Bridging the Gap: Infusing Natural Science Classes with Computer Science Concepts and Skills

Elena Yanakieva[1](✉), Annette Bieniusa[1], Thomas Becka[3],
Brian B. Moser[1,2], Dominik Jerger[1], and Christoph Thyssen[1]

[1] University of Kaiserslautern-Landau, Kaiserslautern, Germany
{elena.yanakieva,annette.bieniusa,christoph.thyssen}@rptu.de
[2] German Research Center for Artificial Intelligence (DFKI),
Kaiserslautern, Germany
brian.moser@dfki.de
[3] Eduard-Spranger-Gymnasium, Landau, Germany

Abstract. As we venture further into an ever-evolving digital society, it has become imperative for schools to adapt and equip future generations with the required skills. Computational Thinking (CT), specifically its algorithmic thinking aspect, plays a significant role in computer science and other sciences, such as biology, chemistry, and physics. It encourages students to reason at multiple levels of abstraction. This ability is particularly useful in natural science, as scientific experiments require critical skills like conceptualization, problem decomposition, and solving or designing structured systems. However, due to organizational and curricular restrictions, these concepts and skills are typically taught separately, leaving it to the learners to connect and apply the acquired skills in the respective context.

To bridge this gap, we designed a two-day workshop that embeds CT in a physics and sustainability context, namely energy-efficient housing. In this workshop, we employ the Calliope Mini, a micro-controller explicitly designed for educational purposes, to teach essential algorithmic thinking and data processing. We carried out the workshop twice in a primary school in Germany, each with 20 children in 4th grade (ages 9–10). In this experience report, we present the multi-disciplinary workshop idea and discuss the outcome and observations from its execution. Overall, our study demonstrates the potential of such a workshop design as a practical tool for teaching CT concepts to children. Finally, we critically examine the challenges of this approach and highlight the importance of proper technical and educational prerequisites for a successful implementation.

Keywords: Computational Thinking · STEM · algorithmic thinking · multi-disciplinary teaching

J.-P. Pellet and G. Parriaux (Eds.): ISSEP 2023, LNCS 14296, pp. 180–193, 2023.
https://doi.org/10.1007/978-3-031-44900-0_14

1 Introduction

The demand for Computational Thinking (CT) is expected to grow significantly in the next decade, particularly as more industries and administrations adopt digital technologies [6,18]. CT is foundational for Computer Science (CS), and a critical skill required in numerous fields, including engineering, science, finance, and healthcare [3]. It is a method for problem-solving that involves breaking down complex problems into smaller, more manageable parts and using algorithms and logic to analyze and solve them [17].

In Germany, there is still a significant education gap regarding CT [11]. However, Germany's educational system has not yet enforced mandatory CS classes in schools on a national level [8,12]. As of 2023, only 5 out of 16 states have added CS to their curriculum [13].

A major challenge is the shortage of CS teachers in Germany [13]. Many schools struggle to find qualified teachers to deliver effective and engaging CS classes [7]. This shortage of teachers can be a significant barrier to implementing mandatory CS classes across the board.

Meanwhile, as part of project GeNIUS[1], we propose a multi-disciplinary approach to infuse CS topics into natural science subjects like biology, chemistry, and physics, as natural sciences often require sensory measurements and algorithmic thinking in practice, both essentially covered by CT. This multi-disciplinary approach will allow children to develop CT from an early age and provide a better understanding of how different fields interplay to solve one particular problem. The project aims to conceptualize school lessons that make use of computer science practices, especially programming, to experiment on a natural science topic. The focus lies on summarizing success conditions in a school setting rather than evaluating the effectiveness of these lessons.

The first lesson that we are working on is on the topic of "energy-efficient insulation materials in houses". The lesson employs programming a micro-controller to conduct an experiment that measures how fast the temperature inside an insulated "house" levels out with the room temperature after initial heating. Before the start of evaluating the lesson series in a school setting, we were invited to hold a workshop at a local primary school with students in the 4th grade (ages 9–10). This gave us the opportunity to evaluate our concept and test our hypothesis that it is possible to successfully teach CT skills in a natural science context.

In this report, we summarize our experience on integrating CS fundamentals into natural science lessons. We further discuss the challenges involved and propose solutions to overcome them.

2 Background

This section introduces Computational Thinking and examines its implications in the educational landscape. Also, it introduces the Calliope Mini, an educational micro-controller designed to simplify coding and technology for children.

---

[1] https://www.genius-schule.de/.

