Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.032494
Article

Tech Science Press

Sustainable Learning of Computer Programming Languages Using Mind Mapping

Shahla Gul[1], Muhammad Asif[1], Zubair Nawaz[2], Muhammad Haris Aziz[3], Shahzada Khurram[4], Muhammad Qaiser Saleem[5], Elturabi Osman Ahmed Habib[5], Muhammad Shafiq[6,*] and Osama E. Sheta[7]

[1]Department of Computer Science, National Textile University Faisalabad, 37610, Pakistan
[2]Department of Data Science, University of the Punjab, Lahore, 54590, Pakistan
[3]Department of Mechanical Engineering, University of Sargodha, Sargodha, 40100, Pakistan
[4]Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan
[5]College of Computer Science and Information Technology, Al Baha University, Al Baha, 1988, Saudi Arabia
[6]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea
[7]College of Science, Zagazig University, Zagazig, 44511, Egypt
*Corresponding Author: Muhammad Shafiq. Email: shafiq@ynu.ac.kr

Received: 19 May 2022; Accepted: 28 June 2022

Abstract: In the current era of information technology, students need to learn modern programming languages efficiently. The art of teaching/learning programming requires many logical and conceptual skills. So it's a challenging task for the instructors/learners to teach/learn these programming languages effectively and efficiently. Mind mapping is a useful visual tool for establishing ideas and connecting them to solve problems. This research proposed an effective way to teach programming languages through visual tools. This experimental study uses a mind mapping tool to teach two programming environments: Text-based Programming and Blocks-based Programming. We performed the experiments with one hundred and sixty undergraduate students of two public sector universities in the Asia Pacific region. Four different instructional approaches, including block-based language (BBL), text-based languages (TBL), mind map with text-based language (MMTBL) and mind mapping with block-based (MMBBL) are used for this purpose. The results show that instructional approaches using a mind mapping tool to help students solve given tasks in their critical thinking are more effective than other instructional techniques.

Keywords: Text programming; blocks programming; novice programmer

1 Introduction

Learning programming is a big challenge for novices. Some of them lose their hearts and discontinue their studies. Others declare it a tedious and challenging subject that requires many skills to master [1,2]. In literature, the factors that affect students' programming learning include problem-solving ability [3], courage, self-confidence, self-esteem [4], self-efficacy, mental models [5–7], and traditional teaching methods used for teaching introductory programming courses [8–10]. Nowadays, two primary programming environments have been used to teach introductory programming courses i.e., Text-based Programming Language; and Blocks-based Programming Language. Both programming environments have their own merits and demerits. A brief detail of these programming languages is already discussed in [11]. This study will try to find the impact of different programming environments on students' learning along with the mind mapping approach. Mind mapping is an effective way to organize, collect, and refine thoughts or ideas. Tony Buzan first introduced it in the 1960s. Mind mapping has been widely used as a thinking, brainstorming, and problem-solving tool in various fields like education, event planning, and organizing information.

Since Mind mapping is a useful visual tool for establishing ideas and connecting them to solve problems, this study uses a mind mapping-based problem-solving technique for solving programming problems. Generally, the programming process involves two main stages, i.e., problem-solving and the implementation phase. The first phase contains an analysis of the problem, planning the solution and finally the solution design. Mind mapping technique can be used for the first phase of programming. Once the solution to the problem is designed, then the translation of this design is made through some programming language.

This study presents mind maps as a problem-solving tool to teach programming language to undergraduate students. To the latest literature survey, it is evident that the use of mind mapping is a new way to use with block and text-based programming environments. The main contributions of the study are the following:

- We investigate which programming environment is more suitable and sufficient for novice programmers concerning problem-solving?
- We examine what will be the effect of using a mind mapping approach with a text-based programming environment on students' programming performance?
- We also figure out that what will be the effect of using a mind mapping approach with a blocks-based programming environment on students' programming performance?

The rest of this paper is organized as follows. First, Section 2 discusses the current state-of-the-art work in the domain, followed by the proposed research methodology in Section 3. Next, the experimental procedure is described in Section 4. Results have been discussed in Section 5, along with discussions. Finally, Section 6 concludes the paper.

