journal homepage: https://journal.pktm.com.my/

International Journal of Technical Vocational and Engineering Technology

e-ISSN2710-7094, Vol. 6, No. 2 (2025)

Empowering Students' Problem-Solving Skills through CDIO-Based Learning Algorithm Using Sphero as Learning Tool in Programming

Astri Idayu Athesan[1*], Mohd Faizul Emizal Mohd Ghazi[1], Dwiny Meidelfi[2]

[1]Department of Academics and Lifelong Learning, Kolej Komuniti Gopeng, KM 16, Jalan Gopeng, Kawasan Perindustrian Gopeng 2, 31600 Gopeng, Perak, Malaysia

[2]Politeknik Negeri Padang, 25163, Padang, West Sumatera, Indonesia

*Corresponding Author email: astriidayu@gmail.com

ARTICLE INFO

Article History:
Received 2 July 2025
Revised 9 September 2025
Accepted 20 October 2025
Published 30 October 2025

Keywords:
Algorithmic;
CDIO;
Educational Robotics;
Problem-Solving Skills;
Programming.

ABSTRACT

This action research investigates the effectiveness of a CDIO-based learning module using the Sphero robot to enhance algorithmic understanding and problem-solving skills among Semester 1 students enrolled in Programming Fundamental (SFC 10403) at Kolej Komuniti Gopeng. Eighteen students completed identical pre-test and post-test Likert-scale questionnaires assessing algorithmic comprehension and problem-solving ability, while structured teacher observation checklists tracked engagement and participation. Post-intervention means scores increased substantially from approximately 2.17–2.44 in pre-test to 3.78–4.33 in post-test, indicating significant gains. This study addresses a research gap in the integration of CDIO pedagogy with educational robotics at the community college level, an area underexplored in existing literature. The findings suggest that combining CDIO with robotics offers a powerful, hands-on approach for strengthening programming pedagogy and aligning TVET curriculum design with Industry 4.0 skills demands, making it a valuable model for broader adoption in STEM and TVET education.

1.0 Introduction

Problem-solving is a crucial skill in the field of information technology, especially in programming courses. However, students at community colleges often face difficulties in grasping fundamental concepts such as algorithms and programming logic. This difficulty is not unique to our institution. Empirical evidence shows that novice programmers frequently struggle with algorithmic thinking, decomposition, and abstraction, the core components of computational thinking. For instance, Malik et al. (2021) observed high failure and dropout rates in introductory programming courses, attributing it partly to inadequate emphasis on algorithmic thinking over syntax and semantics. Research by Just et al. (2025) and Kurniawan et al. (2022) both agree that inexperienced programmers usually pick up syntactical difficulties fast but struggle with logical reasoning and abstract design throughout the course. These results highlight a structural pedagogical problem whereby prioritizing syntax over the concurrent development of algorithmic thinking leads to enduring learning obstacles, especially for students enrolled in non-university settings. These skills become even more critical because they involve the ability to analyse problems, design logical solutions, and systematically implement them.

It has been observed that students face difficulty in understanding and applying algorithm concepts during learning. The focus of the study is on resolving the current issue of determining the reason why students write code without first developing algorithms. Students fail to realize the relevance and importance of algorithms in the programming development process. This is a significant issue since the Programming Life Cycle (PLC), which consists of seven phases, requires algorithms as one of its fundamental elements, and algorithm creation is one of the required steps.

This action research was conducted for the subject of SFC10403 – Programming Fundamentals at Kolej Komuniti Gopeng, one of the core subjects in this course. To solve these issues and inspire students to gain a more thorough understanding and practical learning experience, the instructor employs Design Implement Experience (CDIO standard 5), Integrated Learning Experiences (CDIO standard 7), and Active Learning (CDIO standard 8) techniques. Sphero robots were used to tackle algorithmic problems as part of a CDIO-based learning module.

In the Conceive phase, students were introduced to problems requiring Sphero to complete a task (e.g., form simple shapes, navigating a maze, etc). In the Design phase, students planned their logic through flowcharts and pseudocode. The Implement phase involved coding using the Sphero Edu platform. In the Operate phase, students tested their programs physically using the robot and made improvements. Each lesson focuses on helping students to first understand the concept, then apply it to solve problems logically using Sphero. Data were analyzed descriptively using questionnaire scores and student observations.

