WIP: How to Improve Student Comprehension of Pseudocode Reading and Writing

Anjum Chida
Department of Computer Science
Rice University
Houston Texas, U.S.A
anjum.chida@rice.edu

Ovidiu Daescu
Department of Computer Science
University of Texas at Dallas
Richardson Texas, U.S.A
daescu@utdallas.edu

Abstract*—This work in progress innovative practice paper describes a tentative study on Computer Science students' pseudocode comprehension, aiming to explore intermediate steps in their understanding of reading and writing pseudocodes. As problem-solving skills are integral to undergraduate studies, mastering pseudocode formulation becomes fundamental for these students. In advanced computer science courses, solutions are often presented in pseudocode format, requiring students to express their solutions likewise. Proficiency in comprehending and creating pseudocode becomes crucial. This study aims to evaluate students' aptitude in comprehending pseudocode from written sources and in devising solutions using pseudocode. The investigation is designed to be conducted in multiple phases, employing Think-Pair-Share and many other active learning strategies to enhance pseudocode understanding. This paper presents promising preliminary survey/quiz findings, which were conducted in an advanced algorithm course. The subsequent phases will delve deeper into students' metacognition regarding pseudocode reading and composition, aiming to gain a deeper understanding of learning processes and identify effective instructional strategies. In the realm of Computer Science education, a profound understanding of students' cognitive processes and cognitive frameworks becomes essential in refining instructional techniques and teaching strategies. The objective of this study is to gain a deeper understanding of student's learning processes and to investigate which actions by the instructors will help them the most with regard to pseudocode reading and writing. This encompasses a comprehensive development journey, including theoretical understanding, practical application, and hands-on experiences that collectively contribute to their adeptness in addressing complex problems and solutions within the field of Computer Science. We aim to examine how students understand problems and their solutions when presented in pseudocode format, particularly when they need to represent the solution as pseudocode instead of writing code in a high-level language. In Phase-I of the study, the objective is to assess the effectiveness of using an active learning strategy, such as Think-Pair-Share, in a classroom setting to teach pseudocode as opposed to traditional lecture-based methods. Phase-II of the research will further explore students' metacognitive processes during their engagement with pseudocode, focusing specifically on problem-solving contexts.

Keywords—Pseudocode Comprehension, Active Learning, Think-Pair-Share, Metacognition.

I. INTRODUCTION

Computer science students undergo a continuous process of honing and acquiring essential problem-solving skills throughout their undergraduate degree. Problems are ubiquitous in our lives, and the field of computer science primarily focuses on classifying, discussing, resolving, and understanding the attributes of these problems and their corresponding solutions. Pseudocode serves as a systematic and logical method of representing solutions for certain mathematical problems. It enables students to tackle typical problems effectively; they often realize the importance of pseudocode comprehension when delving into other fields of study such as artificial intelligence, machine learning, cyber security, and more. Typically, some students appear to have mastered the skill of comprehending and formulating pseudocode but many understandably struggle with it [1].

Proficiency in problem-solving and the formulation of pseudocode solutions holds great significance for the success of Computer Science students [2]. While substantial research has been conducted to comprehend student program comprehension and their programming skills in specific languages [3][4], there is a dearth of educational experiments focused on assessing students' ability to comprehend and write pseudocode. This research holds notable importance in this context as it aims to shed light on students' aptitude for comprehending pseudocode from textual sources and their competence in devising original solutions. Although various studies have explored active learning strategies for problem-solving endeavors, these have not consistently addressed the reading or writing of pseudocode [5][6]. The focus here is more on pseudocode comprehension and writing, and less on general problem-solving techniques.

Research indicates that active learning contributes to enhanced long-term retention of course content among students especially when compared to traditional lectures [7-15]. Evidence suggests that computer science (CS) students often possess learning preferences that render an active learning environment crucial for their effectively mastering of the subject matter [14]. Further, the concept of problem-centered learning has a rich history in promoting education based on real-world encounters [16-19]. Engaging in problem-based learning, wherein students collaborate in groups and dedicate time to reflecting on acquired techniques and the efficacy of

employed strategies, aid students in gaining further insights and developing self-directed learning capabilities [20-24].