2 Literature Review

The use of mind mapping in the field of education is worth mentioning. The mind mapping approach has been used effectively in teaching and learning [12–14]. A related literature review is presented. The study [15] used mind maps for conducting brainstorming sessions to resolve the curriculum-related problem. It was found that mind mapping served as an extraordinary thinking and problem-solving tool. An experimental study [16] was conducted in Northern Taiwan to teach a management course to university students. They used a collaborative mind map-based technique to develop problem-solving skills among students. The participants of the study effectively used mind maps to solve the problems.

The mind mapping method is also used for teaching different Computer Science courses; the study [12] tested the usefulness of mind maps in teaching the subject Data Structure and Algorithms. The analysis of the study shows that teaching data structure with mind maps is a more effective practice than teaching it without mind maps, as shown in Fig. 1.

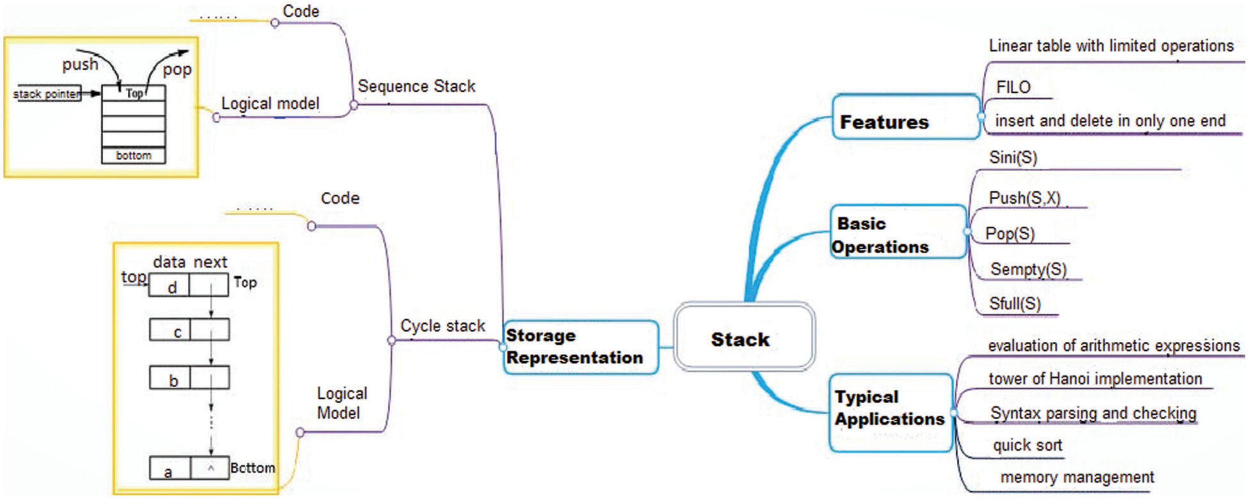

Figure 1: Mind map of the topic stack [12]

The research work [17] used mind mapping to teach computer programming language as an innovative approach. The study concluded that the mind mapping method help students in improving their programming abilities. However, the author didn't name the programming language they used for their experiment. Another study [18] used mind mapping to teach the VB programming language course to undergraduate students and found that mind mapping helped in teaching computer programming and strengthened the students' logical reasoning, as shown in Fig. 2.