2.1 Computational Thinking (CT)

In the context of this work, “thinking like a computer scientist ... requires thinking at multiple levels of abstraction” [17], which is especially useful for natural science education to review a topic from different angles. Wing further argues that “Computational Thinking (CT) is a fundamental skill for everyone, not just for computer scientists” [17].

In other definitions, CT is often divided into different aspects, e.g., pattern recognition, abstraction, decomposition, and algorithms [4]. The CT part of our workshop consists of developing a suitable algorithm for achieving the goal of the experiment. Although in essence all four aspects are employed for this task, our focus naturally lies on algorithmic thinking.

2.2 Calliope Mini

The Calliope Mini is a micro-controller, similar to MicroBit, explicitly designed for educational purposes to introduce children to coding and digital technology [1]. With a variety of programmable features, the Calliope Mini is a valuable companion in modern, digitally-enhanced classes. It features a range of components, including a 5×5 red LED matrix, a temperature sensor, and a light sensor, among others. It can be programmed with block-based programming languages using free editors, such as Microsoft MakeCode [2]. For teaching purposes, we chose block-based programming because it is visual and novice-friendly [16]. Programs can be transferred to the device via Bluetooth. In contrast to other mini-coding devices, such as Arduino, employing different sensors does not require complex wiring. Furthermore, unlike Raspberry Pi, it cannot compile locally and expects already pre-compiled programs. For the purpose of our project, we opted out for using Calliope Mini, as its use has drastically increased in schools all over Germany. Our hope is that these concepts will become part of the curriculum in the future. Therefore, we focus on providing lesson ideas for the tools that are already present. However, we firmly believe that the concepts of the lessons can be executed with any other micro-controller.

3 Workshop Design

To bridge the gap between natural sciences and CS, we designed an experiment on efficient housing insulation using Calliope Mini. It teaches algorithmic skills alongside physics knowledge about heat dissipation and insulation materials. This workshop has been developed as part of a research project called GeNIUS, which aims to develop interdisciplinary cross-curricular STEM lessons. Drawing inspiration from the maker culture, it is intended to foster learning-by-doing, collaboration, hands-on skills, and to give insights into the functioning of technical devices and scientific phenomena. The workshop is divided into two days with four blocks of around 40 to 50 min each and breaks as shown in Fig. 1. In the following, we describe the insulation experiment and the plan for each day in detail.

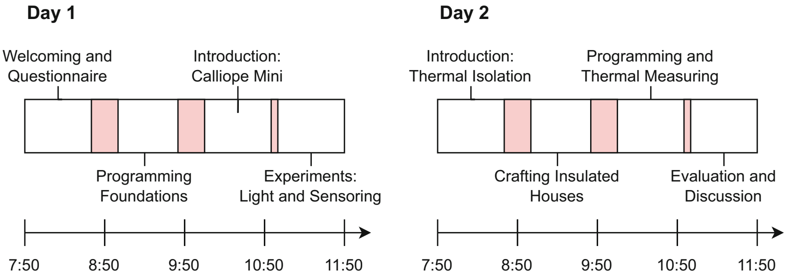

Fig. 1. Schedule for the two workshop days, including breaks (red). (Color figure online)

3.1 Main Experiment: Insulation in Houses

The selected experimental goal 'energy-efficient housing' for this workshop was chosen for its direct relevance and importance in the context of today's environmental challenges. Through the lens of insulated housing analysis, students were introduced to the effects of different insulation materials on a home's thermal stability, visualizing these impacts through data captured in real-time.

Similar natural science experiments are conducted throughout the school year by teachers in natural science classes. However, so far these have been done analogously - using analogous thermometers, monitoring and documenting the temperature by hand. We believe that this is exactly where CS can be integrated and allow the students to explore the possibilities of digital technologies.

The experiment is structured into four phases: insulating the house, implementing a measuring system, performing a measurement, and discussing the results. In the first phase, the students pick the material to insulate their houses. They then use the chosen material to insulate a cardboard box, which we use to simulate a house.

In the second phase, the students implement a program for Calliope Mini to continuously measure the current temperature and send it via Bluetooth to a mobile device. During the third phase, an external temperature sensor is connected to the micro-controller (see Fig. 2a). The sensor is put inside the insulated house, and the Calliope is turned on. Two heat pads are placed inside the house. The Calliope micro-controller is then connected via Bluetooth to an app called Phyphox, which receives the transmitted temperature values and automatically plots them in a diagram [14] (see Fig. 2b). The third phase takes around 30 min in total, where the students observe the temperature curve and compare it with the other insulated houses in the class.

