Journal of Engineering Education Transformations,
Volume No 36, January 2023, Special issue, eISSN 2394-1707

10.16920/jeet/2023/v36is2/23077

A Mini-Ethnography Study of Teaching C Programming to First Year Non-Circuit Branch Students

Prakash Hegade, Aryan, Omkar Patil, Manjunath Netekal and Ashok Shettar
School of Computer Science and Engineering,
KLE Technological University, Karnataka, Hubli

***Abstract*—Problem based learning, finding its root from the medical practices is now a widely used approach in several other domains. The method has also been used in computing and explored and researched with the programming courses as well. This work proposes one such model for first year c programming. A variant of the model was adapted to first design problem scenarios to teach the concept philosophy and then explain the underlying concept with c realization and its syntax. The students coming from non-circuit branches and having lower intrinsic motivation to study the course have shown gradual interest and motivation towards the much-needed 21st century skill. On finding that they lacked the debugging skills, even the syntax was explained with design philosophy and problem based learning. Understanding the class behavior being the research question, the work was analyzed using the mini-ethnography qualitative study. The study was carried out for a month span with seven sessions. The data was collected using observations and forms, and also thematic coding was carried out for the data analysis. The discussion section presents the analysis of the model proving the method to be an effective methodology for teaching and learning. Even the feedback collected for the model is positive and can be employed to teach other courses as well.**

***Keywords*—Mini-Ethnography; Problem Based Learning; Programming; Problem Solving**

***JEET Category*—Research**

I. INTRODUCTION

THE principles of teaching and learning can be understood in varying contexts and from the underlying pedagogy. The principles of education from the schooling system concentrates on objectives, need for the state-of-art, institutional design, characteristics of students, teaching and learning process and selection of materials for curriculum and activities (Thorndike & Gates, 1929). The quality of cognitive engagement has a direct impact on the process of learning (Hannafin & Hooper, 1993). Students need time and space to learn and they do not necessarily learn just because they were told to do so. Scaffolding is an important activity and part of learning (Reiser & Tabak, 2014).

If the learning has to be deep and transformative, then reflections are essential (Mezirow & Taylor, 2009). Some form of reflection oriented questions or triggers must be used to help provoke students reflections (Butler, 1987). The way and method by which students acquire knowledge is important as much as the knowledge they are acquiring (Butler & Nisan, 1986). Feedback from the facilitator must be formative, continuous timely, and individualized to help the students and they will be effective (Krampen, 1987). Assessments must be meaningful to students and students should be regularly assessed in a holistic manner (Kleinert et al., 2002). Experience has been one of the major ingredients that motivate one to learn in major studies of cognitive theories for learning and development (Kolb, 2014). There are several pedagogies that are developed over time that aim to provide experiential learning. Problem Based Learning (PBL) is one such pedagogy.

McMaster introduced tutorials of problem solving not as a method of instruction but as a way of structuring the entire curriculum so that students do not wait until the clinical work begins but make it as a way of learning that could be easily transferred to professional practice (Barrows & Kelson, 1996). PBL, a constructivist learning environment with collaborative problem solving and reflection experiences supports the student learning and collective knowledge building (Hmelo-Silver & Barrows, 2008). Open ended metacognitive questioning, pushing for explanation, re-voicing, summarizing, evaluating hypothesis and mapping, creating learning issues and many other specific strategies have been developed and employed helping students construct casual explanations, reason effectively, and become self-directed learners while maintaining a student-centered learning process (Hmelo-Silver & Barrows, 2006). Even computer supported collaborative techniques have been developed for the effective delivery of PBL (Koschmann et al., 1996). The various PBL models developed across the globe find a stimulus from the classical learning and cognitive theories. From classical brain development theories to experimentation in state-of-art pedagogies, they have had contributions in making PBL on what it is today. Experience, authentic problems are some the noticeable features of all these theories.