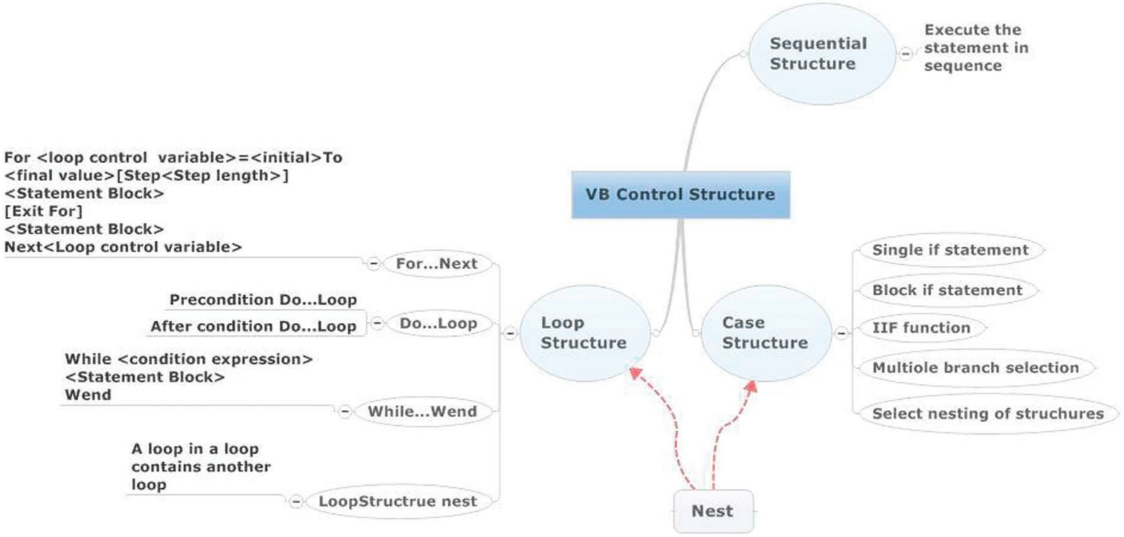
VB Control Structure
Sequential Structure
Execute the statement in sequence
Loop Structure
For...Next
For <loop control variable>=<initial>To <final value>[Step<Step length>] <Statement Block> [Exit For] <Statement Block> Next<Loop control variable>
Do...Loop
Precondition Do...Loop
After condition Do...Loop
While...Wend
While <condition expression> <Statement Block> Wend
LoopStructrue nest
A loop in a loop contains another loop
Case Structure
Single if statement
Block if statement
IIF function
Multiole branch selection
Select nesting of struchures
Nest

Figure 2: VB control structure using mind mapping [18]

3 Methodology

3.1 Research Design

In this research, we conducted a quasi-experimental study with a posttest-only design. The four different approaches i.e., BBL, TBL, MMBBL and MMTBL are used as independent variables. At the same time, programming performance is a dependent variable. This experimental work aims to observe the effects of these four different instructional approaches (TBL, BBL, MMTBL, and MMBBL) on students' programming abilities. MMTBL and MMBBL use 'Mind Mapping' as a thinking and problem-solving tool with their respective programming languages. While the TBL group works with the C programming language (Text-based programming language) and the BBL group works with MIT App Inventor as a visual block-based programming environment, as shown in Tab. 1.

Table 1: Study of different mind mapping techniques

Experimental groups	Teaching method	Post test
1.TBL	Text-based programming language without mind mapping	Programming performance
2. BBL	Blocks-based programming language without mind mapping	Programming performance
3. MMTBL	Text-based programming with mind mapping	Programming performance
4. MMBBL	Blocks-based programming with mind mapping	Programming performance

3.2 Hypotheses

The hypotheses are as follows. H0: No instructional approach will affect the students' programming performance. H1: Mean score of at least one approach significantly affects.

3.3 Participants

Total 160 undergraduate (both male and female) students from two universities of same semester took part in the study. All the students were randomly chosen and equally divided into four groups (TBL, BBL, MMTBL and MMBBL).

4 Experimental Procedures

Instructional sessions of all four groups were conducted according to the need of the experiment. A week-long mind-mapping workshop was held only for the students of groups MMTBL and MMBBL. During the workshop, lectures, self-paced tutorials, and handouts were delivered to students. The same teacher delivered lectures to the students of both groups. To check the effect of the mind mapping approach, experiments were performed using Text-based programming language (C language) and blocks-based programming language (MIT App Inventor) with and without mind mapping.

For the post-test same computing problem was assigned to all the groups. Groups TBL and BBL solved the problem without using mind mapping. In contrast, groups MMTBL and MMBBL used mind mapping with respective programming languages to find the solution of the given problem.

4.1 Instrumentation

The instrument [19] presented in Tab. 2 was adopted to measure the solution correctness of the given programming task.

Table 2: Instrument [19] to assess the solution correctness

Result	Description	Measuring scale
Complete/Correct solution.	Required complete solution, design/program code, and outcome meet with given conditions/desired output	2
Incomplete/partial solution	Incomplete outcome, design/program code/insufficient results	1
Wrong/Incorrect/ No solution	Incorrect solution, design/program code, or outcome does not meet the problem conditions/undesired output	0

4.2 Attitudinal Survey

To know the attitudes of students towards Blocks-based programming environment and Mind mapping technique, a short survey was done. The survey consists of 7-items related to blocks-based programming and mind mapping. In addition, knowing the Cronbach's Alpha reliability coefficient, a reliability test is conducted. The reliability value of the survey is 0.863.