This study aims to investigate the effectiveness of using Sphero as a mediator in a CDIO-based approach to strengthen students' problem-solving skills in algorithm learning. The objectives of the study are as follows:

i. To evaluate the effectiveness of using the Sphero robot as a mediator in CDIO-based algorithm learning.

ii. To identify changes in students' problem-solving skills after engaging in programming activities using the CDIO approach.

2.0 Literature review

The two primary themes of this study are (i) the application of CDIO pedagogy in engineering education and programming technology, and (ii) the use of educational robotics, such as Sphero, to enhance student engagement and develop algorithmic understanding.

2.1 Educational Robotics and Understanding Algorithms

Extensive research has been conducted on effective teaching and learning strategies for novice computer programmers. A central subject in this research is the development of computational thinking skills, which are deemed to be essential for problem-solving in programming. Computational thinking, a key approach to problem solving that requires logical, algorithmic processes alongside reasoning abilities. According to Mulder (2025), computational thinking encompasses four key principles: decomposition, pattern recognition, abstraction, and algorithmic thinking.

In the context of nurturing these concepts, a study by Sukirman et al. (2022) supports the use of active learning approaches through visual block-based programming, which has contributed to the improvement of students' mastery of basic programming concepts in novice education settings. Wang et al. (2021) found that integrating a range of techniques and strategies in programming lessons significantly boosted students' motivation to learn by promoting a more dynamic, effective, and successful learning environment. One such innovative strategy is robot-supported learning, which has gained popularity for cultivating computational thinking.

The use of robotic learning tools has garnered increasing attention among educational researchers. A study by Ouyang et al. (2024) found that robotics interventions in education led to positive improvements in student performance and attitudes in learning. Educational robotics, such as Sphero, have emerged as powerful tools for making programming accessible and engaging. Serving as auxiliary technology, they enable instructors to deliver lessons in a more interactive and meaningful way. The use of Sphero engaged students, particularly new ones, and improve comprehension of fundamental programming principles (Nnass et al. ,2023). Sphero functions not only as a learning tool but also as a catalyst for idea exploration, logical design, and hands-on programming implementation.

It has been scientifically demonstrated that educational robots like Sphero improve both physical interaction and visual learning experiences. A pilot study by Mariadass et al. (2020) showed a post-test improvement of approximately 22% compared to a control group using traditional methods, and students reported higher motivation and better focus when learning with Sphero. Nnass et al. (2023) also stated, besides making syntax simpler, Sphero helps students visualize loops, conditionals, and algorithms through interactive exercises, which fosters computational thinking and problem-solving abilities. Sphero's interactive and physical features are applicable to a variety of learning styles and help to demystify programming for novices, making it a more welcoming and inspiring environment.

These findings underscore the potential of robotics to transform abstract programming concepts into tangible, interactive experiences, a key premise of this research. This study bridges these gaps by applying the complete CDIO cycle with Sphero robots, offering a novel approach to teaching algorithms in technical education.

2.2 CDIO Pedagogy in Programming Education

The CDIO framework is recognized as an effective strategy to help students develop understanding progressively through experience and tangible outcomes. This approach not only emphasizes technical understanding, but also integrates competencies such as collaboration, communication, and innovation, in accordance with international engineering education standards (Wikipedia, 2024).

Kamarudin (2022) provides a complementary perspective, illustrating how CDIO-based integrated curriculum models foster teamwork, communication, and critical thinking through real-world projects. A study by Wang (2019) states that the application of CDIO-based paradigm greatly increased students' programming skills and interest, shown by better course results and encouraging comments. This demonstrates how the CDIO framework has a significant impact on teaching methods that successfully improve students' technical competency, critical thinking, teamwork, and interest through project-based, real-world learning.