PBL has been experimented at various levels. From course to institute level, the methodology has its own significance and learning objectives. Being dependent on several factors,

it's a challenge to apply the model for the programs that run sessions of one to two hours with a classroom strength of 70 or more. Variants have been adapted to suite the demography and class requirements and this paper proposes one such method. PBL sessions were designed for classroom strength of 35 for a programming course which was delivered in a month's span. This paper presents the experimentation as a mini-ethnography study.

Section 2 presents the literature survey on PBL, its integration with programming and section 3 presents the context and the model designed along with research methodology. Section 4 presents the results and data analysis of the model and section 5 presents the discussion. Section 6 concludes the paper.

II. Literature Survey

This section presents the literature survey on the domains of PBL and c programming. Teaching programming, especially for non-circuit branch students is a challenging task. They lack intrinsic motivation as it's not part of their holistic program structure. Though lately programming has become inevitable part of almost all domains of degree programs, most students do not take it seriously. PBL, with research proven benefits can pave a pathway to awaken the intellectual excitement.

PBL has its principles structured around contextual learning and courses are designed accordingly. PBL helps to construct an atmosphere where students would want to learn, make decisions. In PBL method students become accountable for their learning. Designing assessment methods in PBL need substantial and continuing effort (Hillman, 2003). Recent studies have compared PBL with traditional using pre-test and post-test methods. Studies confirm that PBL can improve the mathematical problem solving capabilities and self-confidence (Hendriana et al., 2018). PBL has also been used as an effort to improve student learning outcomes (Malmia et al., 2019). Researches also conclude that the application of problem based learning has more positive effect than direct instruction in terms of critical thinking ability in solving mathematical problems (Umar et al., 2020). Programming is mathematics.

Several research efforts have been made in teaching programming. Foundations of programming have been deliberated (Mitchell, 1996). Semantics of C programming has been studied (Gurevich & Huggins, 1992). Program analysis and specializations have been carried out (Andersen, 1994). Educational ontologies for C programming have been developed (Sosnovsky & Gavrilova, 2006). C programming has been researched from various dimensions and explored for its effectiveness with PBL mode of delivery (Striegel & Rover, 2002). Contemporary approaches have been established (King, 1996).

A four step method, where software packages, technology and tools were introduced to teach C programming effectively (Budny et al., 2002). Research and reflections have been discussed on teaching of C programming language with the design aspects (Gao et al., 2002). Methods for adaptive, engaging, and explanatory visualizations have been created for delivery of C programming course (Brusilovsky & Spring, 2004). Gamified android apps have been developed as well (Talingdan & Llanda, 2019).

PBL has also been researched to design effective methods to teach programming. Of the many available learning strategies that are available, the popular few are: problem based, puzzle based, pair programming, pre-recorded lectures and game themed (Mohorovicic & Strcic, 2011). Comprehension capability is a major factor that decides the concept complexity (Milne & Rowe, 2002). The seven steps method with adaptations has been devised to integrate PBL and programming (Nuutila et al., 2005). Tools like RoboCode with non-prescriptive approach have been developed integrating PBL experience (O'Kelly & Gibson, 2006). Statistical tools like SPSS have been used to analyze the difficulties in learning introductory programming so as to aid the educators in developing the alternative and effective strategies (Looi & Seyal, 2014). PBL method has been combined with flipped classroom mode for programming course resulting in creating enjoyable learning experience (Chis et. al., 2018). Self-learning, active exploration and mutual cooperation are essential elements in the learning process (Peng, 2010). When we talk about open-ended problems or real-world problems, it has its own interpretation in the programming and has to be adapted accordingly (Nuutila et al., 2008).