3.2 Planning: Day One

We dedicate the first day exclusively to programming basics and introducing the students to the Calliope Mini and programming. Also, we set aside the first

(a) Assembling and Programming Calliope Mini

(b) Preparing House and Sensor Integration

(c) Experiment Start with Activated Heat Pads

Fig. 2. Visual documentation of the thermal experiments in this workshop.

block for the pre-knowledge test. So the program for the first day is split into three blocks.

In the first block, we distribute hand-outs to the children with exercises to be solved individually using pen and paper. These provide the basic concepts used in (block-based) programming, including the definition of algorithms, conditions, branching, and different types of loops. In the last exercise, the children must construct an algorithm from scratch. In the last 10 min of the block, the students form teams of two, and each team receives a Calliope Mini. The teacher then leads a discussion about the different parts of the micro-controller.

In the second block, each team receives a worksheet with simple programming tasks to guide them through the different blocks. In the end, they are free to experiment on their own.

In the third block, the students learn to use embedded and external sensors and read their values. They learn how to send these values via Bluetooth and plot them on their tablets using the Phyphox app [14]. For that, they are given the task of writing a program that initializes the Bluetooth connection and sends light and temperature values via Bluetooth. They can experiment with the measurements, e.g., by covering the micro-controller with their hand or holding the temperature sensor in their hands, and see how the graph is affected.

3.3 Planning: Day Two

The second day was designed to further enrich the students' understanding of energy efficiency and insulation principles through hands-on experimentation.

In the first block, the session commences with a teacher-led discussion about energy efficiency and insulation, covering the fundamentals of heat-conductive and insulating materials. The very basics of heat-conductivity and insulation are presented by a short video discussing familiar objects from everyday life - e.g.

why do pans have a plastic handle. Energy efficiency is then presented from the housing point of view - the more energy the house loses, the more energy it needs to produce to stay warm. A thermal camera is used to visualize real-time temperature differences in the classroom, aiding in conceptual understanding. This is followed by a classroom discussion of the childrens' observations of housing insulation and what familiar materials have insulation properties.

During the second block, students are encouraged to tap into their creativity as they craft their insulated 'houses' from cardboard boxes and pre-selected insulating materials.

In the third block, students implement a program to capture temperature data using an external sensor and transmit the values to Phyphox via Bluetooth. Once the sensors are positioned and the Bluetooth connection is established, each team adds two heat pads to their 'house' and starts the measurement. During this period, the students are free to inspect the progress of other teams as the teacher guides a collaborative examination of the houses using the thermal camera. The experimental setup is left to operate over the final break.

The fourth block marks the end of the 30-min measurement period. It concludes with a teacher-led discussion of the resultant temperature graphs. Each team compares their data with others to ascertain the most effective insulation strategy, thereby reinforcing the learning objectives.

4 Results

In this section, we summarize our first-hand experiences gathered during the workshop's implementation. The assessment of a pre- and post-workshop survey is given in Appendix A.

4.1 Execution

During the execution, there were four instructors in the classroom, three of which were part of the design phase of the workshop and one a primary school teacher, who had no prior experience with CS, but whom the students were familiar with. For the execution, we used the Calliope Mini micro-controller and the external Grove temperature sensor. We provided two kinds of insulation materials: plain styrofoam boards and ones with aluminum covers on one side.

We carried out the experiment twice, with 20 students each. The two executions differed strongly due to the sudden misfortune of a failed router and, therefore, connection issues during the first two days.

Group 1 W/O Internet. The execution of Group 1's activities was heavily impacted by unexpected internet connectivity issues (see Fig. 3). The disconnection prompted the early implementation of all offline tasks, necessitating a reshuffling of the Day One and Day Two schedules. This disruption significantly

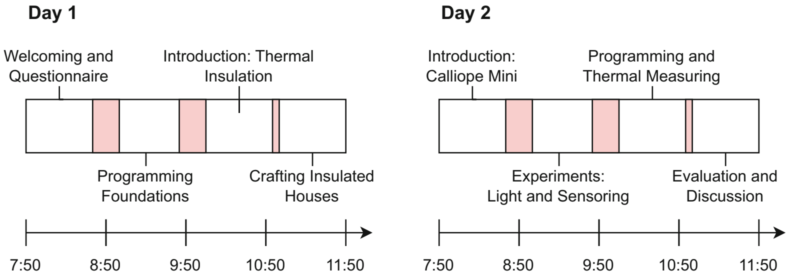

Fig. 3. The actual execution of Group 1, including breaks (red). (Color figure online)

dampened students' enthusiasm as they were unable to engage with the micro-controller as expected, leading to initial disinterest. Nonetheless, anticipation for the digital experiment built up by the day's end.

