2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
ASEE
Paper ID #38566

Promoting Computational Thinking in Integrated Engineering Design and Physics Labs

Dr. Ruben D. Lopez-Parra, University of New Mexico

Ruben D. Lopez-Parra is a Post-doctoral fellow in the Department of Chemical & Biological Engineering at University of New Mexico. His Ph.D. is in Engineering Education from Purdue University and he has worked as a K-16 instructor and curriculum designer using various evidence-based active and passive learning strategies. In 2015, Ruben earned an M.S. in Chemical Engineering at Universidad de los Andes in Colombia where he also received the title of Chemical Engineer in 2012. His research interests are grounded in the learning sciences and include how K-16 students develop engineering thinking and professional skills through diverse learning environments. He aims to apply his research in the design of better educational experiences.

Ravishankar Chatta Subramaniam
Dr. Jason Morphew, University of Illinois, Urbana-Champaign

Dr. Jason Morphew is currently an assistant professor at Purdue University in Engineering Education and Morphew is affiliated with the INSPIRE research institute for Pre-College Engineering and the Center for Advancing the Teaching and Learning of STEM. Dr. Morphew's research focuses on the application of principles of learning derived from cognitive science and the learning sciences to the design and evaluation of learning environments and technologies that enhance learning, interest, and engagement in STEM.

Promoting Computational Thinking in Integrated Engineering Design and Physics Labs

Abstract

Computational thinking has widely been recognized as a crucial skill for engineers engaged in problem-solving. Multidisciplinary learning environments such as integrated STEM courses are powerful spaces where computational thinking skills can be cultivated. However, it is not clear the best ways to integrate computational thinking instruction or how students develop computational thinking in those spaces. Thus, we wonder: *To what extent does engaging students in integrated engineering design and physics labs impact their development of computational thinking?* We have incorporated engineering design within a traditional introductory calculus-based physics lab to promote students' conceptual understanding of physics while fostering scientific inquiry, mathematical modeling, engineering design, and computational thinking. Using a generic qualitative research approach, we explored the development of computational thinking for six teams when completing an engineering design challenge to propose an algorithm to remotely control an autonomous guided vehicle throughout a warehouse. Across five consecutive lab sessions, teams represented their algorithms using a flowchart, completing four iterations of their initial flowchart. 24 flowcharts were open coded for evidence of four computational thinking facets: decomposition, abstraction, algorithms, and debugging. Our results suggest that students' initial flowcharts focused on decomposing the problem and abstracting aspects that teams initially found to be more relevant. After each iteration, teams refined their flowcharts using pattern recognition, algorithm design, efficiency, and debugging. The teams would benefit from having more feedback about their understanding of the problem, the relevant physics concepts, and the logic and efficiency of the flowcharts.

Introduction and Theoretical Framework

Computational thinking (CT) is a critical skill for everyone to live in a world shaped by technology. Seymour Papert coined the term CT in 1986 [1], but Wing [2]spread it broadly as a fundamental skill for everyone, which involves "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" [2, p. 33]. This conception extended computational thinking beyond merely programming skills to focus on how computer scientists think when solving problems. With this definition, CT represents a way of thinking that supports inquiry in STEM disciplines, helps individuals succeed in a technological society, and enables personal empowerment [3]–[5]. CT is particularly relevant for engineers considering how new technologies based on automated systems and artificial intelligence have become predominant in the workplace [6], [7]. Furthermore, research has found a robust connection between CT and other cognitive skills such as critical thinking, spatial reasoning, and problem-solving skills [8], [9]. As a consequence,

there is a global interest, particularly within engineering education, in preparing scientists and engineers in CT [10].

Research about teaching and learning CT is still in its infancy, and currently there is not a clear definition of CT across STEM disciplines. Multiple definitions and operationalizations have been proposed making it difficult to design and assess teaching materials aimed at developing CT [11]–[16]. This lack of consensus has obstructed the development and assessment of teaching approaches to support students' development of CT [17], [18]. However, most common approaches include framing CT in terms of abstraction, automation, and decomposition [17]. In a recent comprehensive literature review Shute, Sun, and Asbell-Clarke [11] reviewed existing models of CT, then summarized the findings by advancing a model of CT that includes six facets common to most existing models.

Shute, et al. [11] defined CT as the "conceptual foundation required to solve problems effectively and efficiently with solutions that are reusable in different contexts" [11, p. 151]. This model is particularly relevant for this study, as Shute and colleagues envisioned a model for CT that was applicable across content areas and disciplines. In addition, this model was developed to facilitate assessment of CT by students engaged in problem solving tasks. CT as viewed in this model is composed of six facets: Decomposition, abstraction, algorithms, debugging, iteration, and generalization. Problem decomposition refers to breaking down a complex problem into more manageable sub-problems that when combined lead to an effective solution of the overall problem. Abstraction refers to extracting the "essence" of a complex problem to obtain the relevant information and interrelationships needed to address the problem through data collection and analysis, pattern recognition, and modeling. The CT facet of algorithms refers to the development of a logical design or ordered instructions that are needed to find efficient solutions to the problem. When this logical sequence of steps or instructions are developed to form an effective procedure, this process can be automated to solve similar problems. Debugging refers to identifying and fixing errors in the algorithm, both during the development of the algorithm and when students attempt to transfer the algorithm to a new context. Iteration is the process of revisiting effective algorithms to improve their efficiency until an optimum state is reached. Generalization occurs when the algorithms and CT skills are transferred to effectively address problems in other domains. Because iteration and generalization require the problem context to allow sufficient time for reflection and modification of the solution to be observed, we focused on the first four facets for this study. Future work will examine subsequent design challenges to find evidence for iteration and generalization.

Teaching and learning CT in non-computer science contexts has been particularly challenging in higher education. Considering the challenges with introducing new courses into the curriculum, a feasible option to promote students' CT is to infuse this skill within the current courses. Multidisciplinary learning environments such as integrated STEM courses present a potentially

transformative opportunity to help students develop CT skills within existing STEM courses while also learning disciplinary content knowledge [19]–[21]. For example, modeling is a key process in both CT and scientific inquiry that engage students in abstracting and decomposing problems [22]. Specifically, within K-12 education efforts have been made to integrate CT into physics instruction. These efforts include the development of appropriate assessments [23], [24] and the promotion of learning CT in conjunction with physic concepts [25]. In contrast, there is a lack of research in higher education about how to effectively integrate the teaching of CT into STEM instruction and how students develop CT in multidisciplinary contexts.

One promising method for facilitating the development of CT in higher education is through the introduction of flowcharting. Flowcharts have been commonly used in industry since their development by IBM in 1970 [26]. In educational settings, flowcharts have been integrated into instruction to facilitate students' learning of programming and CT in computer science and other engineering contexts [27]–[31]. The use of flowcharts supports novice programmers' learning by minimizing the cognitive load associated with learning the syntax of a new programming language and using friendly representations of programming logic and common language constructs like conditional statements or loops [29]. Previous studies have found that flowcharts can successfully scaffold students' CT by supporting students' visualization of their reasoning [4], [32], decomposing of the task [33], [34], and separating the design or logic of the solution from the coding [28]. When designing feasible algorithms, developing CT facets (i.e., decomposition, abstraction, etc.) is often more relevant than writing correct syntax [35]–[37]. This study focuses on examining student produced flowcharts as the aim of the study was to analyzing students' development of CT rather than students' learning of coding.