According to research by Bravo et al. (2017), engineering students' collaborative learning and project management and creativity skills were enhanced using robotics through the CDIO initiative. A study later by Abdulla et al. (2019) stated that computer science students' practical abilities and inventiveness significantly improved when CDIO oriented classrooms and Outcome Based Education (OBE) were dynamically integrated. Findings by Yuan et al. (2024) also stated that significant gains in computer science students' practical abilities and inventiveness were also demonstrated by programming courses that used OBE and CDIO-based instructional design. This demonstrates that incorporating robotics into the CDIO framework, particularly in conjunction with OBE, greatly improves students' capacity for collaborative learning, creativity, and practical abilities in both computer science and engineering settings.

Kamarudin (2022) extends this discourse by introducing three integrated curriculum models (IntraICM, InterIC, and 3IC) based on CDIO, implemented within Malaysia's Department of Polytechnic and Community College Education (DPCCE). These models emphasize interdisciplinary collaboration and industry engagement, demonstrating improved learning outcomes and reduced student workload, which align with the experiential learning goals of this study. Hashim et al. (2023) developed learning modules designed using mBot to implement problem-based learning using Bloom's taxonomy paradigm. Students were subsequently able to experience real-time outputs from the codes they wrote, bridging the gap between abstract code and tangible applications, aligning with CDIO's emphasis on implementing and operating real systems.

Overall, the key research gap identified is the lack of focused action research systematically applies the full CDIO approach using Sphero robots in the teaching and learning of algorithms at technical higher education institutions such as community colleges. Existing studies tend to focus on only one phase of the CDIO framework or are conducted are conducted at the primary or secondary education levels. While CDIO and robotics have been studied individually, their integrated application in programming education, particularly at the community college level, remains underexplored. Therefore, this study is significant in empirically evaluating how intermediary tools such as the Sphero robot can support the effectiveness of the CDIO approach in holistically and comprehensively enhancing students' problem-solving skills within programming courses.

3.0 Methodology

The purpose of this action research project was to assess how well the CDIO pedagogical intervention using Sphero enhanced students' comprehension of algorithms and problem-solving abilities. It also allows the lecturer to identify learning issues directly and improve instructional practices through continuous reflection. The study used a relatively small sample (n = 18) that was purposefully selected from a single cohort, which may have limited the statistical power available for advanced inferential tests and the generalizability of the results. Thus, according to Faber et al. (2014), small sample action research is suitable to be use for pilot-level or exploratory educational interventions as it provides rich contextual insights.

Data were collected using two main instruments, which is pre- and post-tests and an observation checklist. The questionnaires used in pre- and post-tests evaluated students' comprehension of algorithms and their capacity for problem-solving. The observation checklist is used to document student engagement and behaviour.

3.1 Pre- and Post-Test Instrument

To enhance the rigor of the study, both the pre- and post-test questionnaires underwent expert review for content validity, involving input from two academics specializing in programming pedagogy. This instrument utilizes a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree) to measure students' understanding of algorithm concepts and their problem-solving skills. Each participant responded to the same set of items before (pre-test) and after (post-test) the intervention through the CDIO module involving Sphero programming, allowing for analysis of individual score changes.

The use of this scale aligns with measurement methodologies commonly applied in educational research on algorithms, such as studies assessing attitudes toward algorithm-based education using a 5-point Likert scale and analysed through descriptive statistics. The Likert scale is recognized as a valid and widely used psychometric measurement tool in educational literature.

The instrument used in this study consists of 10 statements divided into two main subscales. The first subscale assesses Algorithm Understanding, which includes items Q1, Q3, Q4, Q5, and Q7. The second subscale evaluates Problem-Solving Skills, comprising items Q2, Q6, Q8, Q9, and Q10. Each item is rated using the Likert scale with a minimum score of 1 and a maximum of 5. Data were analysed descriptively, involving the calculation of the mean and standard deviation for each item in both subscales.