The main objective is to examine how students understand problems and their solutions when presented in pseudocode format, particularly when they need to represent the solution as pseudocode instead of writing code in a high-level language. Additionally, the study aims to explore how students initiate their solution formulation phase. The primary research inquiries addressed by this study are as follows:

1. Does active learning contribute to students' comprehension of reading pseudocode?
2. Does active learning contribute to students' comprehension of writing/formulating pseudocode?
3. How can instructors leverage a thorough analysis of students' metacognition concerning problem-solving and pseudocode-based solutions?

To address these inquiries, the study seeks to amass diverse data points through surveys, examination outcomes, and interviews. Positioned more as an exploratory endeavor, the study's design not only seeks to validate hypotheses but also aims to foster the generation of novel hypotheses concerning students' understanding of pseudocode.

This paper presents initial findings from a study conducted through surveys administered before and after an active learning session in one advanced algorithm course. The active learning session, utilizing the Think-Pair-Share technique, spanned a single class period. Based on these preliminary results, it can be inferred that the subject holds significant importance in the realm of Computer Science education. This study stands out in its distinctive approach to investigating the impact of active learning strategies on pseudocode comprehension and formulation. The subsequent phases of the study aim to address inquiries about students' metacognitive capacities and their specific challenges in constructing solutions. There are only a limited number of studies that directly tackle the inquiries posed by this study.

II. Study Design

The target demographic for this study comprises of undergraduate students majoring in Computer Science. The aim is to investigate both novice and intermediate learners. Thus, our attention will be directed towards juniors and sophomores, specifically those enrolled in introductory programming and advanced-level algorithm design and analysis courses.

In Phase-I of the study, the main aim is to measure the students' comprehension and confidence levels before and after the active learning session. The survey/quiz results will aid in the development of enhanced activities and scenarios for future semesters. In the same phase, the exam scores of students on pseudocode questions will be compared with those from the previous semester to assess any statistically significant improvements. Phase-I will be conducted for both first year students and sophomores in their "Intro to Java Programming" course, as well as for junior and senior students in their advanced algorithm course.

In Phase-II, the objective is to conduct interviews with students at different proficiency levels and stages in their degree program to gain insights into how they construct their pseudocode solutions, identify areas where they encounter difficulties, and understand the most challenging aspects. The teaching assistants who will be conducting the interviews are graduate students, who actively serve as TAs for the advanced-level courses. Equally important, training these TAs will form a vital component of the study.

The results from Phase-I will aid in formulating the interview questions. The interviews will be conducted by multiple teaching assistants to be hired from a pool of graduate students. To ensure consistency and adherence to the research protocol, these TAs will receive training before conducting the interviews. The hypothesis of Phase-I of the study is that the active learning technique will enhance students' ability to comprehend and formulate pseudocode. Multiple surveys and quizzes will be conducted to investigate and establish whether the active learning approach indeed leads to improved proficiency in reading and writing pseudocode among the students. Some preliminary results from this phase are discussed in the next section.

The second phase does not have a specific hypothesis; instead, it aims to identify areas where students' may face challenges in formulating pseudocode solutions. This endeavor may aid us in improving course and curriculum development for future students. The research aims to explore the potential of narrative inquiry and connected learning approaches in enriching the learning experiences of undergraduate Computer Science students, particularly when it comes to problem-solving in different courses throughout their academic journey.

III. Preliminary Survey/Quiz Results

This section examines the survey results and the outcomes of a single active learning session conducted in the fall of 2023. The survey was administered as a class quiz to ensure maximum participation; hence it is referred to as a survey/quiz in this paper. The presented findings are derived from an advanced algorithm course (usually taken by sophomores and juniors) at our institution, which included 78 students. Despite the advanced nature of these students' understanding of pseudocode and algorithmic concepts, they reported significant benefits from the Think-Pair-Share active learning session. These results highlight that even students with a high proficiency in problem-solving and pseudocode design still find substantial advantages in participating in active learning sessions.