Laboratory experiences in undergraduate physics courses are full of opportunities to engage students in CT. The integration of CT and engineering design (ED) can foster students' learning during these experiences (e.g., [38]–[41]). However, there is a need for further research about the best ways to integrate CT and ED in undergraduate STEM courses and to examine how students develop these skills within multidisciplinary STEM contexts [20], [21]. Following calls to create more integrated and impactful laboratory experiences in undergraduate physics [42], [43], we integrated CT and ED into a traditional introductory undergraduate physics lab for engineering and science majors. The goal of this multidisciplinary approach to teaching physics was to promote students' conceptual understanding of physics while fostering scientific inquiry, mathematical modeling, ED skills, and CT. In this context, we proposed the following research question to better understand the undergraduate students' learning of CT in a multidisciplinary STEM environment:

To what extent does engaging students in integrated engineering design and physics labs impact their development of computational thinking?

Methods

We used a generic qualitative research approach to explore the students' CT in the integrated engineering design and physics labs. Generic qualitative research is not guided by a specific set of philosophic assumptions and provides flexibility when the study's focus or the kind of data does not fit a typical qualitative approach [44]. This research approach is appropriate for this study due to the exploratory nature of this work and the complexity of our data. Namely, we looked for evidence of CT in the flowcharts developed by six teams of undergraduate students when addressing an ED challenge. The following sections provide details about the context, settings, participants, data collection and analysis, and limitations of this study.

Context

This study was part of a larger pedagogical intervention that aimed to transform the laboratory experiences in a large-enrollment first-semester calculus-based undergraduate physics course at a large U.S. midwestern land-grant university by integrating CT and ED. The physics course has an annual enrollment of about 2400 students, about 80-90% of which are engineering majors, and the remaining students are primarily physics or other natural science majors. The course adopts the principle-based approach that is followed in the course text: *Matter and Interactions* (*M&I*) vol. 1 [45]. Specifically, it covers the (linear) momentum, energy, and angular momentum principles with a focus on systems thinking, modeling, assumptions, and approximations. The course's weekly schedule includes two 50-minute lectures, one 120-minute laboratory, and one 50-minute recitation focused on problem-solving. The laboratory segment has a total of 13 sessions throughout the semester. This study focused on the sessions 2 through 6.

Laboratory setting

The design of the lab sessions has undergone several changes over the last four years to integrate ED, CT and scientific inquiry into the physics labs. Each lab session included around 50 students working in groups of two or three, was led by a graduate teaching assistant, and was supported by an undergraduate teaching assistant. The graduate teaching assistants were trained in leading ED-integrated physics labs at the beginning of the semester, attended weekly meetings, and performed the labs in advance to discuss the best practices for instructing students. Each lab session typically began with a brief introduction to the days' lab by the graduate teaching assistant. Students then worked in teams to complete the ED as long as learning CT, and physics concepts following printed instructions.

Integration of ED. Our more recent intervention integrated ED through challenges that followed seven essential ED characteristics [46]. The seven ED characteristics are: being client-driven and goal oriented, having an authentic context, stating the constraints clearly, making use of

materials, resources and tools familiar to students, developing a prototype that may be a product or a process, generating multiple solutions, and engaging in teamwork.

Integration of CT. CT tasks consisted of working with Jupyter Notebooks hosted by Google Colab and creating flowcharts to document their algorithms. Apart from being free for anyone with a Google account, these Notebooks combine executable code and rich text in a single document. These are particularly useful for running Python codes, typing equations in Latex, uploading images and CSV files, performing data analysis, creating, and sharing files without having to download or install anything. Students were not expected to have prior programming experience, as the focus of the CT tasks were on applying physics concepts to edit and modify existing code relevant to the physical concept of the lab. Carefully designed scaffolds were provided for the students to facilitate learning elements of programming in the context of the laboratory experiment.

Integration of scientific inquiry. The physics tasks consisted of hands-on activities using PASCO equipment and PASCO Capstone software, which are aligned with the three principles defined by course textbook (i.e., linear momentum, energy, and angular momentum principles), along with guided activities in Python programming.

Students engaged in two ED challenges throughout the laboratory sessions. This study focused on the first ED challenges that worked as an introduction to design experience for the students. The next section describes in detail this first design challenge.

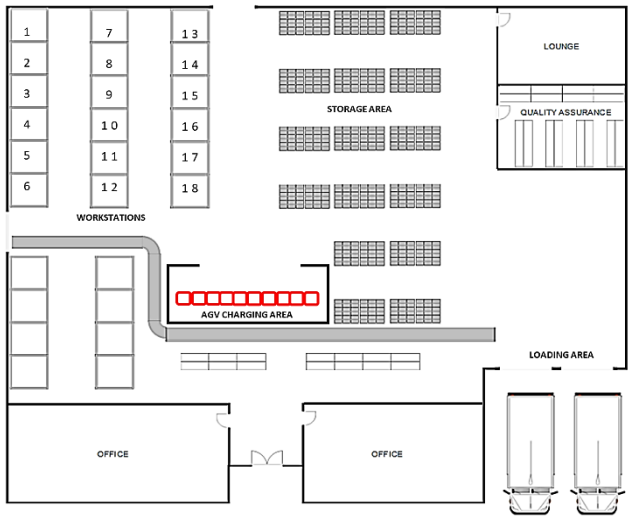
Engineering Design Challenge

Students completed the first design challenge in labs 0 through 5. Lab 0 consisted of a tutorial for the students to get familiar with the use of Jupyter Notebooks and was a work-from-home task. Lab 1 introduced the ED challenge to the students (Figure 1) and presented a short YouTube video on robots being used to move packages in an Amazon workstation to help them gain an understanding of a real-life situation. Based on this information, they started framing the problem by identifying the client, stakeholders, end-users and requirements. Then, students worked on a hands-on physics experiment based on the concept of 1-D motion with constant speed.

A partner shipping company is interested in making its operations more efficient using modern technology. One of their more time-consuming operations is handling the packages inside the facilities. Thus, they want to implement Automated Guided Vehicles (AGVs) at their new warehouse. AGVs are a class of robots used for transporting items directly to the workstations where human workers prepare the items for shipment to the customers. The company is requesting your team to develop an algorithm to remotely control its AGVs in a timely and safe manner. They will review your proposal and send the algorithm to a team of expert programmers who will implement it in Python. They will need a visual representation (e.g., flowchart) of your algorithm with a written description of how it works and any additional information that you consider relevant.

Figure 1. First ED challenge proposed to the students as part of their physics lab.

In Lab 2, students were provided with further details about the constraints of the problem and a layout to help them further frame their problem (See Figure 2). Students were introduced to the idea of flowcharts as a tool to visually represent algorithms. Then, based on an example in the lab instructions; students designed the first flowchart as a team to visually represent the logic of an algorithm to remotely control automatic guided vehicles (AGVs). After that, students worked of accelerated motion in 1-D.