Table 3.1: Likert-Scale Questionnaire to Measuring Algorithmic Understanding & Problem-Solving Ability

Item	Behavioral Indicator
1	I understand what an algorithm is and its importance in programming.
2	I am able to break down a problem into smaller, manageable steps.
3	I can design a sequence of steps (algorithm) to solve simple problems.
4	I can identify the correct order of operations in a given algorithm.
5	I find it easy to translate a real-world problem into a logical set of instructions.
6	I am confident in debugging (identifying and fixing) errors in algorithms or code.
7	I can use flowcharts or pseudocode to represent algorithms clearly.
8	I am able to solve new programming challenges by applying what I have learned about algorithms.
9	I feel confident explaining my problem-solving approach to others.
10	I believe my problem-solving skills have improved through learning programming.

3.2 Observation Checklist Instrument

The teacher observation checklist in this study was created in accordance with the principles of the Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP), a well-known and validated framework for classroom observations that evaluates student interaction, affect, and engagement in technologically advanced learning environments (Ocumpaugh et al. (2015). BROMP prioritizes the use of precise behavioural markers, systematic, momentary time sampling, and real-time coding of observable emotional, social, and engagement states.

This instrument was used by lecturers as a structured observation form to assess student behaviour and participation during teaching sessions involving Sphero. Each student is evaluated across ten behavioural indicators using a scale ranging from Weak (1) to Consistent (4). The types of indicators include active participation, initiative in Sphero programming, application of logic, collaboration, persistence, adherence to the CDIO guidelines, and proper use of equipment as shown in Table 3.2. This study used ten behavioural indicators assessed by 4-point scale which allowed lecturers to record observations quantitatively and consistently during each session. To ensure more accurate results, the observable actions were listed as recommendations.

Table 3.2: Checklist for Observation Indicators

Behavioral Indicator	Observable Action/Description
Active participation in group discussions	Exchanges thoughts, answers inquiries, and interacts with peers
Demonstrates initiative during Sphero tasks	Leads or offers to code or manage Sphero.
Practices logical thinking / articulates solution steps	Explains algorithmic reasoning and makes logical suggestions.
Collaborates effectively in group work	Cooperates, shares resources, and helps others.
Seeking help when facing difficulties	Asking for assistance when facing problems
Shows perseverance in solving problems	Demonstrates persistence in resolving issues
Clearly explain the steps of the solution	Clearly outline the steps involved in the solution.
Follows the CDIO framework in task implementation	Follows the CDIO framework when performing tasks.
Uses Sphero and equipment safely	Safely makes use of Sphero and equipment
Displays increased confidence and independence in action	Shows greater self-assurance and self-reliance in action.

3.2 Research Procedure

This study was conducted in three main phases. In the pre-intervention phase, all students completed an initial assessment (pre-test) using a Likert-scale instrument to assess their baseline understanding and skills. Next, during the implementation phase of the CDIO sessions with Sphero, the lecturer carried out structured observations based on a behavioural checklist throughout the activities. In the post-intervention phase, all students completed a final test (post-test) using the same Likert scale, and lecturer observations continued at the end of each main session using the same checklist.

Data analysis involved the use of descriptive statistics to compare pre- and post-test scores to evaluate the effectiveness of the intervention. In addition, the observation data were analysed quantitatively through average score calculations and descriptively based on behavioural observation notes. Both instruments were designed to provide a comprehensive quantitative and qualitative overview of the impact of the CDIO intervention using Sphero on students' understanding of algorithms and their problem-solving skills.

4.0 Discussion of analysis and findings

It is anticipated that the Sphero-based CDIO module would not only increase students' motivation and self-assurance but also improve their understanding of algorithmic concepts while enhancing their problem-solving and computational thinking skills.

4.1 Pre- and Post-Test

According to the study data, following the use of the Sphero robot as an intervention, the average scores of all 18 students increased across each of the 10 items measured on the Likert scale. The two primary focuses of this intervention were students' problem-solving abilities and their comprehension of algorithm concepts.

Based on the pre-test data, the average scores for all 10 items ranged from 2.17 to 2.44, with standard deviations between 0.383 and 0.511. This indicates that students' understanding of algorithm concepts and their problem-solving skills were still at a low to moderate level before the intervention was conducted. For example, item Pre_Q1, which measures overall understanding of algorithms, recorded a mean score of 2.44 with a standard deviation of 0.511, while item Pre_Q7, related to specific skills, recorded a lower mean of 2.17 with a standard deviation of 0.383. These findings reflect those students were still in the early stages of understanding and lacked confidence in completing tasks involving algorithmic concepts.