PBL in programming also has challenges to overcome. Studies indicate issues with respect to group dynamics, uncertainty in the tolerance and study skills that pedagogy demands were not easy to adhere to by everyone in the class (Kinnunen & Malmi, 2005). When used for programming, like in traditional approach, there is a need for specific focus on workshop conduction, the nature of problems used during the class sessions, the kind of group structures, the designed learning outcomes, assessment process designed according to the context and process of evaluation to be used (O'Kelly et. al., 2004). The process needs preparation and continuous engagement from the faculty to iteratively improve the process (Azer, 2011). A survey carried out from 300 programming teachers classifies the challenges of teaching programming with PBL into two categories: intrinsic and extrinsic. Intrinsic includes the skills, knowledge, the attitude of students along with the methodology used for teaching and learning. Extrinsic includes the time management for the process, lack of support with respect to infrastructure, the practice needed for programming and the process used for student assessment (Sentence & Csizmadia, 2017).

A structured PBL implementation plan is crucial to the process especially when it is being designed at the course level. As research and studies indicate that PBL is more effective when it spans for longer time and across courses than for an individual course, the faculty needs to structure and organize the delivery meticulously (Chen et. al., 2021). The learning objectives and activities must be designed considering the time frame and the skills that need to be developed by the students. This paper proposes one such approach for the programming course delivery using PBL and active learning strategies.

III. Model and Methodology

This section presents the study context, practice and the model followed along with the research methodology used.

A. Context

Considering need of job-market demand for the programming skills required by the engineering students; university follows the curriculum delivery model of teaching C programming and Data Structures to all the first year students in first and second semester respectively. As the students collectively reported that they were not able to understand the Data Structures, sessions were planned to revise the required the C basics. Different set of faculty were allotted than the course teachers handling the regular semester. There were a total of seven sessions of 3 hours each which were furthers divided into 2 types - five sessions with conceptual delivery and practice and two sessions dedicated for practice sessions. Practice problems were also scheduled on HackerRank platform.

B. Class Structure

The class was divided into two batches of 35 each using simple random sampling strategy (Singh, 2003). The batch under study had 17 students from Civil engineering, 11 from Mechanical, and the remaining from Bio-Technology and Automation & Robotics engineering. Of 35, there was only one student who had a score of 90+ which earned him letter grade 'S' in the course and two students with 'A' grade. The batch had 16 'C' grades being the majority and the remaining spread across 'B', 'D', and 'E'. Coming from the non-circuit background the students had a lower intrinsic motivation towards the course which was evident from the survey carried out at the beginning of the session one. A consent form was shared and students were asked to rate on their interest for programming on a scale of 1 to 10. 1 indicated their highest interest and 10 the least. The result can be seen in Figure 1 below.

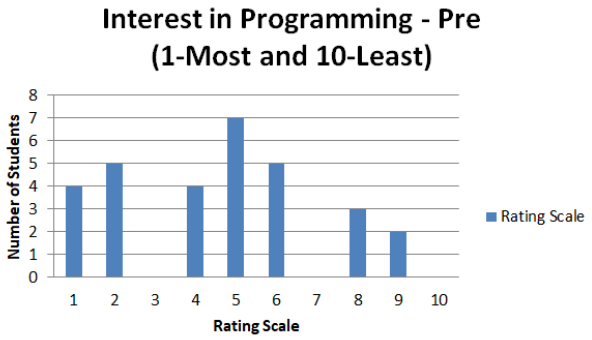

Fig. 1: Survey on interest in programming - pre

The class was approximately fifty percent split between interested and not interested as seen the Figure 1 above.

C. Sessions Structure

The session's structure can be seen in Table I below.

TABLE I
Session Structure

Session. No.	Topic	Type
1	Functions, Decision Control	PBL, Active Learning
2	Loops, Array	PBL Session
3	Session 1 and 2 programming	Practice Session
4	Strings and Array	PBL, Active Learning
5	Pointers	PBL Session
6	Session 3 and 4 programming	Practice - HackerRank
7	Structures and revision	PBL, HackerRank

The sessions were split across learning and practice. Online platform HackerRank was used for the programming practice as students were already familiar with the usage.