Your algorithm should at least ensure the AGV will pick up a variable number of items from the loading area, take them to a specific workstation, and go back to the loading area. For your reference, the company provides the layout of one of its warehouses. Each trip must take the minimum time and always be safe for both robots and workers. In case of a collision with a wall, the AGV breaks when it has a momentum of 220 kg*m/s or more. Additionally, the algorithm should be as automatic as possible, requiring minimum human interaction.

Figure 2. Problem constraints and layout provided to the students.

In labs 3 and 4, the teams completed more traditional physics labs for most of the lab time.

However, at the end of each lab, they were asked to discuss the physics concepts and principles relevant to the ED challenge present in the physics lab, the assumptions and approximations that may have been made in their ED, and to document the changes in their ED as a result of the lessons learned during the lab. See Table 1 for additional details about the ED, CT and physic tasks that were integrated each week. (See Table 1). Finally, in lab 5, the teams designed their final flowcharts using concepts learned from the lab completed over the previous three weeks and wrote a technical report for the client. This report required the teams to reflect about their CT and ED process by analyzing the evolution of their flowchart throughout the five lab sessions.

Table 1. Main tasks performed by the students throughout the first ED challenge.

Lab #	CT & ED Tasks	Hands-on Physics Tasks
0	• Jupyter Notebook - a tutorial	• None
1	• Presenting the ED problem: To develop an algorithm to remotely control automatic guided vehicles (AGVs)	• Uniform 1-D motion: Position update using PASCO carts.
2	• Continue the ED problem: What are criteria and constraints? • Introduce the concept of flowchart	• Accelerated motion in 1-D: Momentum update using PASCO carts.
3	• Present criteria and constraints: (1) Every move must be always safe and (2) each trip of an AGV must take minimum time and • Further work on first constraint: Use metrics to evaluate the AGV safety • Update flowchart to incorporate criteria, constraints and metrics	• Smart-Fan accessory mounted on a PASCO cart: Momentum Principle
4	• Further work on second constraint: Find out the AGV path that would minimize the movement time • Update the flowchart	• Smart-Fan accessory mounted on a PASCO cart: speeding up and slowing down
5	• Further work on second constrain: Use a python simulation to analyze 2-D motion with variable acceleration • Complete ED challenge: Final flowchart is developed • Reflect about the ED process	• None

Participants

The participants of this study were 17 undergraduate students organized in five teams of three students (Teams 1, 2, 3, 5, and 6) and one team of two students (Team 4). All teams were part of the same lab section that had around 50 students in total. Most of the participants were first-year engineering students. Teams 1 to 4 were chosen randomly, and teams 5 and 6 were chosen because one author of the present study observed their performance during the lab sessions. The

graduate teaching assistant who leads the lab sessions had taught the class once before and the undergraduate teaching assistant had taken the class previously.

Data collection and analysis

Our main data source was the students lab reports. Namely, we collected the five lab reports associated with the first ED challenge of each team. These reports included the students answers to the ED tasks, CT tasks, and physics tasks. For this study, we focused on the flowcharts developed for each team. We analyzed four flowcharts per team for a total of 24 flowcharts. Additionally, to better understand the context of the lab, one researcher observed the five lab sessions, and took notes about the general structure of the lab, the interaction of the teaching assistants with the students, and the performance of two teams (Team 5 and 6).

The teams flowcharts were coded throughout several rounds of open coding using the CT model of Shute, et al. [11]. Initially, two researchers open-coded the 24 flowcharts looking for evidence of the CT according to the model of Shute, et al. Then, a third researcher expert on students' learning discussed the initial codes looking for relationships and potential emerging themes. After this conversation, the two initial researchers performed a second round of coding based on the initial codes to identify the evidence associated with the CT facets of problem decomposition, abstraction, algorithms, and debugging.

Limitations

We identified a few limitations in this study. As may be evident, the representative sample size is quite small. This study does not aim to find generalizable results; instead, we provide rich descriptions aiming for the transferability of the results to similar integrated physics lab contexts. Also, because the labs are completed in teams, the flowchart is a product of a collaborative learning environment, and therefore individual development of CT cannot be identified. Finally, since we did not observe teams as they were creating their flowcharts, the rationale for student decisions in creating the flowchart are not always clear. We plan to collect process data in future research to better understand the students decision making when producing the flowcharts.

Findings

This study explored the extent to which undergraduate students developed CT skills when engaging in a physics lab integrated with ED and CT tasks. The six teams included in this study demonstrated development across four facets of CT (i.e., decomposition, abstraction, algorithm design, and debugging) while working on designing their flowcharts. The following paragraphs describe evidence for how teams developed within each CT facet throughout the iterations of their flowchart.

Problem decomposition

Problem decomposition was mostly evident in the initial flowchart created in lab 2 of the design challenge. Teams decomposed the problem of remotely controlling the movement of an AGV into a few sub-problems (see Figure 3). Namely, they broke the larger problem into the following sub-problems in their first flowchart: charge the AGV, AGV receives instructions, AGV moves to destination, AGV senses the load, AGV picks up items.

In addition to the sub-problems associated with moving the AGV, Team 2 identified contextual aspects of the problem, such as the presence of big slopes that may impede the AGV from transiting across the warehouse. This focus seems to indicate that the team was positioning the problem from the clients' perspective by thinking about features of the warehouse that may become a problem for the AGV. However, they decided to remove the slope verification in their second flowchart. In doing so, this team may not have considered the sub-problem of sloping of the floor to be as relevant for the algorithm as other constraints. That lab session focused on the requirements of safety and timely movement; thus, students may have been persuaded to only focus on those requirements. Students could have been encouraged to integrate the context of the problem more into their algorithms.