5 Results and Discussion

We have used one-way ANOVA to test the null hypothesis. Tab. 3 shows the solution for programming problems using analysis of variance. It shows statistically highly significant ($P < 0.01$) results, i.e., at least two groups (TBL, BBL, TBL and MMBBL) regarding problem's solution had a significant difference from each other, hence rejecting the null hypothesis.

Table 3: Analysis of variance

Variation	Freedom degree	Sum of squares	Mean squares	F score	Prob.
Size of group	3	26.07	8.7	16.4**	0.000
Error	157	82.03	0.6		
Total	160	108.1			

Note: ** = Highly significant ($P < 0.01$).

We have compared these mean values by using the LSD (Least Significant Different) test at a 5% significance level as shown in Tab. 4. It is evident from Tab. 4 that TBL has the lowest score of 0.83 with a standard error of 0.143 as compared to MMTBL, BBL and MMBBL. A mean value of 1.08 was obtained using BBL with an error of 0.136, which is higher compared to TBL. However, these were significantly less different by using BBL and TBL.

Table 4: Comparison of means by LSD test at 5% level of significance

Group	Mean ± SE
1. TBL	0.83 ± 0.143 B
2. BBL	1.08 ± 0.136 B
3. MMTBL	1.75 ± 0.086 A
4. MMBBL	1.73 ± 0.080 A

Means sharing similar letters are statistically non-significant ($P > 0.05$). Moreover, the mean value of BBL was statistically significant ($P < 0.05$) and lower weight as compared to groups MMBBL and MMTBL. The groups MMTBL and MMBBL were non-significantly different from each other but had significantly higher mean values than the groups TBL and BBL. The mean score values for Groups MMTBL and MMBBL were 1.75 and 1.73, with standard error values of 0.086 and 0.080, respectively.

To check whether the results of the programming problem's solution of different groups are produced by chance or whether these values vary because of the different instructional methods, the Chi-square test was performed as depicted in Tab. 5. Since the Chi-square value is 41.82 and Prob. is 0.000, it is said with 99.9% confidence that these results are produced due to the different instructional methods used for four other groups. The results of this test show the relationship between the programming abilities of students and the different instructional strategies used to teach programming to them.

Table 5: Chi-square distribution

Solution	Groups				Total
	1. TBL	2. BBL	3. MMTBL	4. MMBBL	
No solution	20	13	2	1	36
	50.00%	32.50%	5.00%	2.50%	22.50%
Incomplete solution is produced	7	11	6	9	33
	17.50%	27.50%	15.00%	22.50%	20.60%
Appropriate solution is produced	13	16	32	30	91
	32.50%	40.00%	80.00%	75.00%	56.90%
Total	40	40	40	40	160
	100.00%	100.00%	100.00%	100.00%	100.00%

Note: Chi-square value = 41.82; Prob. = 0.000.

No Solution: Fig. 3 shows the result analysis of all four groups. It can be seen that 50% of students in the TBL group failed to provide the required solution for a given computing problem. The results of the BBL group are comparatively better than the TBL, but still, 32.5% of students provided no solution to the programming problem. The results of both the MMTBL and MMBBL show that very few students provided no answer. Incomplete Solution: Fig. 3 shows that both TBL and MMTBL have almost an equal number of students who produced an incomplete solution. Similarly, BBL and MMBBL have practically the same results for an insufficient solution to the problem.

Appropriate Solution: It can be seen clearly from Fig. 3 that most of the students of group MMTBL were successful in producing the accurate solution. The MMTBL and MMBBL group results were comparatively better than the results of the other two groups (TBL and BBL). Fig. 4 also shows the group-wise percentage regarding solution correctness. To check students' attitudes towards Mind mapping, a survey was conducted. The participants were from two groups (MMMBBL and MMTBL). A 5-point Likert scale was used.