D. Mini-Ethnography

Ethnography is the qualitative study to capture the social meanings and activities in the logically occurring natural setting. The researcher participates directly in the process and involves in data collection without imposing any meaning to it externally (Brewer, 2000). Ethnography is based on the premise that culture is learned and shared amongst the members of the group, which usually is and most importantly, it can be described and understood (Morse et. al., 1995). The method has been exhaustively used in education settings (Pole & Morrison, 2003). The study with respect to the problem context has been discussed on the lines of objective and span of study – to be either holistic or micro (Hammersley, 2006). Ethnography has cultural knowledge, thick description and cultural insider (Croucher & Cronn-Mills, 2021).

Though ethnography usually spans for years as originated from the anthropology perspective, considering the nature, objective, and type of the study and as it is set in a social setting, one can also conduct mini-ethnography (Weinstein & Ventres, 2000). Mini-ethnography came in because of time and financial limit which can also span from weeks to months (Storesund & McMurray, 2009). Choosing the right design matters in the study so that it enables the researcher to answer the research question, assist in reaching the data saturation and complete the study in stipulated time (Fusch et. al., 2017). The study is guided by a general research question to focus on the objective of the study.

E. Research Question

To fit in the context of mini-ethnography as the question is narrow and specific (White, 2009), we formulate our research question as understanding the student behavior in the PBL oriented programming classes. We frame the following sub-questions with respect to the study:

- What is effective PBL delivery model to teach fundamental programming concepts
- How does the context differ for the non-circuit branch students
- What is the student behavior and motivation in and off the class sessions

F. Model Design

The model employed to teach the concepts can be seen in

Figure 2 below.

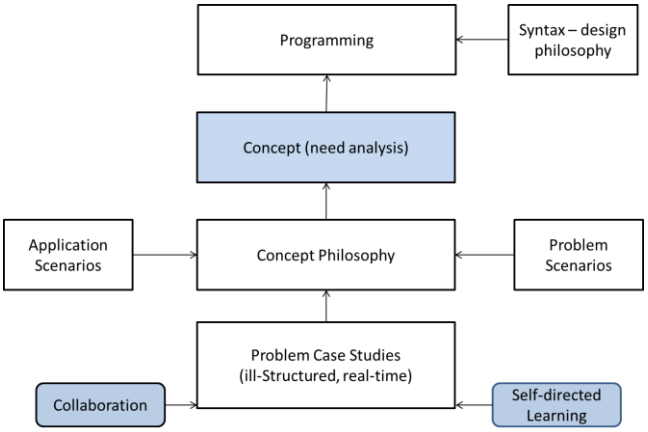

Fig. 2: The teaching-learning model

Students were first introduced with problem case studies that were ill-structured building knowledge towards concept philosophy. For example, to teach functions, case study was designed for decomposition. Once when students in teams arrived at the philosophy, they were further provided with the pertinent problem and application scenarios. After this process, the concept was explained along with the philosophy of the programming syntax and how C program provided it in the library. For everything they studied, the learning was backed with the design philosophy. Several active learning strategies like one-minute summary, pop-quiz, problem solving case studies were timely introduced in the process. No slides were used in the delivery and sessions were in chalk-talk book-pen mode.

IV. Results and Data Analysis

In this study the data is collected using the observations and the feedback forms collected using Google forms. This section presents the summary of noted experience and further elaborates on the student class performance and feedback.

A. Classroom Experiences

Three research students participated in the process of data collection by being part of the class and recording the class experience. The first experience was noted form session one. The second experience was noted from session two and third experience noted from session four.

1) Researcher one experience

There were many students who did not respond nor were attentive to what was being taught. The students were slow to catch the concepts which they already learnt in previous semester. The instructor spent sufficient time on basic topics, but still the students were lagging to cope up. Students discussed during case studies but did not arrive at required goal. Most of them did not know the basic syntax. While explaining the concepts, task was given for students to write the memory diagram and most of the class was not sure on what to do. Only a handful completed the task. Students just sat looking at the board. They lagged at the basics. They were trying to discuss and help each other but still lagged in getting concepts right. They were unaware of correct syntax. When instructor quoted a movie example they were not able to relate as they lagged in conceptual basics. The classroom was a mess and there were also discussions on topics not related to the class. Many were not interested to even to seek help to debug the errors.