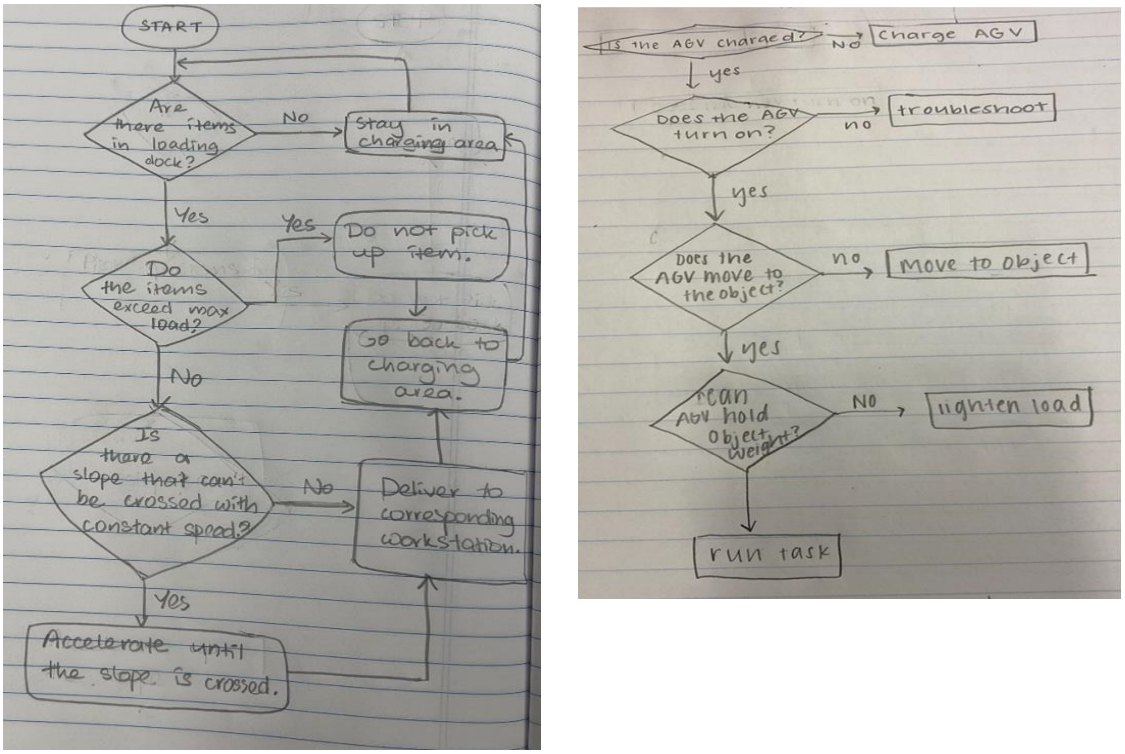


Figure 3. Initial flowcharts developed by Team 2 (left) and Team 1 (Right) showing different ways to decompose the problem and different levels of algorithm design.

The ability for teams to engage in problem decomposition varied greatly in the initial flowcharts; however, all teams ended up breaking up the subproblems into actionable tasks that could be operationalized by a team of programmers (i.e., the end-user of the algorithm) by the end of the design challenge. The time needed for teams to fully decompose the larger problem into actionable sub-problems varied. While Teams 1, 2, 3, and 6 presented only the sub-problems in their first flowchart, Teams 4 and 5 identified actionable tasks that would solve some sub-problems. For instance, instead of stating the sub-problem "move to object," Team 5 identified at least three actionable tasks associated with that sub-problem: Determine the current position, calculate the fastest path, and drive to the destination. In identifying the actionable tasks required to address each sub-problem, Team 5 was able to develop an initial flowchart that took other teams at least two or three more sessions to develop. As such, Team 5 may have been very engaged in the activity and wanted to advance as much as possible in the beginning of the design challenge, or they may have come into the lab course with more experience or proficiency in CT.

Abstraction

Abstraction involves extracting the most relevant information to address the problem and often is accompanied by data collection and pattern recognition. To make progress on the design challenge, teams needed to extract information in the form of constraints to design an algorithm that would fulfill the client's needs and adhere to physical constraints imposed by physical laws and principles. The lab sessions scaffolded students in recognizing relevant contextual data and physics principles to integrate into the flowcharts. The following paragraphs describe how teams exhibited abstraction in their flowcharts throughout the five lab sessions.

The teams developed their initial flowchart during lab 2. As the last section, "Problem decomposition," mentioned, the teams mostly decomposed the design problem into subproblems during the initial flowchart. Their first flowcharts showed how teams defined several constraints based on the general problem statement using the provided data and their previous experiences. For example, Team 5 used video information provided as part of the lab to extract how AGVs lift objects inside the warehouse. Namely, they identified the sequence: "Board truck via ramp," "drive below pallet," "Lift pallet vertically," "exit truck via ramp." Even though the problem did not specify the constraint that the AGV should move towards a ramp and then lift the load, Team 5 defined and integrated these additional constraints in their initial flowchart. In a similar way, other teams added actionable tasks in their algorithm to ensure that the AGV would emit an alarm before hitting something, stop when encountering an obstruction, or determine if the AGV encountered a steep slope. The integration of these additional constraints found outside of the problem context helped to create a more realistic and feasible algorithm to address the design challenge and meet the client's needs and context.

During subsequent lab sessions, the teams continued to develop their flowcharts using abstraction and drew on the traditional physics labs to integrate physics concepts into their flowcharts. Labs 3 and 4 scaffolded the teams' ability to implement physical constraints by connecting physics concepts with design criteria from the problem context. As an example, the problem statement of the design challenge required the teams' algorithms to ensure that their AGV would move safely through the warehouse and make timely deliveries. In the case of a safe movement, the problem stated that the AGV should not carry a momentum higher than 220 Kg*m/s any moment. During lab 3, students engaged in hands-on experiments aimed at examining the motion of the PASCO cart with constant acceleration. However, teams were not given explicit instructions about how to integrate the momentum constraint in their flowcharts. Teams took three different approaches to integrate their conceptual physics knowledge with the design constraint (see Table 2). Out of the three approaches observed by these teams, the third approach would be desirable (variable mass and speed), and result in a more feasible flowchart. However, not all teams reached this level of integration. This result implies that to fully progress with abstracting this aspect of the flowchart, students may need additional feedback concerning the optimization and feasibility of their flowchart in addition to the experimental investigations.

Table 2. Strategies used by the teams to integrate the concept of momentum into their flowcharts.

Integration of momentum metric	Description
Momentum as a measurable entity	The teams added the question "Is the momentum less than 220 Kg*m/s?" This basic approach would assume that the AGV can sense the momentum. Teams 2 and 4 started with this approach in their first flowchart and changed to a more sophisticated way of integration for the second. Team 1 presented this approach in the third flowchart and did not improve it.
Constant mass and variable speed	Even though the problem stated that the AGV should carry a variable load less than 340 Kg, Teams 2, 3, and 4 assumed the load should be always maximum. Thus, they included in their third flowchart that the AGV would only sense its speed and calculate its momentum. Based on the momentum, the speed would be increased or decreased.
Variable mass and speed	Teams 5 and 6 defined in their second flowchart that AGV should sense the mass when charging the load; then, it should also sense the speed and calculate the momentum. Based on the result, the AGV would increase or decrease its speed.

Most teams implemented the design constraint of timely deliveries in the second and third iterations of their flowcharts by drawing on the data simulation tasks required by the physics labs. During labs 4 and 5, teams used a simulation to determine the time required for a simulated AGV to move throughout the 18 workstations (see layout in Figure 2). They analyzed the data generated by the simulations and ranked the stations in order by the time required to travel to each station. All of the teams, except for Team 1, integrated the information from the simulations into their flowchart as an input for the AGV to decide the order of the workstations that the AGV should use to deliver items to minimize travel time. While Team 1 failed to incorporate the

lessons from the simulations into their flowchart, this team did add an actionable task, “Bring load to the nearest workstation,” into their final flowchart. It may be that Team 1 did not consider necessary to include the list of workstations into the algorithm, or they may have failed to notice that delivering to the nearest workstation may not result in the fastest overall trip duration. Overall, teams demonstrated some development in their ability to abstract the essence of the problem in the form of constraints, then to integrate these constraints into the flowcharts. In addition to the variation associated with the extracted information, teams organized the actionable tasks in their algorithms using different sequences, which is explored in the next section.