These results are from a specific active learning session lasting 75 minutes, where students were tasked with writing pseudocode for a practical problem: *inserting an element into a sorted array*. The aim was to assess the impact of this approach on students' pseudocode proficiency and explore the depth of understanding achieved through collaborative problem-solving and subsequent discussions. The active learning session commenced with students individually solving the pseudocode problem within a stipulated time of 15 minutes. Following this, they engaged in a 15-minute

TABLE I. RESULTS FROM SURVEY/QUIZ BEFORE PSEUDOCODE INTERVENTION CLASS

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I receive full credit for my pseudocode solutions to given problems.	17.14%	58.57%	15.71%	4.29%	0.00%
I can successfully perform a dry run of the pseudocode provided in a textbook.	11.43%	48.57%	28.57%	10.00%	0.00%
Do you believe that pseudocode is a helpful tool for problem-solving and programming?	4.29%	32.86%	34.29%	27.14%	0.00%

collaborative discussion with a partner, referencing the pseudocode evaluation rubric used by the instructor for assessing assignments and exams for the same course. The session culminated in randomly selected students presenting their solutions to the class. The ensuing discussion encompassed aspects of algorithmic efficiency, the correctness of individual solutions, and the exploration of diverse problem-solving approaches. We administered surveys/quizzes both before and after the Pseudocode Intervention class session. Table I and Table II show the questions and responses from the pre-Pseudocode Intervention survey/quiz, while Table III and Table IV present the questions and results from the post-Pseudocode Intervention survey/quiz.

TABLE II. RESULTS FROM SURVEY/QUIZ BEFORE PSEUDOCODE INTERVENTION CLASS

Questions	All The Time	Some of the Time	Never
I receive full credit for my pseudocode solutions to given problems.	17.14%	78.57%	2.86%
I can successfully perform a dry run of the pseudocode provided in a textbook.	30.00%	62.86%	5.71%
Do you believe that pseudocode is a helpful tool for problem-solving and programming?	88.57%	2.86%	7.14%

The first survey/quiz, conducted before the Pseudocode Intervention class, included 12 questions (8 opinion scale/Likert questions and 4 essay questions) and students had a week to complete it. The second survey/quiz, conducted after the class, consisted of 5 opinion scale/Likert questions, 1 multiple-choice question, and 2 essay questions, with the same one-week completion period. We have not analyzed the essay questions in this study and will address them in future papers. The first survey/quiz had 71 participants, while the second had 75 out of a total of 77 students. The questions are detailed in the corresponding tables along with the student participation results.

The survey/quiz results from Table-I show that 75.71% of students understand pseudocode solutions from textbooks, and 60.00% feel confident in writing pseudocode for problems. However, 38.57% express a neutral or disagreeing stance on their confidence. Additionally, 37.14% find formulating pseudocode solutions not usually difficult, while 61.43% express a neutral or disagreeing perspective on the difficulty level. These findings suggest varying levels of confidence and perceived difficulty among students in understanding and formulating pseudocode solutions.

TABLE III. RESULTS FROM SURVEY/QUIZ AFTER PSEUDOCODE INTERVENTION CLASS

Question: - Which aspect of the session was most helpful to you? Select all that applies.	Percentage Selected
The given problems	40.00%
Working with a partner	72.86%
The learning pace	35.71%
The lectures before and after the active learning sessions	14.29%
None of the Above	1.37%

The survey/quiz results from Table-II reveal insights into students' perspectives on pseudocode-related experiences and beliefs. Regarding the reception of full credit for pseudocode solutions, a substantial 78.57% report receiving such credit only some of the time, indicating potential areas for improvement in the evaluation process. In terms of performing dry runs of provided pseudocode, 62.86% can successfully do so only some of the times, suggesting challenges or areas for enhancement in understanding and applying pseudocode. However, there is a strong consensus (88.57%) among students that pseudocode is a helpful tool for problem-solving and programming. In summary, the results suggest that while students may face challenges in receiving full credit for pseudocode solutions and performing dry runs, there is a strong consensus among them regarding the usefulness of pseudocode as a tool for problem-solving and programming.

TABLE IV. RESULTS FROM SURVEY/QUIZ AFTER PSEUDOCODE INTERVENTION CLASS

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The Pseudocode Intervention increased my ability to understand the problem while reading pseudocode solutions from a textbook.	17.81%	63.01%	10.96%	4.11%	1.37%
The Pseudocode Intervention increased my ability to understand the solution itself in terms of pseudocode while reading from a textbook.	20.55%	58.90%	15.07%	2.74%	1.37%
The Pseudocode Intervention increased my ability to understand the problem while trying to write a pseudocode solution for the problem.	27.40%	53.42%	15.07%	1.37%	1.37%
The Pseudocode Intervention increased my ability to formulate a pseudocode solution to a given problem.	23.29%	54.79%	13.70%	5.48%	1.37%
The Pseudocode Intervention increased my ability to run a dry run of the pseudocode given in a textbook	26.03%	47.95%	19.18%	1.37%	2.74%

