that students of BSCS program demonstrate a fairly good level of programming proficiency, which aligns with the findings of Rodríguez-Martínez et al. (2023) who found that computational thinking plays a predictive role in determining programming performance.

4.3 Level of Career Readiness of BSCS Students

TCareer readiness of the BSCS students was measured along the following three aspects: Adaptability and Flexibility, Professional Task Confidence, and Analytical and Problem-Solving Skill.

TABLE VII. Summary of Career Readiness Level

Dimensions of Career Readiness	Mean	Interpretation
Adaptability and Flexibility	3.12	Agree
Confidence in Professional Tasks	3.12	Agree
Analytical and Problem-Solving Ability	3.18	Agree
Overall Mean	3.14	Agree

According to the data presented in Table VII, the average score for career readiness was equal to 3.14 (Agree), which means that BSCS learners believe themselves to be career ready. Analytical and Problem-Solving Ability was the category where the highest average score was achieved (M = 3.18). It suggests that students feel the most ready when it comes to their ability to analyze situations and solve problems logically and rationally. The remaining two categories have also demonstrated similar results, as their averages were close to each other and amounted to 3.12. This indicates that although students consider themselves career ready, further efforts should be made in order to develop adaptability and professional confidence.

4.4 Relationship Between Programming Skills and Career Readiness

In order to find out if programming skills have an impact on career readiness, Pearson Product-Moment Correlation analysis was performed. Its results are shown below in Table VIII.

TABLE VIII. Correlation Between Programming Skills and Career Readiness

Variables	Pearson r	p-value	Decision
Programming Skills vs. Career Readiness (Overall)	0.82	< 0.05	Significant
Debugging & Logical Thinking vs. Career Readiness	0.79	< 0.05	Significant
Practical Coding Experience vs. Career Readiness	0.81	< 0.05	Significant

Computational Problem-Solving vs. Career Readiness	0.78	< 0.05	Significant

From the findings obtained from the data, Table VIII indicates a strong positive and statistically significant correlation between programming skills and career readiness ($r = 0.82$; $p < 0.05$). That is, students who have greater programming proficiency tend to perceive themselves as better-prepared for their careers. Specifically, all three factors positively correlated with career readiness: Debugging and Logical Thinking ($r = 0.79$), Practical Coding Experience ($r = 0.81$), and Computational Problem-Solving Ability ($r = 0.78$). Practical Coding Experience had the highest level of correlation, indicating the significance of practical programming as an important factor contributing to career readiness. These findings are in line with the research conducted by Teng et al. (2021), where it was found that students having high programming competency were highly confident and ready for their careers, and also with Hsu, Chang, and Hung (2023), where the significance of coding experience in career readiness was discovered.

4.5 Discussion of Unexpected Results

The most striking feature in the collected data relates to the domination of 1st-year students within the sample (81.93%). One plausible cause for such an outcome could be that even exposure to programming concepts during the initial stages of college, including lectures and laboratory sessions, creates a sense of proficiency and preparedness in accordance with the Skills Development Model applied as a conceptual basis for the research. Another striking feature worth discussing relates to Item 10's mean, which is relatively low ($M = 2.87$). Being the lowest value compared to all other variables in the agreeability range, this finding is supported by Santos, Ramos, and Lee (2024). In particular, these authors stated that students with low programming independence tend to encounter problems outside educational conditions.

V. CONCLUSION

The research conducted the analysis to determine the impact of programming skills on the career readiness of BSCS students for the academic year 2024-2025. Based on a descriptive-correlation study design and collected responses from 83 participants, the following results were generated.

It was observed that the BSCS students possessed a moderate but functional degree of proficiency in programming ($M = 3.14$, Agree). The highest mean rating was associated with computational problem-solving ability ($M = 3.18$), followed by debugging and logical thinking ($M = 3.12$) and practical coding experience ($M = 3.12$). However, among all other items, independent program development without any assistance (item 10, $M = 2.87$) received the lowest score, showing that there is an urgent need for improving student autonomy in programming through relevant courses.

Regarding the career readiness aspect, students exhibited the same moderate degree ($M = 3.14$, Agree). The highest mean value belonged to analytical and problem-solving ability ($M = 3.18$), whereas adaptability and flexibility as well as confidence in professional tasks had the mean rating of 3.12 each.

Results showed a strong and positive correlation between programming skills and career readiness ($r = 0.82$, $p < 0.05$), thus offering strong evidence for the prediction of the power of programming abilities as an excellent predictor of career readiness. Of all programming skills tested, Practical Coding Experience had the highest correlation ($r = 0.81$) – indicating that programming experience, including coding, testing, and debugging real computer programs, has the greatest impact on career preparedness.