Algorithms

The CT facet of algorithms includes designing logical and ordered instructions to address the problem efficiently. Teams were asked to develop an algorithm with a logical sequence of actionable tasks to remotely control the movement of an AGV inside the clients’ warehouse, as summarized in the flowcharts. The teams’ initial flowcharts exhibited a variety of logic structures ranging from linear sequences to basic loops. For example, while Team 1 presented a linear sequence of 4 steps to address the problem, Team 2 constructed a complex sequence of steps with loops (see Figure 3). Throughout the design challenge, all teams developed more complex flowcharts in terms of the number of included steps, the number of selection statements (e.g., if-then), and the number of loops. For instance, Team 5’s flowchart evolved from an initial flowchart with a complex structure similar to the presented by Team 2 to their final flowchart, which incorporated two flowcharts so that the algorithm operated more efficiently, and the flowcharts were easier to understand (see Figure 4).

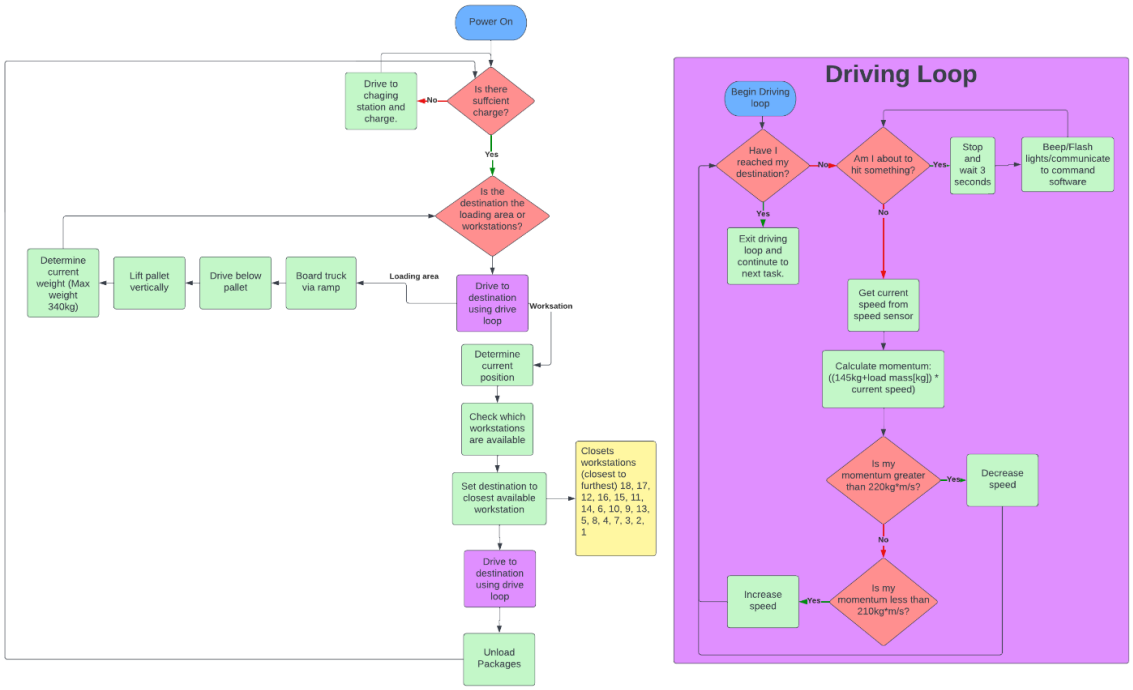


Figure 4. Final algorithm developed by Team 5 and represented using two flowcharts.

Although the final flowcharts incorporated complex structures, integrated both design and physical constraints, and demonstrated integration of physics concepts into the algorithm, most of the flowcharts still exhibited problems with logic, such as the presence of disordered instructions or abrupt ends where the algorithm would break down. For example, Team 3's final flowchart indicated that the AGV would begin at the charging area, where it receives the order to pick up an item (see Figure 5). The AGV then moves to the loading area, where it picks up the item. Finally, when moving to the workstation, the AGV would accelerate to the maximum speed to meet the momentum criteria, but only for an item at the maximum weight. In this flowchart, the team did not consider that the speed should be monitored and regulated during the initial movement of the AGV, from the charging station to the loading area. In addition, the team appears to fail to recognize that both the weight and speed need to be monitored and covary when regulating the AGVs momentum. Finally, their flowchart included an abrupt end when the load was greater than the maximum weight allowed. Consequently, even though the flowchart includes many of the required elements (i.e., safe and timely movement), a programmer, as a final user, would have to reinterpret the flowchart in order to use it. Students may have needed more scaffold to understand the level of detail that a useful flowchart for the final user should have.

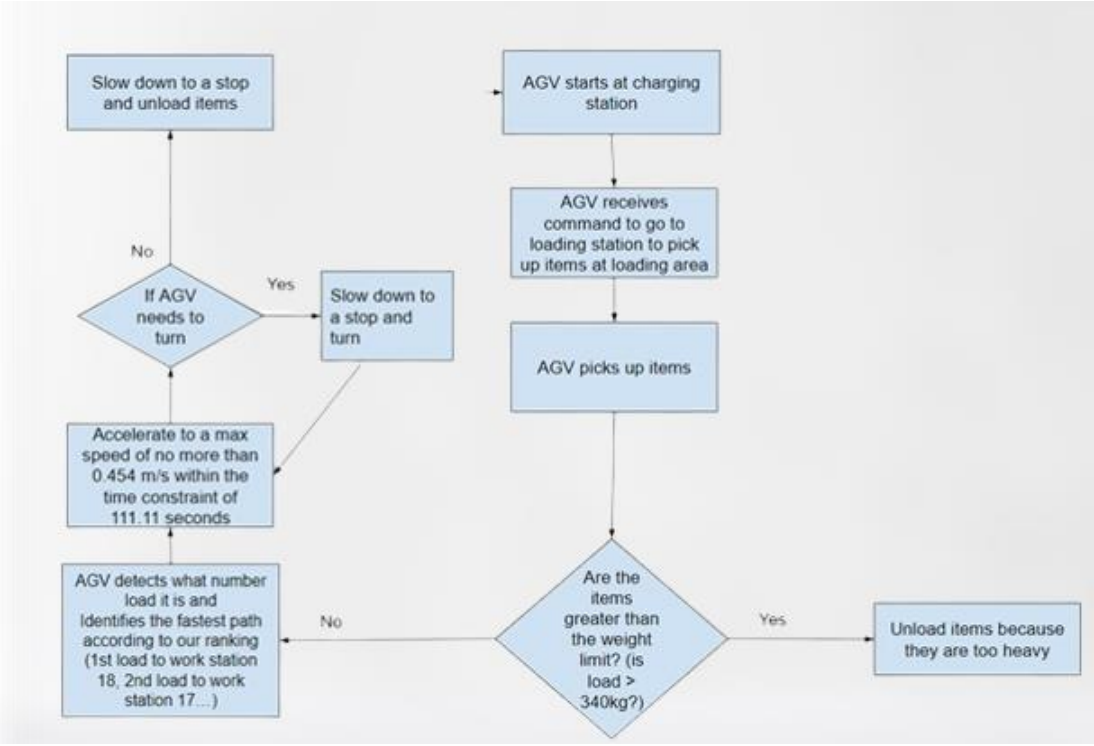


Figure 5. Final flowchart developed by Team 3 including abrupt ends.