2) Researcher two experience

Though students understood the philosophy of concepts, gave examples, they were not able to map the concepts to programs. Students were interactive in the class. Majority of the class were not able to code even the simple tasks like taking input, conditions, loops, etc. While some copied what was written on board as is, they made no efforts to understand why. Most students even missed the flow that input has to be taken first before coding the processing part. They had no idea on where to use the syntax elements and why to use them. When the syntax design philosophy was explained, step by step starting from main and the concept studied, students were able to understand the need and the way compiler processes the lines. Students did not know where to use the required programming constructs like break, continue etc. They seemed confident with concepts but not with code. Some students were busy correcting syntax without understanding what the error was.

3) Researcher three experience

Students participated actively in the case study discussions. Students had the basic structure of code and a few students struggled with basic syntax. Those who completed the code were helping others. Repeated solving of similar problems reduced the number of mistakes with time. Many students needed repetition of concepts. The student engagement was fairly good with lively participation. A few students were attempting to visualize the code on paper and failed to do so. While still made many errors on the first problem, the numbers were minimized with repeated similar examples. There was willingness in almost everyone to write the code.

B. Test and Feedback Data

The session started with two questions as they had already studied the concepts once. The first question was why do we need functions and the second one was why do we need decision control statements. Students had to answer in small paragraphs. The answers were thematically coded and the results can be seen in Table II.

TABLE II
WHY FUNCTIONS - PRE

Keywords	Number of Students
break	3
chunks	2
code	3
complexity	2
easier	2
efficiency	2
execution	4
functions	12
logic	2
perform	7
particular task	3

program	16
rewriting	3
smaller	3
understand	3

The thematically coded statistics for decision control statements can be seen in Table III.

TABLE III
DECISION CONTROL - PRE

Keywords	Number of Students
conditions	6
execution	11
group	5
order	3
program	5
statements	11

The same question was asked after the first PBL session where the concepts were covered with design philosophy and case studies and the thematic coding of the answers for why functions can be seen in Table IV.

TABLE IV
WHY FUNCTIONS - POST

Keyword	Number of Students
program	14
library	7
perform a specific task	7
decompose	7
function call	4
readability	4
rewriting	3

The thematic coding for the question asked on why do we need decision control can be seen in Table V.

TABLE V
DECISION CONTROL - POST

Keyword	Number of Students
conditions	15
decide the order	4
group of statements	8
execution	14

As seen from the Tables II to V we can observe that the number of words coded is less and the quality is high in the post as compared to the pre.

A survey was taken at the end of first session asking for their interest in programming and the results are recoded in Figure 3. Students had to rate on the scale of 1 to 10 as asked previously.

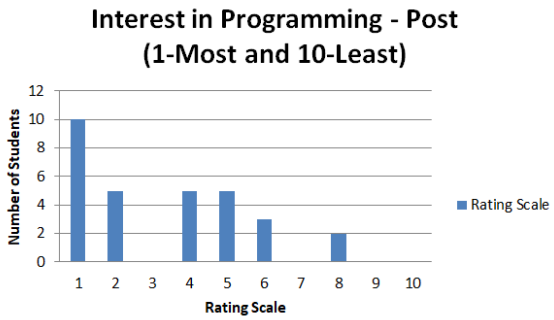

Fig. 3: Survey on interest in programming – post

The question was again repeated on the last day, last session and the results are presented in Figure 4.

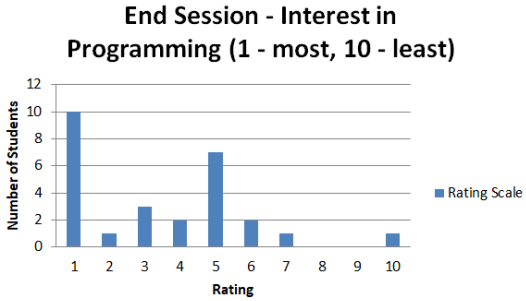

Fig. 4: Survey on interest in programming – end

A feedback was collected from the students asking if the method was employed in the previous semester, would it have an impact on the learning process and nearly 90% of the students agreed to it that it would have positive impact. The result can be in Figure 5.