Table-III shows responses to the question regarding the most helpful aspects of the active learning class period. A significant 72.86% of students identified working with a partner as the primary contributor to the session's effectiveness, emphasizing a preference for collaborative learning. Additionally, 40.00% found solving given problems to be beneficial, highlighting the importance of practical problem-solving activities. The learning pace was deemed helpful by 35.71% of students, indicating an appreciation for the speed or rhythm of the session. A smaller percentage (14.29%) pointed to the lectures before and after the session as their most helpful aspect, suggesting varied preferences in learning approaches. Overall, these findings underscore the diverse perspectives on effective learning strategies within the student cohort.

The survey/quiz results from Table- IV indicate positive outcomes from the Pseudocode Intervention across various aspects of students' abilities. About 80.82% (17.81% strongly agree, 63.01% agree) believe the intervention increased their ability to understand problems while reading pseudocode solutions from a textbook. For the question about understanding pseudocode solutions from textbooks, 79.45% (20.55% strongly agree, 58.90% agree) report an enhanced ability to understand pseudocode solutions when reading from a textbook. Regarding the question about understanding a particular problem, 80.82% (27.40% strongly agree, 53.42% agree) feel the intervention improved their ability to understand problems while attempting to write pseudocode solutions. In response to the question about formulating a pseudocode solution, 78.08% (23.29% strongly agree, 54.79% agree) acknowledge an increased ability to formulate pseudocode solutions for given problems. Finally, for the question about dry run, 74.32% (26.03% strongly agree, 47.95% agree) state that the intervention enhanced their ability to perform a dry run of pseudocode provided in a textbook. Overall, a majority of students express positive sentiments about the beneficial impact of the Pseudocode Intervention session.

We recognize that the data presented reflect students' opinions on their experience on a single active learning session. We also acknowledge the challenge of assessing the effectiveness of active learning activities like Think-Pair-Share without a control group for comparison. We will address these issues in the subsequent papers of the study. Nonetheless, we remain optimistic about the findings and are committed to continuing this investigation to meet the objectives outlined in the introduction.

IV. ETHICAL DATA COLLECTION

The research aims to prioritize avoiding significant ethical implications and minimizing risks to individuals involved. The data collected is devoid of personal or other identifiable private information, ensuring no identifiable data points are revealed and no substantial breaches of data privacy occur. This practice will continue in future phases, and any mention of grades will be presented only in a collective manner. The data gathered for the survey/quiz discussed in this paper was acquired after securing Institutional Review Board (IRB) exempt approval from our institution's IRB board. Each participating student had to sign a consent form to authorize the inclusion of their survey data in this report.

V. CONCLUSION

This research underscores the positive impact of active learning on students' pseudocode proficiency. The collaborative Think-Pait-Share session not only enhanced individual understanding but also fostered a collective exploration of efficient algorithms and diverse solution approaches. The findings contribute to the ongoing discourse on effective pedagogical strategies in computer science education, emphasizing the value of active learning in cultivating algorithmic proficiency among students.

In conclusion, the preliminary survey/quiz results are highly promising, suggesting that students, despite their confidence in understanding and creating pseudocode in the algorithm class, found the active learning and in-depth discussions on pseudocode solutions remarkably beneficial. In this context, we have effectively addressed the initial two questions outlined in the introduction of this paper. While the interpretability of the results might be debated, they generally indicate a positive trend. The Pseudocode Intervention section was particularly well-received, with students

expressing that it significantly contributed to their understanding of problems and the creation of pseudocode solutions.

This study could offer numerous advantages and benefits to students, other prospective learners, and educators in the field of Computer Science. The goal of this study is to disseminate our findings and methodology within the Department of Computer Science at our institution and throughout the university. This effort aims to serve as an educational resource for fellow instructors involved in similar research pursuits.

ACKNOWLEDGMENT

We would like to acknowledge the Center for Teaching and Learning at University of Texas at Dallas, especially Dr. Karen Huxtable-Jester. Additionally, we extend our gratitude to Phat Nguyen from the CS department for supporting this project. Our heartfelt thanks to Gloria Shenoy for helping formulate the survey/quiz questions.

REFERENCES

[1] L. Winslow, "Programming pedagogy- a psychological overview", ACM Sigcse Bulletin 28(3), 17-22, 1996.