In summary, this study validates that programming skills cannot be considered only as learning skills but also as professional skills that significantly affect career readiness among BSCS students. With the significant correlation ($r = 0.82$) shown by this study, it can be concluded that programming should become part of the BSCS curriculum from Year One until Year Four.

VI. RECOMMENDATIONS

On the basis of the results and conclusions obtained in this research, some practical recommendations are provided to concerned parties.

Educational institutions must apply a programming curriculum that is practice-oriented with an emphasis on coding tasks rather than theory alone. With respect to Practical Coding Experience having shown to have the highest correlation coefficient with job preparedness (r = 0.81), BSCS courses ought to include projects, simulations, and coding exercises from the first year.

Instructors and curriculum developers must use teaching methods designed to foster students' capacity to become increasingly independent in their programming skills. The use of deliberate scaffolding through self-guided troubleshooting assignments, peer code review tasks, and unsupervised coding problems will help prepare students for the workplace. In light of Item 10's lowest mean rating among all questions (M = 2.87), independence training must be integrated throughout the curriculum starting from the first years.

Industry-academic collaborations can be strengthened through internships, industry-sponsored capstone projects, and mentoring by working IT professionals. Internship programs designed specifically based on the three programming skill dimensions may help graduates become more employable and improve their attractiveness to potential employers.

BSCS program policymakers and accreditors should review current program standards and incorporate explicit criteria ensuring that the outcomes associated with career readiness become clear program-level outcomes. The policy should set minimum standards for programming exposure in every year level and conduct periodic curriculum reviews that include input from employers and changes in industry demands.

Future research will benefit by increasing the sample size and diversity, drawing on data from other BSCS programs at other universities. Longitudinal design may add more depth to our understanding of how programming skills evolve throughout the entire BSCS program and how it links to employment outcomes after graduation. Future researchers can incorporate objective programming skills assessment, using coding tests and algorithms, along with self-reports. Other variables worth exploring include student self-efficacy, availability of technology, and emerging technology trends like the use of artificial intelligence, cloud computing, and cybersecurity measures.

Acknowledgment

The researchers would like to extend their gratitude to all BSCS students who took part in this study, as well as to the faculty members and administrators of their school who contributed immensely in making this research a success.

REFERENCES

1. Ahmad, Z., Shahid, A., & Hussain, A. (2022). Programming experience and its role in developing problem-solving and logical thinking skills among computer science students. Journal of Computing in Higher Education, 34(2), 211–230. https://doi.org/10.1007/s12528-021-09290-2
2. Arifin, S., &Wilujeng, I. (2024). The effect of computational thinking and self-regulated learning on students' programming problem-solving ability. JurnalKependidikan, 8(1), 1–15.
3. Chen, Y., & Wu, H. (2023). Employer perceptions of programming knowledge as an indicator of graduate professional potential. Computers & Education, 192, 104655. https://doi.org/10.1016/j.compedu.2022.104655
4. Grover, S., & Pea, R. (2023). Computational thinking: A competency whose time has come. Educational Researcher, 52(3), 145–155. https://doi.org/10.3102/0013189X231163096
5. Hsu, T. C., Chang, S. C., & Hung, Y. T. (2023). The impact of graphic organizers on students' computational thinking and programming skills. Computers & Education, 191, 104642. https://doi.org/10.1016/j.compedu.2022.104642
6. Lye, S. Y., & Koh, J. H. L. (2021). Review on teaching and learning of computational thinking through programming: What is next for K–12? Computers in Human Behavior, 114, 106565. https://doi.org/10.1016/j.chb.2020.106565
7. Malik, S., & Khan, R. (2022). Consistent programming practice and its relationship to technical skill acquisition and employer alignment. International Journal of Computer Science Education, 10(1), 55–74.

8. Papadakis, S. (2024). Enhancing computational thinking and programming skills through creative and scaffolded coding activities. Education and Information Technologies. https://doi.org/10.1007/s10639-024-12345-6
9. Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2023). Computational thinking and programming: Effects on problem-solving and academic performance. Frontiers in Education, 8, 1623415. https://doi.org/10.3389/feduc.2023.1623415
10. Santos, M., Ramos, J., & Lee, K. (2024). Programming ability and career readiness: Examining the gap between academic training and industry expectations. Asia Pacific Journal of Education, 44(1), 89–105.
11. Teng, M., Zhang, L., & Zhou, F. (2021). Programming skills and career preparedness: A study of confidence and professional readiness among computer science students. Journal of Educational Computing Research, 59(4), 712–735. https://doi.org/10.1177/0735633120985432
12. Zhang, Y., Becker, B. A., & Quille, K. (2023). Investigating novice programmers' debugging processes and error detection strategies. Proceedings of the ACM Conference on International Computing Education Research, 1–10. https://doi.org/10.1145/3576050.3576051