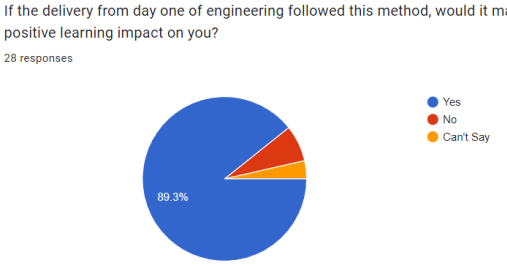

Fig. 5: Feedback on the method employed

V. DISCUSSION

If we observe the thematic coding results, we can notice the difference in answer quality between pre and post. The quality of keywords is better in post as compared to the pre. Words like decomposition, readability etc. are used are used for function answers in post while pre has words like program, functions, which are directly associated words. The groups are small in case of post. The number is small and precise indicating the quality.

If we observe the comparison of interest towards the programming from pre and post, we can notice that though it is only one session, there is a significant shift in the number. The result is presented in Figure 6.

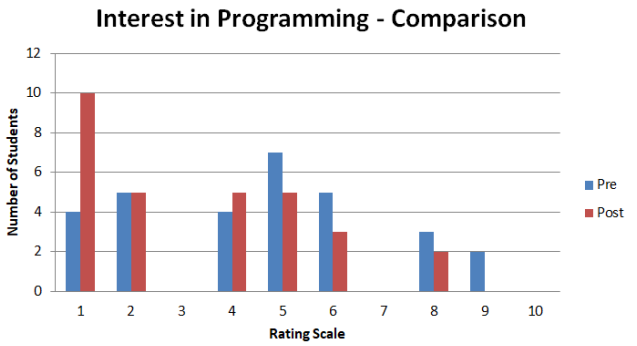

Fig. 6: Pre and post interest comparison

Based on the feedback even the model adapted as presented in the Figure 2 is found to be effective to teach the programming. As the sessions are of three hours each, there is enough scope to have case studies and longer discussions.

From the class experiences that are noted over three sessions from three research students we can observe that:

- Students were not clear with objectives of case studies in the first session which improved with following sessions
- Though students were able to understand the philosophy and the concepts, they were not able to understand the syntax
- With repeated similar problems and on understanding the philosophy of the syntax, most were able to write the correct syntax
- Common errors were resolved after repeated exercises
- Unless the need was explained and students were intrinsically motived, they showed no interest in ask completion
- For the syntax to be applied students need to know the process of why
- Engagement and discussions with respect to course concepts increased with time
- There was a spike in interest as the classes progressed and number of students who completed and helped others increased with time
- Case studies from real time help to connect students immaterial of engineering domain they come from
- Motivation can be increased if the design philosophy is understood by the students

VI. Conclusion

The problem based learning approach followed to revise the c programming to non-circuit branch students was effective as compared to the traditional methods. While students appreciated the method employed, also showed motivation and interest towards programming knowing that it is required for all the domains and not just computer science. The case studies designed related to the concept philosophy rather than the underlying concept directly playing amidst the known and unknowns connecting to the problem and application scenarios. The approach can be employed even to the computer science students. The further goal of this work is to design templates which can assist in creating the first level ill-structured problem case studies.

References

Andersen, L. O. (1994). *Program analysis and specialization for the C programming language* (Doctoral dissertation, University of Cophenhagen).

Azer, S. A. (2011). Introducing a problem-based learning program: 12 tips for success. *Medical teacher*, *33*(10), 808-813.

Barrows, H. S., & Kelson, A. (1996). Problem-based learning and problem solving. *Probe: Newsletter of the Australian Problem-based Learning Network*, *26*, 8-9.