Debugging

Debugging involves identifying and fixing errors in the algorithm. For the teams in this study, debugging was mostly evident in the finalization of the flowchart rather than as an aspect employed in every iteration. After completing the initial flowchart, teams mainly focused on integrating new information in the second and third iterations of the flowcharts rather than on identifying errors in their logic or flowcharts and correcting those errors. Conversely, the teams demonstrated debugging in the third and the final flowcharts. The debugging mainly focused on correcting information of the actionable tasks or their sequence. For example, as Figure 6 shows, Team 1 had abrupt ends present in their initial flowchart, which was uncorrected in subsequent iterations. In the final flowchart, however, Team 1 corrected their flowchart by adding new loops, thus preventing the AGV from being stuck at the end of one delivery.

The labs were designed to facilitate and encourage reflection, giving students the opportunity to correct their flowcharts in each iteration. However, teams instead focused on adding new information such as the momentum criterion rather than reflecting on and evaluating previous iterations. The flowcharts were not rigorously graded, and in the lab sections, the teaching assistants emphasized completion rather than accuracy, which may have resulted in students being less motivated to correct previous flowcharts. In contrast, between the third and the final iterations, the teaching assistants encouraged students to present a final accurate solution. This shift in the messaging from the teaching assistants appears to have encouraged students to take the time to focus on debugging and improving their flowchart.

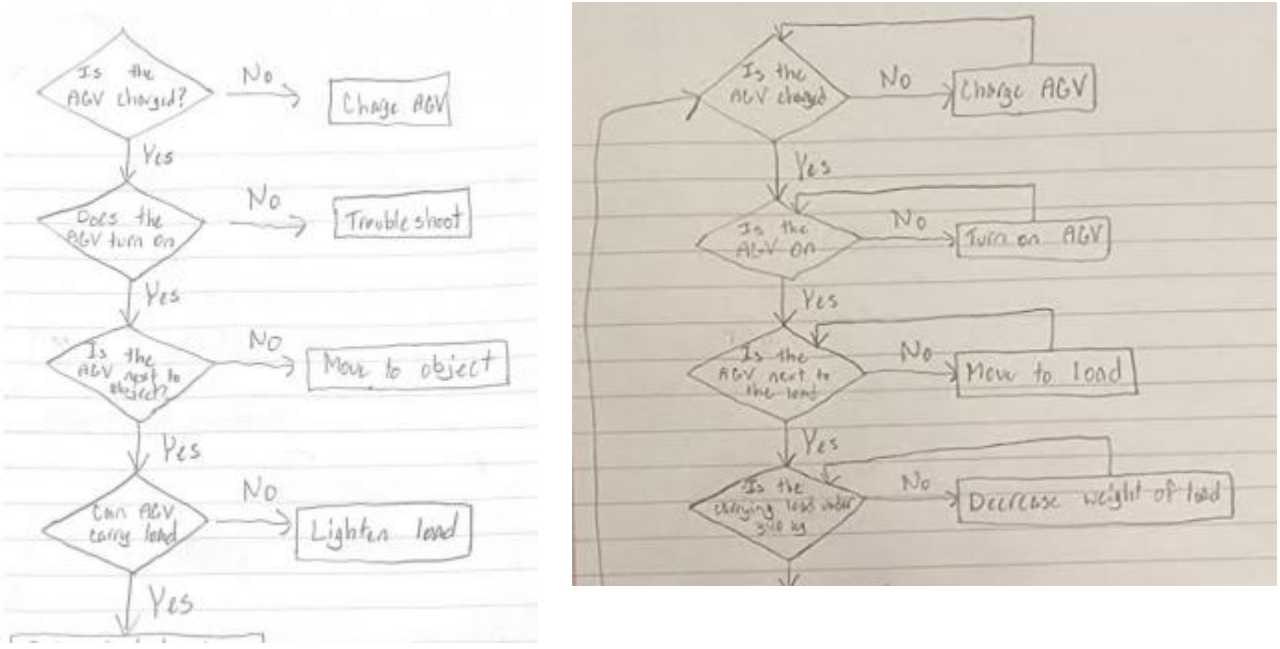


Figure 6. Team 1 flowcharts 3 (left) and 4 (right) where they corrected the abrupt ends.

Discussion, Conclusions, and Implications

The participants demonstrated a progression of CT across the four facets examined in this study as evidenced by the content and sequence of the actionable tasks included in the flowcharts. Overall, teams started the design challenge by engaging in a superficial decomposition of the main problem by describing the sub-problems as actionable tasks without paying much attention to the logic of the sequence of steps. Given that this was the first-time students were asked to engage in a design challenge in the course, this initial superficiality is not surprising.

Although decomposing the problem is one of the most challenging CT facets, using a familiar context may help students to perform it [16]. The AGVs' motion in a warehouse provided a context close enough to the students' life; so, they could engage in the ED challenge and start decomposing the problem in the first lab session. The teams' initial flowcharts present an opportunity to support instructors in determining how well students understand the design problem, identifying deviation from canonical scientific understanding, and encouraging students to consider other perspectives (e.g., the clients and stakeholders) on the design problem the algorithm is meant to address.

In subsequent labs, students demonstrated abstraction in the flowcharts by collecting and analyzing data relevant for the problem. Across the five weeks, a progression from using data associated with the context of the problem to using mathematical equations and physics concepts to make the solution more feasible was observed. For example, teams first identified and integrated information related to the warehouse layout. Then subsequent weeks, the teams identified and integrated information related to physics concepts such as the momentum or position update requirements to ensure a safe and timely movement. The teams observed in this study quickly grasped and applied the contextual information from the problem documents; however, they needed time and scaffolding to correctly integrate the concept of momentum. Applying the concept of momentum to an authentic design problem seems to add a layer of complexity that teams struggle to incorporate into their algorithms. One reason that applying momentum to solve this design problem is complicated, is that while the equation only contains two variables, the variables must both be monitored and adjusted as a pair, because a change in one variable will impact the maximum value the other value can take on. Thus teams need to decompose the concept of momentum into the variables of velocity and mass and co-locate the monitoring and setting of the values for these variables logically inside the flowchart.

In constructing the flowcharts, teams prioritized integrating information into the flowcharts over evaluating the logic underlying the flowchart sequences. Throughout the lab sessions, teams mostly engaged in three of the five processes of programming followed by computer science students: abstract the problem from its description, generated sub-problems, transform sub-problems into sub-solutions [37]. They were not fully involved in the last processes of

recomposing the sub-solutions into a working program and evaluating and iterating to produce an optimum solution. Given this result, the final flowcharts still contained some errors, resulting in flowcharts that were not optimal or efficient. One reason for this is that teams seemed to focus on integrating the correct information rather than analyzing the logic of the flowchart. Flowcharts needs to be simple enough to be practical and complex enough to represent clearly the algorithm [31]. The teams seemed to lack the experience with ED and CT to accurately self-evaluate the level of detail needed in the flowcharts to make them practical but still complete.