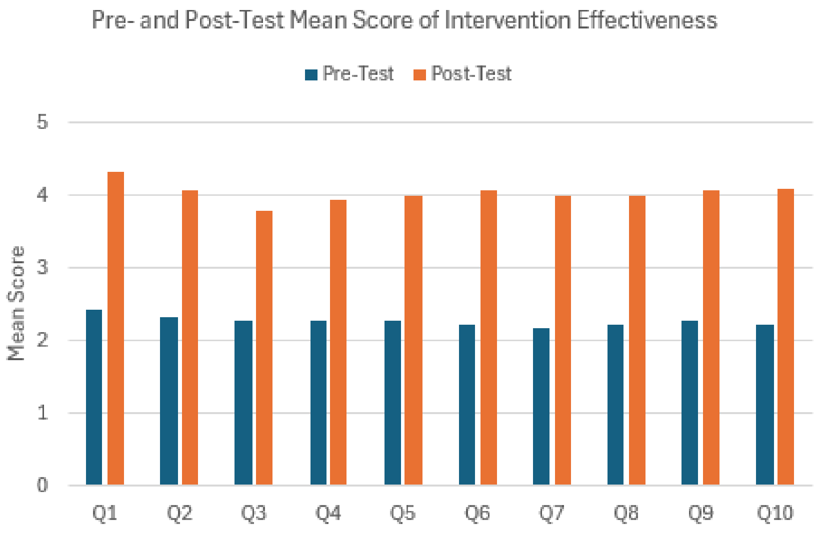

Figure 4.1: Pre – and Post-Test Mean Score of Intervention Effectiveness

Students responses to statements such as "I can solve new programming challenges" (Q8) "I am confident in debugging" (Q6)" and "I can break down a problem into smaller steps" (Q2) showed significant improvements following their experience with the Sphero learning activities. These results are consistent with earlier studies indicating that the application of robotics, like Sphero, provides chances for instantaneous introspection and practical testing (Mariadass et al.,2020). Through a visual trial-and-error process that is easier for students to understand, this method improves algorithmic logic and problem-solving abilities.

The post-test shows that pupils indicate overall positive outcomes, with all mean scores at or above 3.78 on a 5-point scale. While numerous items, including Q5, Q7, and Q8, achieved the maximum mean of 4.00, Q1 had the highest score of 4.33. This indicates that students rated themselves between "agree" and "strongly agree" on statements assessing their confidence in problem-solving and their comprehension of algorithms after completing the Sphero-CDIO module.

4.2 Observation Checklist

Through the observation checklist, the lecturer documented significant improvement in several student behavior indicators throughout the implementation of the module. Every one of the ten behavioural indicators on the post-activity teacher observation checklist recorded means scores above 3.0. These observational findings align with the findings from the pre- and post-intervention tests, which indicate that the CDIO approach, combined with the use of Sphero, effectively supports active participation, student collaboration, and engagement in project-based learning**.**

The observation revealed some noteworthy results, such as the highest scores for teamwork, resilience in problem-solving, and active participation in group discussions, which demonstrate great collaboration and tenacity. The highest scores were 3.39 for "Shows perseverance in solving problems," "Collaborates effectively in group work," and "Active participation in group discussions."