Brewer, J. (2000). *Ethnography*. McGraw-Hill Education (UK).

Brusilovsky, P., & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. In *EdMedia+ Innovate Learning* (pp. 1264-1271). Association for the Advancement of Computing in Education (AACE).

Budny, D., Lund, L., Vipperman, J., & Patzer, J. L. I. I. I. (2002, November). Four steps to teaching C programming. In *32nd Annual Frontiers in Education* (Vol. 2, pp. F1G-F1G). IEEE.

Butler, R. (1987). Task-involving and ego-involving properties of evaluation: Effects of different feedback conditions on motivational perceptions, interest, and performance. *Journal of educational psychology*, *79*(4), 474.

Butler, R., & Nisan, M. (1986). Effects of no feedback, task-related comments, and grades on intrinsic motivation and performance. *Journal of educational psychology*, *78*(3), 210.

Chen, J., Kolmos, A., & Du, X. (2021). Forms of implementation and challenges of PBL in engineering education: a review of literature. *European Journal of Engineering Education*, *46*(1), 90-115.

Chis, A. E., Moldovan, A. N., Murphy, L., Pathak, P., & Muntean, C. H. (2018). Investigating flipped classroom and problem-based learning in a programming module for computing conversion course. *Journal of Educational Technology & Society*, *21*(4), 232-247.

Croucher, S. M., & Cronn-Mills, D. (2021). Ethnography. In *Understanding Communication Research Methods* (pp. 93-111). Routledge.

Fusch, P. I., Fusch, G. E., & Ness, L. R. (2017). How to conduct a mini-ethnographic case study: A guide for novice researchers. *The Qualitative Report*, *22*(3), 923-942.

Gao, H., Qiu, Z., Wu, D., & Gao, L. (2015). Research and reflection on teaching of C programming language design. In *International Conference of Young Computer Scientists, Engineers and Educators* (pp. 370-377). Springer, Berlin, Heidelberg.

Gurevich, Y., & Huggins, J. K. (1992). The semantics of the C programming language. In *International Workshop on Computer Science Logic* (pp. 274-308). Springer, Berlin, Heidelberg.

Hammersley, M. (2006). Ethnography: problems and prospects. *Ethnography and education*, *1*(1), 3-14.

Hannafin, M. J., & Hooper, S. R. (1993). Learning principles. *Instructional message design: Principles from the behavioral and cognitive sciences*, *2*, 191-227.

Hendriana, H., Johanto, T., & Sumarmo, U. (2018). The Role of Problem-Based Learning to Improve Students' Mathematical Problem-Solving Ability and Self Confidence. *Journal on Mathematics Education*, *9*(2), 291-300.

Hillman, W. (2003). Learning how to learn: problem based learning. *Australian Journal of Teacher Education*, *28*(2), 1-10.

Hmelo-Silver, C. E., & Barrows, H. S. (2006). Goals and strategies of a problem-based learning facilitator. *Interdisciplinary journal of problem-based learning*, *1*(1), 4.

Hmelo-Silver, C. E., & Barrows, H. S. (2008). Facilitating collaborative knowledge building. *Cognition and instruction*, *26*(1), 48-94.

King, K. N. (1996). *C programming: a modern approach*. WW Norton & Co., Inc.

Kinnunen, P., & Malmi, L. (2005). Problems in problem-based learning-experiences, analysis and lessons learned on an introductory programming course. *Informatics in Education*, *4*(2), 193-214.

Kleinert, H., Green, P., Hurte, M., Clayton, J., & Oetinger, C. (2002). Creating and using meaningful alternate assessments. *Teaching Exceptional Children*, *34*(4), 40-47.

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.

Krampen, G. (1987). Differential effects of teacher comments. *Journal of Educational Psychology*, *79*(2), 137.

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. *CSCL: Theory and practice of an emerging paradigm*, 83-124.

Looi, H. C., & Seyal, A. H. (2014). Problem-based learning: An analysis of its application to the teaching of programming. *International Proceedings of Economics Development and Research*, *70*, 68.