[2] S. Tasneem, "Critical thinking in an inductory programming course", Journal of Computing Science in Colleges, 27(6), 81-83, 2012.

[3] A. L. Olsen, "Using pseudocode to teach problem-solving", Journal of Computing Science in Colleges, 21(2), 231-236, 2005.

[4] S. Peltsverger & S. Debnath, "Instructional Pseudocode Guide to Teach Problem-Solving", ITiCSE 2019.

[5] M. Jalalitabar & Y. Wang, "Demystifying the Abstractness: Teaching Programming Concepts with Visualization", ACM SIGITE, 2022.

[6] U. Obaidellah, T. Blascheck, D. T. Guarnera & J. I. Maletic, "A Fine-grained Assessment on Noice Programmers' Gaze Patterns on Pseudocode Problems", Proceeding of the ACM Symposium on the Eye Tracking Research and Application (ETRA), Article No. 56, 1-5, 2020.

[7] B. Yulianto, H. Prabowo, R. Kosala & M. Hapsara, "Novice Programmer = (Sourcecode) (Pseudocode) Algorithm", Journal of Computer Science, 14(4), 477-484. 2018.

[8] C. Gao & B. Goda, "Student Centered Learning Create a Significant Learning Experience by Using Flipped Classroom Approach", ACM SIGITE, 2019.

[9] G. G. Roy, "Designing and Explaining Programs with a Literate Pseudocode", ACM Journal of Educational Resource in Computing, Vol. 6, No. 1 Article 1. 2006.

[10] A. Rosso and M. Daniele , " Our Method to Teach Algorithmic Development", ACM SIGCSE Bulletin vol.32, Issue No. 2, 2000.

[11] J.C. Giordano & M. Carlisle, "Toward a More Effective Visualization Tool to Teach Novice Programmers", ACM SIGITE'06, 2006.

[12] O. Hazzan and J. Kramer, "Abstraction in computer science & software engineering: A pedagogical perspective," Frontier J., vol. 4 no. 1 pp. 6-14, 2007.

[13] S. A. A. de Freitas , W. C. M. P. Silva and G. Marsicano, " Using an active learning environment to increase students' engagement", in Proc. IEEE 29th Int. Conf. Softw. Eng. Educ. Training, Dallas, TX , USA, 2016, pp 232-236.

[14] T. Briggs, "Techniques for active learning in CS courses", J. Comput. Sci. Colleges, vol 21, no. 2, pp. 156-165, Dec 2005.

[15] M. Prince, "Does active learning works? A review of the research", J. Eng. Educ. Vol. 93, pp. 232-231, Feb 2013.

[16] C. E. Hmelo-Silver, ''Problem-based learning: What and how do students learn?'' Educ. Psychol. Rev., vol. 16, no. 3, pp. 235–266, Sep. 2004.

[17] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, ''Active learning increases student performance in science, engineering, and mathematics,'' Proc. Nat. Acad. Sci. USA, vol. 111, no. 23, pp. 8410–8415, Jun. 2014. doi: 10.1073/pnas. 1319030111.

[18] J. Michael, ''Where's the evidence that active learning works?'' Adv. Physiol. Educ., vol. 30, no. 4, pp. 159–167, Dec. 2006. doi: 10.1152/advan. 00053.2006.

[19] J. Knobloch, J. Kaltenbach, and B. Bruegge, ''Increasing student engagement in higher education using a context-aware Q&A teaching framework,'' in Proc. 40th ICSE-SEET, Gothenburg, Sweden, 2018, pp. 136–145.

[20] C. C. Bonwell and J. A. Eison. Active Learning: Creating Excitement in the Classroom. ASHE-ERIC Higher Education Report No. 1., Washington, D.C.: The George Washington University, School of Education and Human Development, 1991.

[21] C. Meyers and T. B. Jones. Promoting Active Learning: Strategies for the College Classroom. Jossey-Bass Inc., Publishers, 350 Sansome Street, San Francisco, CA 94104, 1993.

[22] M. Prince. Does active learning work? a review of the research. Journal of Engineering Education, 93(3), 2004.

[23] T. Tang, S. Rixner, and J. Warren. An environment for learning interactive programming. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, 2014.

[24] T. Tang, R. Smith, S. Rixner, and J. Warren. Data-driven test case generation for automated programming assessment. In Proceedings of the 21st Annual ACM Conference on Innovation and Technology in Computer Science Education, 2016.