Fig. 5 shows that students of groups MMBBL and MMTBL strongly agreed that mind maps are more straightforward and more accessible in use than flowcharts. Since mind maps are less restrictive than flow charts and provide more freedom to put your thoughts on paper, most of the students give a positive response against the statement. Fig. 6 shows that both MMBBL and MMTBL groups agreed with the statement that the mind mapping approach is helpful in problem-solving. In addition, students' positive feedback shows the effectiveness of this approach. Generally, students avoid using flowcharts or other problem analyzing techniques for programming and jump directly to the code section. However, mind maps will force them to think before jumping to the code. Students' responses against the statement that mind maps enforce programmers to think before code show that most of them agree with the statement, as shown in Fig. 7.

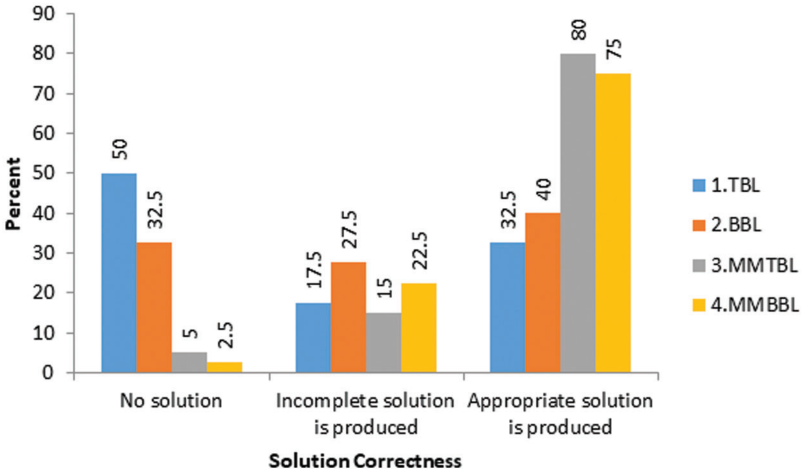

Figure 3: The result analysis of solution correctness

This study focuses on using mind mapping as a problem-solving tool for teaching two different programming languages to students. The experiments were performed with and without using mind mapping. First, the study examines the sole impact of text-based programming language (C language) and Blocks-based Programming language (MIT App Inventor) on students' programming performance. Students who studied C language were placed in group TBL and students who studied MIT App Inventor were placed in BBL group. The comparison of the results of both groups shows that group BBL had a mean value of 1.08 with a standard error of 0.136, which is statistically different and higher mean value from group TBL.

This shows that programming languages affect the programming abilities of the students. Blocks-based programming languages, designed for novice programmers, provide a simple and easy start compared to syntax-based or text-based programming languages, but these languages have their limitations like dealing with complex problems. Secondly, this study investigates the effect of mind mapping with two programming languages on students' programming performance. When the programming performance of the MMTBL group was compared with the TBL group, a significant difference was found. Though students of MMTBL took more time to solve the given problem but they performed much better than the students of the TBL group, which shows that mind mapping helps novice programmers in their thinking process to solve the given problem. Likewise, when the programming performance of the MMBBL group was compared with the students of the BBL group, there was a significant difference which confirms the effectiveness of the mind mapping approach over the traditional method. While analyzing the results of students of the MMTBL and MMBBL groups, no significant difference was found. The finding of this experimental work shows that students who use mind maps for programming perform better than students who do not use mind maps for programming. And also confirms the results of other studies [12–17] regarding the effective use of mind mapping in teaching computer programming to students. Therefore, novice programmers should use some problem-solving or thinking tools such as mind mapping or flow charts to help them in computer programming.