Malmia, W., Makatita, S. H., Lisaholit, S., Azwan, A., Magfirah, I., Tinggapi, H., & Umanailo, M. C. B. (2019). Problem-based learning as an effort to improve student learning outcomes. *Int. J. Sci. Technol. Res*, *8*(9), 1140-1143.

Mezirow, J., & Taylor, E. W. (Eds.). (2009). *Transformative learning in practice: Insights from community, workplace, and higher education*. John Wiley & Sons.

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, *7*(1), 55-66.

Mitchell, J. C. (1996). *Foundations for programming languages* (Vol. 1, pp. 2-1). Cambridge: MIT press.

Morse, J. M., & Field, P. A. (1995). *Nursing research: The application of qualitative approaches*. Nelson Thornes.

Mohorovicic, S., & Strcic, V. (2011). An overview of computer programming teaching methods. In *Central European Conference on Information and Intelligent Systems* (p. 47). Faculty of Organization and Informatics Varazdin.

Nuutila, E., Törmä, S., & Malmi, L. (2005). PBL and computer programming—the seven steps method with adaptations. *Computer Science Education*, *15*(2), 123-142.

Nuutila, E., Törmä, S., Kinnunen, P., & Malmi, L. (2008). Learning programming with the PBL method—experiences on PBL cases and tutoring. In *Reflections on the Teaching of Programming* (pp. 47-67). Springer, Berlin, Heidelberg.

O'Kelly, J., & Gibson, J. P. (2006). RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 217-221).

O'Kelly, J., Mooney, A., Bergin, S., Gaughran, P., & Ghent, J. (2004). An overview of the integration of problem based learning into an existing computer science programming module. *Pleasure by Learning*.

Peng, W. (2010). Practice and experience in the application of problem-based learning in computer programming course. In *2010 International Conference on Educational and Information Technology* (Vol. 1, pp. V1-170). IEEE.

Pole, C., & Morrison, M. (2003). *Ethnography for education*. McGraw-Hill Education (UK).

Reiser, B. J., & Tabak, I. (2014). Scaffolding. In *The Cambridge Handbook of the Learning Sciences, Second Edition* (pp. 44-62). Cambridge University Press.

Talingdan, J. A., & Llanda, C. R. (2019). Assessment of the effectiveness of learning theories using gamified Android app in teaching C programming. In *IOP Conference Series: Materials Science and Engineering* (Vol. 482, No. 1, p. 012030). IOP Publishing.

Thorndike, E. L., & Gates, A. I. (1929). Elementary principles of education.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469-495.

Singh, S. (2003). Simple random sampling. In *advanced sampling theory with applications* (pp. 71-136). Springer, Dordrecht.

Sosnovsky, S., & Gavrilova, T. (2006). Development of educational ontology for C-programming.

Storesund, A., & McMurray, A. (2009). Quality of practice in an intensive care unit (ICU): A mini-ethnographic case study. *Intensive and Critical Care Nursing*, *25*(3), 120-127.

Striegel, A., & Rover, D. T. (2002). Problem-based learning in an introductory computer engineering course. In *32nd Annual Frontiers in Education* (Vol. 2, pp. F1G-F1G). IEEE.

Weinstein, J., & Ventres, W. (2000). Mini-ethnography: Meaningful exploration made easy. *FAMILY MEDICINE-KANSAS CITY-*, *32*(9), 600-610.

White, K. L. (2009). Meztizaje and remembering in Afro-Mexican communities of the Costa Chica: Implications for archival education in Mexico. *Archival Science*, *9*(1), 43-55.

Umar, U., Kaharuddin, A., Fauzi, A., Widodo, A., Radiusman, R., & Erfan, M. (2020, August). A Comparative study on critical thinking of mathematical problem solving using problem based learning and direct intruction. In *1st Annual Conference on Education and Social Sciences (ACCESS 2019)* (pp. 314-316). Atlantis Press.