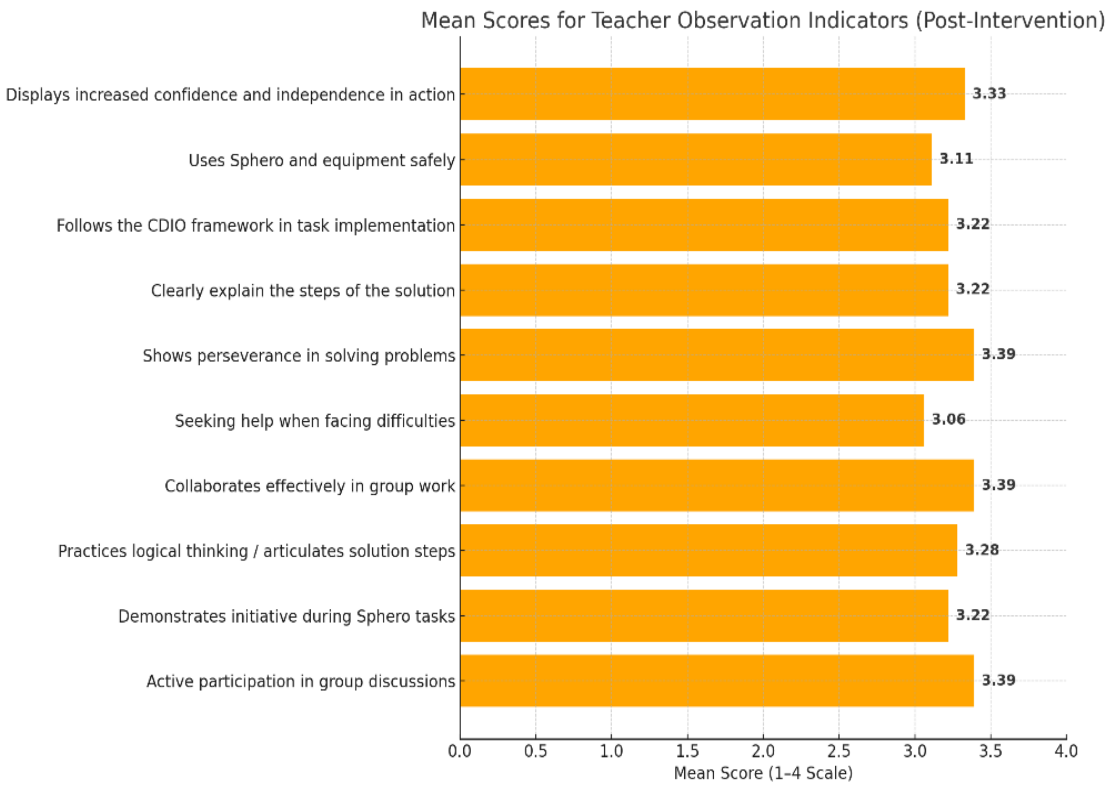

Figure 4.2 Post-Intervention for Mean Scores for Teacher Observation Indicators

As demonstrated by indicators for initiative and logical thinking that also scored higher than 3.2, students demonstrated proactive actions during tasks and actively engaged in analytical reasoning. Furthermore, the mean score for "Displays increased confidence and independence in action" was 3.33, suggesting that students' self-efficacy had improved after the intervention. Even though "Seeking help when facing difficulties" had the lowest mean score (3.06), it was still above the "sometimes" cutoff, indicating a healthy balance between handling issues independently and seeking assistance when necessary.

4.3 Interpretation and Implications

The consistently high post-intervention scores from both instrument scores demonstrates how well the CDIO-based Sphero robotics module in promoting student participation, teamwork, and critical thinking. These findings are consistent with earlier research showing that robotics-enhanced learning can enhance social and affective competencies like initiative, communication, and self-confidence in addition to cognitive abilities. It supports those of Nnass et al. (2023), who emphasized the importance of experiential learning in cultivating higher-order thinking abilities.

Similarly, Kamarudin (2022) confirmed CDIO's capability in guiding students through systematic and collaborative problem-solving. This was validated by the uniform improvements in scores observed in this study. Furthermore, Sukirman et al. (2022) stressed the importance of interactive and visual methods to teach algorithm concepts. Sphero provided a tangible, physical platform for students to test and perceive abstract concepts such as loops and conditions, reaffirming the value of interactivity in programming education.

In addition, Nnass et al. (2023) found that Sphero enhances student motivation and engagement, an effect also seen in this study, with students giving positive feedback and showed consistent improvement. This underscores the role of motivation as a key factor in learning effectiveness. This is partly because robotic kits help development of computational thinking through the tradition of tangible hands-on learning. Importantly, this study addresses a gap in existing literature concerning the integration of CDIO with Sphero in technical education.