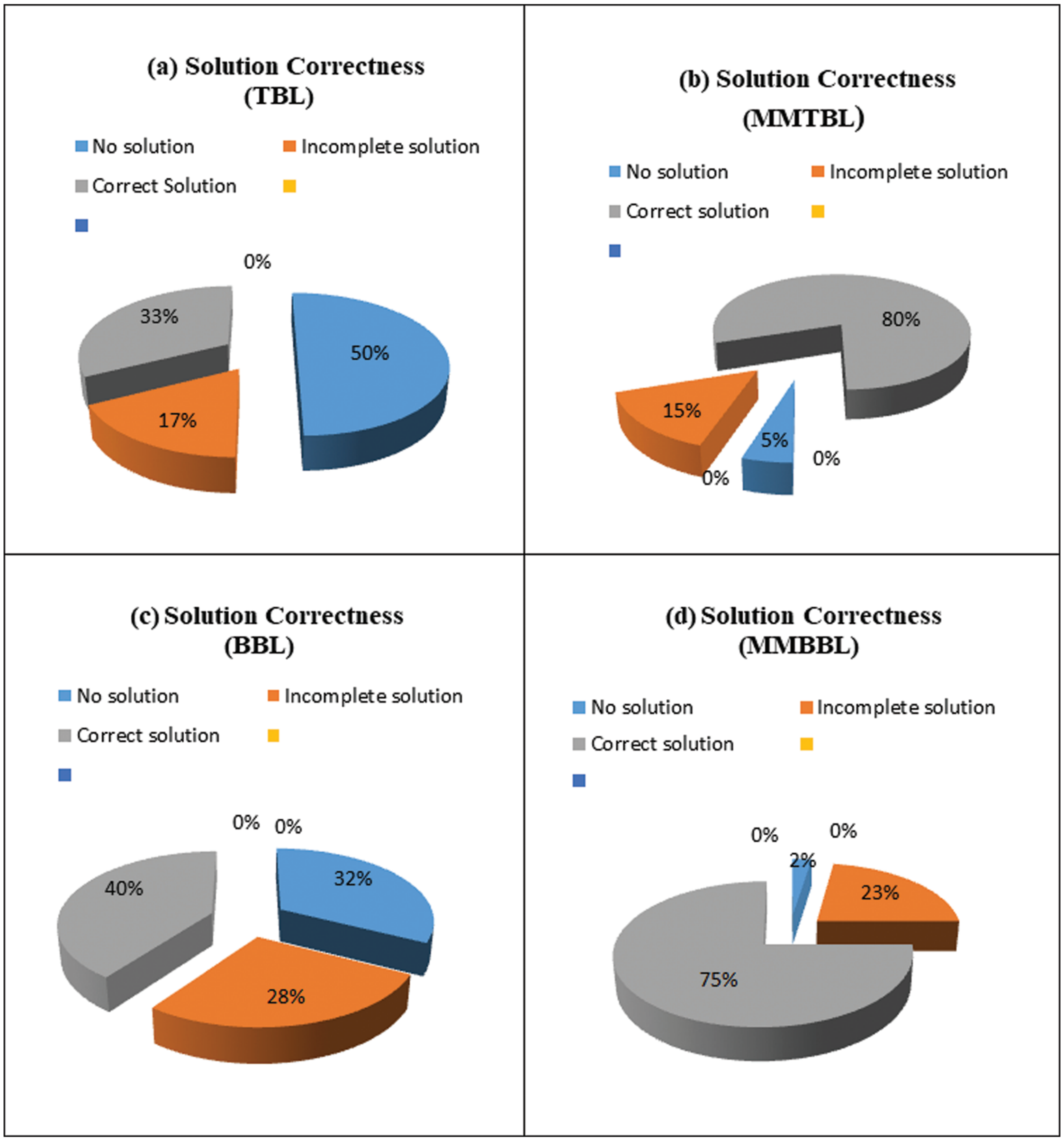

Figure 4: Group-wise percentage regarding solution correctness

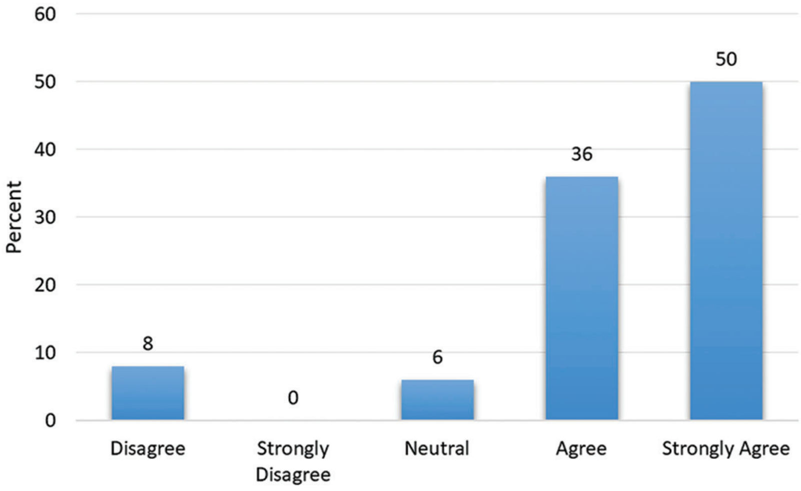

Figure 5: Students satisfaction level regarding mind mapping

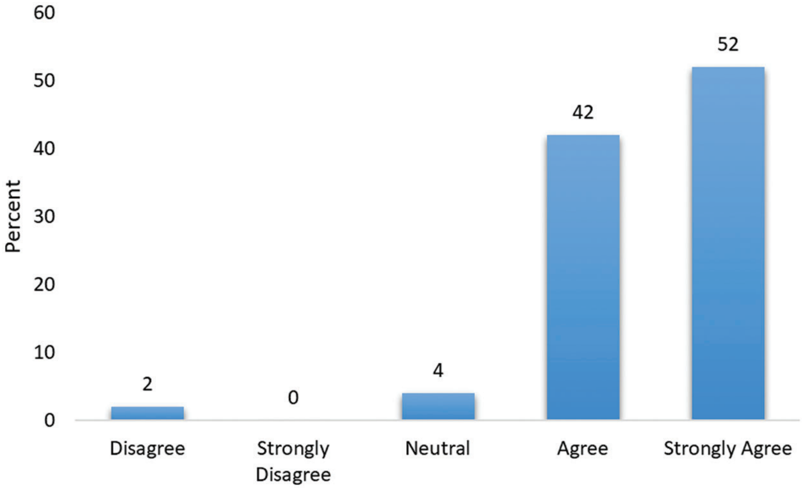

Figure 6: Use of mind maps in problem-solving

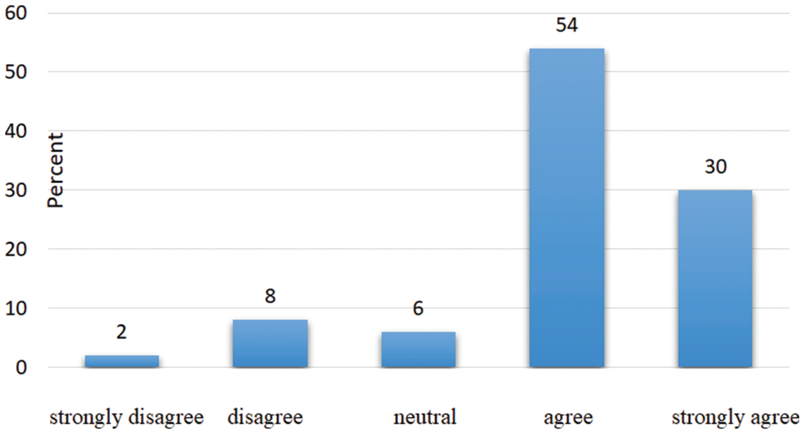

Figure 7: Mind maps enforce programmers to think before code

6 Conclusions

This experimental study presents how the mind mapping approach impacts the learner's programming ability during teaching different programming languages. For this purpose, the usefulness of mind maps is tested with two programming environments. The use of mind maps with both programming environments proved helpful in assisting students in critical thinking and solving computational problems. Therefore, students must be encouraged to use such problem-solving tools to help them better understand programming concepts. It can also be concluded that teaching programming can be more productive using this MM-based technique. The outcome of this study is beneficial for academicians and university professors to teach modern programming languages through mind maps at the school and university levels. One of the key limitations of this study is the limited time period. As we have used only two programming languages for this study, however, the study can be enhanced to include more programming languages to check the effectiveness of the MM-based approaches. Another prominent extension could also be possible through text to mind map conversion automatically.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. E. Burge and G. C. Gannod, "Girls on the go: A CS summer camp to attract and inspire female high school students," in *Proc. ACM Technical Symp. on Computer Science Education*, Colorado, USA, pp. 615–620, 2013.