The influence of the course context likely contributed to the over reliant focus on information over algorithm logic. Although the laboratory sessions aimed to integrate physics with ED while developing CT, the grading of the lab reports likely led teams to prioritize including the physics content in the flowcharts over spend time iterating on the flowchart to optimize the logic of the algorithm and generalizing the algorithm so that it could be applied to solve other problems. Furthermore, the initial grading of the flowcharts did not focus on quality or logic, therefore teams did not receive specific feedback to help scaffold the processes of iteration and generalization. One conclusion from these observations is that finding a balance between the learning objectives and aims of multiple domains or disciplines when working in integrated contexts will naturally lead to a prioritization of the goals of one of the disciplines for any given task. As such, it is important to view integrated STEM instruction as helping students to identify connections between disciplines and developing the skills necessary to transfer the skills and practices between the disciplines to solve authentic and multidisciplinary problems.

Providing continuous and immediate feedback is fundamental for promoting student learning. This is especially true in interdisciplinary settings where the learning outcomes are to develop CT, engage in ED, develop scientific inquiry skills, and learn and apply physics principles in authentic contexts. In this study, teams worked independently during most of the lab time without communicating with other teams or the teaching assistants. Most of the interactions with the teaching assistants consisted of asking questions about the step-by-step procedures of the lab, or practical questions about use of the lab equipment. Conceptual discussions concerning the physics concepts were presented at the beginning of the labs, and occasionally between individual teams and the teaching assistants, however discussions concerning CT or ED were relatively few and brief. Part of this lack of discussion of engineering and mathematical practices is that students and teaching assistants tend to take an epistemological stance that disciplines are siloed and should be learned in isolation [40], [41]. Then, they may perceive that developing CT and ED are not as important as completing the step-by-step procedures of the lab. Future studies could explore the students' perceptions of the importance of learning CT, ED, scientific inquiry, and physic principles in the laboratory context.

A key engineering practice within ED is communication with clients and stakeholders to identify and scope the problem, and to receive feedback about the sufficiency of the proposed solution. In

the AGV problem, teaching assistants played the role of the client and iterative discussions with the teaching assistants simulating clients would benefit students during debugging and iteration. As such, we think that providing space for reflection and iteration is a critical aspect of integrated STEM labs and time for these discussions should be included as an explicit feature in integrated design challenges. As part of the design challenge, students should be encouraged to discuss their ideas with teaching assistants and other teams. Particularly critical for this design challenge is the need for teaching assistants to play the role of programmers (i.e., the final user of the algorithm) to facilitate and scaffold iteration, evaluation, and generalization. During these conversations, students need feedback about their understanding of the problem, the relevant physics concepts, and the logic and efficiency of the flowcharts. In addition, teams need time to reflect on the efficiency of their flowcharts to revise and improve both the effectiveness and efficiency of the processes they are designing. These additions to the design challenge may give teams a better idea of the details expected of an efficient algorithm, promote the importance of the flowchart activity, and help students understand the importance of science, engineering, and mathematical practices as part of the laboratory.

Acknowledgements

The authors would like to thank Dr. Sanjay Rebello, Dr. Carina Rebello and Mr. Amir Bralin for their support with the design of the learning materials and the project logistics. This work was partially funded by the National Science Foundation under Grant No. DUE 2021389. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Papert, *Mindstorms. Children, computers and powerful ideas*. New York, NY: Basic Books, 1980.

[2] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006, doi: 10.1145/1118178.1118215.

[3] A. Çiftçi and M. S. Topçu, "Improving early childhood pre-service teachers' computational thinking teaching self-efficacy beliefs in a STEM course," *Res. Sci. Technol. Educ.*, pp. 1–27, 2022, doi: 10.1080/02635143.2022.2036117.

[4] J. Zhang, B. Meng, L. Zou, Y. Zhu, and G. Hwang, "Progressive flowchart development scaffolding to improve university students' computational thinking and programming self-efficacy," *Interact. Learn. Environ.*, pp. 1–18, 2021, doi: 10.1080/10494820.2021.1943687.

[5] C. Vieira, M. Penmetcha, A. Magana, and E. Matson, "Computational thinking as a practice of representation: A proposed learning and assessment framework," *J. Comput. Sci. Educ.*, vol. 7, no. 1, pp. 21–30, 2016, doi: 10.22369/issn.2153-4136/7/1/3.

[6] Deloitte, "2018 Deloitte and the manufactoring institute skills gap and future of work study," UK, 2018. [Online]. Available: https://www.themanufacturinginstitute.org/research/2018-deloitte-and-the-manufacturing-institute-skills-gap-and-future-of-work-study/

[7] World Economic Forum, “The future of jobs report,” Switzerland, 2018.
[8] H. Y. Durak and M. Saritepeci, “Analysis of the relation between computational thinking skills and various variables with the structural equation model,” *Comput. Educ.*, vol. 116, pp. 191–202, 2018, doi: 10.1016/j.compedu.2017.09.004.
[9] M. Román-González, J.-C. Pérez-González, and C. Jiménez-Fernández, “Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test,” *Comput. Hum. Behav.*, vol. 72, pp. 678–691, 2017, doi: 10.1016/j.chb.2016.08.047.
[10] T.-C. Hsu, S.-C. Chang, and Y.-T. Hung, “How to learn and how to teach computational thinking: Suggestions based on a review of the literature,” *Comput. Educ.*, vol. 126, pp. 296–310, Nov. 2018, doi: 10.1016/j.compedu.2018.07.004.
[11] V. J. Shute, C. Sun, and J. Asbell-Clarke, “Demystifying computational thinking,” *Educ. Res. Rev.*, vol. 22, pp. 142–158, 2017, doi: 10.1016/j.edurev.2017.09.003.
[12] P. Curzon, M. Dorling, T. Ng, C. Selby, and J. Woollard, “Developing computational thinking in the classroom: a framework,” *Computing At School*, 2014. http://eprints.soton.ac.uk/369594/10/DevelopingComputationalThinkingInThe ClassroomaFramework.pdf
[13] S. Grover and R. Pea, “Computational thinking in K-12: A review of the state of the field,” *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013, doi: 10.3102/0013189X12463051.
[14] F. Kalelioglu, Y. Gulbahar, and V. Kukul, “A framework for computational thinking based on a systematic research review,” *Balt. J. Mod. Comput.*, vol. 4, no. 3, pp. 583–596, 2016.
[15] D. Weintrop *et al.*, “Defining Computational Thinking for Mathematics and Science Classrooms,” *J. Sci. Educ. Technol.*, vol. 25, no. 1, pp. 127–147, 2016, doi: 10.1007/s10956-015-9581-5.
[16] C. C. Selby, “Relationships: Computational thinking, pedagogy of programming, and Bloom’s taxonomy,” in *Proceedings of the Workshop in Primary and Secondary Computing Education*, New York, NY, 2015. doi: 10.1145/2818314.2818315.
[17] J. A. Lyon and A. J. Magana, “Computational thinking in higher education: A review of the literature,” *Comput. Appl. Eng. Educ.*, vol. 28, pp. 1174–1189, 2020, doi: 10.1002/cae.22295.
[18] D. E. Sondakh and K. Osman, “Reflecting on CT studies in secondary education: Informal program,” in *Proceedings of the 4th International Science, Mathematics and Technology Education Conference (ISMTEC 2018)*, Bangkok, Thailand, Oct. 2018.
[19] Y. Lin, M.-T. Wang, and C.-C. Wu, “Design and implementation of interdisciplinary STEM instruction: Teaching by computational physics,” *Asia-Pac. Educ. Res.*, vol. 28, no. 1, pp. 77–91, 2019, doi: 10.1007/s40299-018-0415-0.
[20] C. Wang, J. Shen, and J. Chao, “Integrating computational thinking in STEM education: A literature review,” *Int. J. Sci. Math. Educ.*, vol. 20, no. 8, pp. 1949–1972, 2022, doi: 10.1007/s10763-021-10227-5.
[21] Y. Li *et al.*, “On computational thinking and STEM education,” *J. STEM Educ. Res.*, vol. 3, no. 2, pp. 147–166, 2020, doi: 10.1007/s41979-020-00044-w.
[22] P. H. M. Sins, E. R. Savelsbergh, and W. R. van Joolingen, “The difficult process of scientific modelling: An analysis of novices’ reasoning during computer-based modelling,” *Int. J. Sci. Educ.*, vol. 27, no. 14, pp. 1695–1721, 2005, doi: 10.1080/09500690500206408.
[23] R. Zakwandi, “A framework for assessing computational thinking skills in the physics classroom: study on cognitive test development,” *SN Soc. Sci.*, vol. 3, no. 46, pp. 1–15, 2023, doi: 10.1007/s43545-023-00633-7.