However, it is important to acknowledge possible confounding factors that may have influenced these outcomes, including the novelty effect of using Sphero as an interactive learning tool and the lecturer's facilitation style, which could enhance engagement independently of the intervention's design. Furthermore, as the present study involved a homogenous cohort from a single community college, the results may vary across different student demographics, such as gender, prior programming experience, and digital literacy levels. Future research should therefore explore these demographic variables to better understand how the intervention's effectiveness may differ among diverse TVET student populations.

Overall, the combined use of CDIO and Sphero not only improved student understanding but also offered a more engaging and effective learning experience. The consistent progress across students indicates this approach is adaptable and scalable for broader implementation across different courses or educational settings.

5.0 Recommendations

Several recommendations can be made to further enhance this research approach. Firstly, robotics tools like Sphero should be integrated as a regular instructional strategy in foundational programming courses. Its visual and hands-on nature assist students, especially those with weaker academic backgrounds, bridge gaps in understanding abstract concepts like loops and conditions.

Future research should include a larger sample size, randomized samples and include a control group to lessen any training bias that may improve statistical robustness and external validity. The instruments used were also subjective, relying on the Likert scale and observations checklists. Therefore, it is recommended that future studies incorporate more objective assessment tools, such as code analysis or real programming tasks, to obtain a more accurate representation of students' skills.

Additionally, this study did not investigate the long-term knowledge retention. It is advised to conduct follow-up assessments to assess how well students can remember algorithm principles over an extended length of time.

Furthermore, alternative assessment tools such as algorithm design rubrics, student reflection journals, and mini presentations should be implemented. These allow students to analyse their work and understand problem-solving processes in greater depth. The ability to articulate logic and explain solutions is essential in the technical workforce and should be emphasized early. Therefore, CDIO-based training with interactive tools like Sphero should be viewed not just as a teaching aid, but as strategic, long-term approach for nurturing confident, competent, and industry-ready graduate.

From a broader TVET curriculum design perspective, the findings of this study underscore the value of embedding experiential, robotics-enhanced learning within programming and problem-solving modules to strengthen industry-relevant competencies. Integrating the CDIO approach with interactive tools like Sphero not only enriches student engagement but also ensures the development of critical thinking, collaboration, and technical proficiency in authentic contexts. Such alignment supports the production of holistic TVET graduates who can analyse problems, design solutions, and operate systems effectively meeting the skill demands of Industry 4.0 and future technological advancements.

6.0 Conclusion and Future Research

This study confirms that integrating the CDIO approach with the use of the Sphero robot in basic programming courses can effectively enhance students' problem-solving skills. The consistent improvement in scores and positive feedback reflect the approach's success in helping students understand abstract concepts like algorithms through structured, visual, and practical learning. The study enriches the literature by showcasing the effectiveness of the CDIO model in technical education contexts like Community Colleges and opens the door to broader implementation of innovative learning strategies in vocational and technical education.

Acknowledgements

The authors would like to express their sincere gratitude to students of Kolej Komuniti Gopeng who participated in this study. The authors also wish to thank the student evaluation teams for their collaboration during the observation sessions, their lively interaction and insightful comments.

Author Contributions

Athesan A. I.: Conceptualization, Methodology, Training Module, Writing-Original Draft, Trainer, Preparation; **Emizal M. F.**: Data Curation, Validation, Supervision, Editing, Writing-Review and Editing; **Meidelfi D.**: Writing-Review and Editing.

Conflicts of Interest

The manuscript has not been published elsewhere and is not under consideration by any other journal. All authors have reviewed and approved manuscript and declared no conflict of interest.

6.0 References

Abdulla, M., Motamedi, Z., & Majeed, A. (2019). Redesigning telecommunication engineering courses with CDIO geared for polytechnic education. *Proceedings of the Canadian Engineering Education Association (CEEA*. https://doi.org/10.24908/pceea.vi0.13855)

Bravo Sanchez, Flor & Cano Morales, Martha & Hurtado, Jairo & Otalora, Camilo & López, Gloria. (2017). Using Robotics to Generate Collaborative Learning, through the CDIO Initiative.