[2] F. Allan and J. Margolis, "Unlocking the clubhouse: Women in computing," in *Proc. Technical Symp. on Computer Science Education*, Colorado, USA, pp. 23, 2013.

[3] A. Gomes and A. J. Mendes, "Learning to program-difficulties and solutions," in *Proc. Int. Conf. on Engineering Education*, University of Coimbra, Portugal, 2007.

[4] I. McPherson, "Reflexive learning: Stages towards wisdom with dreyfus," *Educational Philosophy & Theory*, vol. 37, no. 5, pp. 705–718, 2005.

[5] S. Wiedenbeck, D. LaBelle and V. Kain, "Factors affecting course outcomes in introductory programming," in *Proc. Workshop of the Psychology of Programming Interest Group*, Carlo, Ireland, pp. 97–110, 2004.

[6] R. E. Bruhn and P. J. Burton, "An approach to teaching java using computers," *ACM SIGCSE Bulletin*, vol. 35, no. 4, pp. 94–99, 2003.

[7] D. Radošev, T. Orehovački and A. Lovrenčić, "Verificator: Educational tool for learning programming," *Informatics in Education*, vol. 8, no. 2, pp. 261–280, 2009.

[8] A. Moldavan, M. Alesia, C. Edwards-Leis and J. Murray, "Design and pedagogical implications of a digital learning platform to promote well-being in teacher education," *Teaching & Teacher Education*, vol. 115, pp. 103732, 2022.

[9] M. Svanström, J. Sjöblom, J. Segalàs and M. Fröling, "Improving engineering education for sustainable development using concept maps and multivariate data analysis," *Journal of Cleaner Production*, vol. 198, pp. 530–540, 2018.

[10] A. Bredillet and N. Christophe, "Learning and acting in project situations through a meta-method (MAP) a case study: Contextual and situational approach for project management governance in management education," *International Journal of Project Management*, vol. 26, no. 3, pp. 238–250, 2008.

[11] S. Gul, M. Asif, W. Ahmad and U. Ahmad, "Teaching programming: A mind map based methodology to improve learning outcomes," in *Proc. Int. Conf. on Information and Communication Technologies*, Karachi, Pakistan, pp. 209–213, 2017.

[12] F. He, X. Miao, B. Wu and S. Yao, "Using mind map as learning tool in 'data structure' teaching," in *Proc. IEEE Int. Conf. on Computer & Information Technology*, Shaanxi, China, pp. 761–764, 2014.

[13] M. Li and R. Yuan, “Enhancing students’ metacognitive development in higher education: A classroom-based inquiry,” *International Journal of Educational Research*, vol. 112, pp. 101947, 2022.

[14] N. A. Alrehaili, M. A. Aslam, D. H. Alahmadi, D. A. Alrehaili, M. Asif *et al.,* “Ontology-based smart system to automate higher education activities,” *Complexity*, vol. 2021, pp. 1–20, 2021.

[15] F. Paykoç, B. Mengi, P. O. Kamay, P. Önkol, B. Özgür *et al.,* “What are the major curriculum issues?: The use of mind mapping as a brainstorming exercise,” in *Proc. Int. Conf. on Concept Mapping*, Tallinn, Estonia, vol. 2, pp. 1–4, 2004.

[16] G. J. Hwang, C. H. Wu, L. Huang and F. R. Kuo, “A mind map-oriented mobile learning approach to promoting creative thinking ability of students in a business course,” in *Proc. IEEE Int. Conf. on Wireless, Mobile & Ubiquitous Technology in Education*, Kagawa Japan, pp. 242–248, 2012.

[17] M. N. Ismail, N. A. Ngah and I. N. Umar, “The effects of mind mapping with cooperative learning on programming performance, problem solving skill and metacognitive knowledge among computer science students,” *Journal of Educational Computing Research*, vol. 42, no. 1, pp. 35–61, 2010.

[18] Y. Liu, Y. Tong and Y. Yang, “The application of mind mapping into college computer programming teaching,” *Procedia Computer Science*, vol. 129, pp. 66–70, 2018.

[19] F. P. Deek, S. R. Hiltz, H. Kimmel and N. Rotter, “Cognitive assessment students’ problem solving and program development skills,” *Journal of Engineering Education*, vol. 88, no. 3, pp. 317–326, 1999.