[24] Y. Yin, R. Hadad, X. Tang, and Q. Lin, "Improving and assessing computational thinking in maker activities: The integration with physics and engineering learning," *J. Sci. Educ. Technol.*, vol. 29, pp. 189–214, 2020, doi: 10.1007/s10956-019-09794-8.
[25] H. A. Dwyer, B. Boe, C. Hill, D. Franklin, and D. Harlow, "Computational thinking for physics: Programming models of physics phenomenon in elementary school," in *Proceedings of the 2013 Physics Education Research*, Portland, OR, Feb. 2014, pp. 133–136. doi: 10.1119/perc.2013.pr.021.
[26] IBM, "Flowcharting techniques." International Business Machines Corporation, 1970.
[27] D. Zheng, G. Yang, J.-H. Chen, J.-W. Li, L.-L. Zheng, and X.-L. Li, "Improving students' python programming abilities: An instructional method based on gap-filling flowchart scaffolding," in *Proceedings of the 2022 IEEE 2nd International Conference on Educational Technology (IEEE ICET 2022)*, Beijing, China, Jun. 2022, pp. 203–207. doi: 10.1109/ICET55642.2022.9944456.
[28] C. Cabo, "Effectiveness of flowcharting as a scaffolding tool to learn python," in *Proceedings of the 2018 IEEE Frontiers in Education Conference (FIE)*, San Jose, CA, Oct. 2018, pp. 1–7. doi: 10.1109/FIE.2018.8658891.
[29] D. D. Cook, "Flowgorithm : Principles for teaching introductory programming using flowcharts," in *Proceedings of the 2015 American Society for Engineering Education/Pacific South West Conference*, Pomona, CA, 2015, pp. 158–167.
[30] S. M. Dol, "Animated flowchart with example followed by think-pair-share activity for teaching algorithms of engineering courses," in *Proceedings of the 2018 IEEE Tenth International Conference on Technology for Education (T4E)*, Chennai, India, Dec. 2018, pp. 186–189. doi: 10.1109/T4E.2018.00048.
[31] R. Smetsers-Weeda and S. Smetsers, "Problem solving and algorithmic development with flowcharts," in *Proceedings of the 12th Workshop in Primary and Secondary Computing Education*, Nijmegen, Netherlands, Nov. 2017.
[32] D. Giordano and F. Maiorana, "Teaching algorithms: Visual language vs flowchart vs textual language," in *Proceedings of the 2015 IEEE Global Engineering Education Conference (EDUCON)*, Tallinn, Estonia, Mar. 2015, pp. 499–504. doi: 10.1109/EDUCON.2015.7096016.
[33] J. M. Chambers, M. Carbonaro, and M. Rex, "Scaffolding knowledge construction through robotic technology: A middle school case study," *Electron. J. Integr. Technol. Educ.*, vol. 6, pp. 55–70, 2007.
[34] N. Ibrahim, N. F. S. Saifuzzin, A. A. Seman, N. A. Wahab, and A. Osman, "Flowchart discovery game for basic programming course (Flowgame)," *J. Appl. Fundam. Sci.*, vol. 10, no. 1S, pp. 1109–1122, 2018, doi: 10.4314/jfas.v10i1s.81.
[35] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, 2003, doi: 10.1076/csed.13.2.137.14200.
[36] L. E. Winslow, "Programming pedagogy—a psychological overview," *ACM SIGCSE Bull.*, vol. 28, no. 3, pp. 17–22, 1996, doi: 10.1145/234867.234872.
[37] M. McCracken *et al.*, "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," *ACM SIGCSE Bull.*, vol. 33, no. 4, pp. 125–180, 2001, doi: 10.1145/572139.572181.
[38] J. Morphew, K. J. Kaufman-Ortiz, S. N. Rebello, and C. M. Rebello, "Case study on engineering design intervention in physics laboratories," in *Proceedings of the 2022 ASEE*

Annual Conference Excellence throuugh Diversity, Minneapolis, MN, Jun. 2022. [Online]. Available: https://strategy.asee.org/42071
[39] J. Morphew, R. Lopez, A. Bralin, R. Subramaniam, S. Rebello, and C. Rebello, "Integrated STEM: Impact of engineering design and computer science in STEM labs," in *Proceedings of the Eighth Annual STEM Education Conference*, West Lafayette, IN, Jan. 2023. doi: 10.5703/1288284317601.
[40] J. W. Morphew, C. M. Rebello, A. Bralin, and N. S. Rebello, "Infusing design: impact of engineering design in physics labs on student engagement and perception," presented at the Poster presentation at the 2022 IUSE Summit, Washington, D.C., Jun. 2022.
[41] J. W. Morphew, A. Bralin, T. Chapman, C. M. Rebello, and N. S. Rebello, "Student perception of engineering design activities in introductory physics labs," presented at the Paper presentation at the summer meeting of the meeting of the American Association of Physics Teachers, Vancouver, Canada, 2021.
[42] N. G. Holmes and C. E. Wieman, "Introductory physics labs: We can do better," *Phys. Today*, vol. 71, no. 1, pp. 38–45, 2018, doi: 10.1063/PT.3.3816.
[43] V. K. Otero and D. E. Meltzer, "The past and future of physics education reform," *Phys. Today*, vol. 70, no. 5, pp. 50–56, 2017, doi: 10.1063/PT.3.3555.
[44] K. Caelli, L. Ray, and J. Mill, "'Clear as mud': Toward greater clarity in generic qualitative research," *Int. J. Qual. Methods*, vol. 2, no. 2, pp. 1–13, 2003, doi: 10.1177/160940690300200201.
[45] R. W. Chabay and B. A. Sherwood, *Matter and interactions*. John Wiley and Sons, 2015.
[46] B. M. Capobianco, C. Nyquist, and N. Tyrie, "Shedding light on engineering design," *Sci. Child.*, vol. 50, no. 5, p. 58, 2013.