Faber, J., & Fonseca, L. M. (2014). How sample size influences research outcomes. *Dental Press Journal of Orthodontics*, *19*(4), 27–29. https://doi.org/10.1590/2176-9451.19.4.027-029.ebo

Hashim, A. S., Yusof, M. a. M., Arshad, N. I., & Muazu, A. A. (2023c). Development of Robot to Improve Learning of Programming Skills among Students. *Iraqi Journal for Computer Science and Mathematics*, 1–11. https://doi.org/10.52866/ijcsm.2023.02.03.001

Kamarudin, N. (2022). Enhancing quality TVET graduates through three integrated curriculum models : The DPCCE experience. International Journal of Technical Vocational and Engineering Technology (IJTVET), 1(1), 1-14.

Kurniawan, O., Jégourel, C., Lee, N. T. S., De Mari, M., & Poskitt, C. M. (2022). Steps Before Syntax: Helping Novice Programmers Solve Problems using the PCDIT Framework. Proceedings of the Annual Hawaii International Conference on System Sciences/Proceedings of the Annual Hawaii International Conference on System Sciences. https://doi.org/10.24251/hicss.2022.121

Malik, S. I., Tawafak, R., Alfarsi, G., Ashfaque, M. W., & Mathew, R. (2021). A Model for Enhancing Algorithmic Thinking in Programming Education using PAAM. International Journal of Interactive Mobile Technologies (iJIM), 15(09), 37. https://doi.org/10.3991/ijim.v15i09.20617

Mariadass, J. D., Athesan, A. I. B., & Ghazi, M. F. E. B. M. (2020). SPHERO robotics and STEAM education in diploma level: A pilot study using SPHERO robotics module for problem solving and program design course. International Journal of Research Publications, 66(1). https://doi.org/10.47119/ijrp1006611220201610

Mulder, D. (2025). Puzzling our way into computational thinking. Journal of Technology-Integrated Lessons and Teaching, 4(1), 1–8. https://doi.org/10.13001/jtilt.v4i1.9205

Nnass, N. I., Muñoz, N. J., Cowling, N. M. A., & Hadgraft, N. R. (2023). Unlocking the potential of programming education: Enhancing conceptual understanding and student engagement with Sphero SPRK Robot. Journal of Pure & Applied Sciences, 22(3), 280–285. https://doi.org/10.51984/jopas.v22i3.2835

Ocumpaugh, J., Baker, R. S., & Rodrigo, Ma. M. T. (2015). Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) 2.0 Technical and Training Manual. *New York, NY and Manila, Philippines: Teachers …*, 1–6. Retrieved from http://www.columbia.edu/~rsb2162/BROMP.pdf

Ouyang, F., & Xu, W. (2024). The effects of educational robotics in STEM education: a multilevel meta-analysis. *International Journal of STEM Education*, *11*(1). https://doi.org/10.1186/s40594-024-00469-4

Sukirman, N., Pramudita, D. A., Afiyanto, A., & Utaminingsih, N. (2022). Block-Based Visual Programming as a tool for learning the concepts of programming for novices. *International Journal of Information and Education Technology*, 12(5), 365–371. https://doi.org/10.18178/ijiet.2022.12.5.1628

Wang, S., Liu, J., Dong, Y. Q., & Wei, Q. (2021). C programming language teaching based on CDIO. In *Proceedings of the International CDIO Conference* (pp. 474–484). Chalmers University of Technology

Wang, Y. (2019). Research on programming practice based on CDIO concept. In *Proceedings of the 2nd International Conference on Education Science and Social Development (ESSD 2019)*. https://doi.org/10.2991/essd-19.2019.54

Wikipedia contributors. (2024, August 29). *CDIO initiative*. Wikipedia. https://en.wikipedia.org/wiki/CDIO_Initiative

Yuan, X., Wan, J., An, D., Lu, J., & Yuan, P. (2024). Multi-method integrated experimental teaching reform of a programming course based on the OBE-CDIO model under the background of engineering education. *Scientific Reports*, *14*(1). https://doi.org/10.1038/s41598-024-67667-6