On the second day, the absence of an internet connection persisted, and we could not establish an offline server for Bluetooth implementation with the Calliope Mini. Our iPads proved futile in this offline environment, with mobile Internet hotspots providing only limited support. Resorting to our last available option, we manually compiled students' programs on a laptop and transmitted them via USB to the Calliope Mini. This workaround, although functional, resulted in queues and suppressed autonomous experimentation, fostering frustration.

While the experiment proceeded smoothly, students displayed waning concentration and interest, preferring play over participation in the post-experiment discussion. The feedback for this session indicated that it was "very boring," signifying the tiredness due to the unforeseen network outage.

Group 2 W/Internet. In contrast, the workshop's second iteration proceeded as initially planned (see Fig. 1). The students displayed high levels of engagement and enjoyment as they interacted with the micro-controllers and developed their programs.

During the outdoor experiment, with temperatures around 10°, students were allowed to play outside. Nonetheless, they demonstrated keen interest, periodically checking their 'houses' and comparing their data with other groups. The thermal camera inspections of their houses added an extra layer of intrigue.

Post-experiment discussions witnessed active participation, with students enthusiastically presenting their findings. The successful execution of this workshop and the hands-on approach resulted in highly positive feedback, with the group endorsing it as "one of the best workshops they have attended".

5 Discussion

The workshop's successful execution demonstrated the feasible interplay of natural sciences and CT, an important step forward given the integral role these systems play in our everyday lives. CT skills have become essential in the current era, cutting across numerous professional fields.

Our vision embraces a future where both educators and learners across all disciplines benefit from digital technologies and CT skills. Common perceptions, held by both young and old, frequently cast computers and software as enigmatic 'black boxes'-facilitating 'magic' in our daily applications. Our workshop's design aimed to demystify these notions for both students and teachers.

The participants were not only introduced to the formulation of simple, machine-understandable commands but also developed a clearer understanding of how devices interpret these commands. By leveraging digital technologies to convey insights derived from natural sciences-in this case, insulation and energy loss-we believe that the workshop fostered the development of multiple essential skills relevant to everyday life.

Creativity. During our workshop, we witnessed many out-of-the-box solutions. Not only that, but the children were eager to test out their programs in many scenarios, often proposing new ideas about where to measure temperature and what they would like to see next.

Digital Literacy and Electro-Mechanical Skills. The Calliope Mini has multiple ports. The children had to examine and conclude how to approach the wiring, i.e., which port does Calliope get power from, which port is used for data transmissions, and which for sensor measurements. Furthermore, they had to learn the Calliope Mini programming environment, which was a completely new application for them. Yet, they were quick to learn how to browse through the different block regions and find the blocks they needed.

Collaboration Skills. Throughout the workshop, the children worked on the tasks in teams of two. We observed great teamwork, where both kids would work together to achieve a common goal. For example, connecting the Calliope Mini with the tablet via Bluetooth is done by simultaneously pressing 3 buttons. Each team figured out that both groupmates were needed for this task. For the programming tasks, we observed different types of collaborations among the students. The first type would first discuss the solution and then implement it and test it together. The second type would involve one child implementing the program, another checking the provided material, and testing the program. They would then debug it together.

Graph Reading. Being able to deduce information from graphs is an important skill, which is learned at the end of 4th grade or the beginning of 5th grade in German schools [15]. Thus, the children for which the workshop was designed did not know how to create or read graphs. We had prepared material to slowly introduce them to this topic, introducing only the most necessary, like reading the maximum or the minimum, and interpreting steep and gradual lines. However, during the execution, we saw no need for this material as the kids were fascinated that they could see something moving and being created in real-time.

They were able to compare their graphs with the graphs of their classmates intuitively and to interpret what the highest temperature was and how fast it has been reached.

Further Aspects. Besides the skills mentioned so far, we also observed boosts in confidence and interest. Many children had expressed uncertainty before the workshop. After the workshop, all of them became more confident and eager to try out more.

Integrating such a high number of skills for achieving a task requires concentration and commitment from everyone involved. While this is a general challenge for children in this age group, we observed that in the second group even children who were known for disrupting lessons participated enthusiastically with good results.

5.1 Supporting and Hindering Influences on the Learning Process

In the following, we summarize the lessons learned from our workshops.

Didactical Aspects. Integrating active and collaborative teaching concepts makes the learning experience more engaging and interactive. In our experience, leaving time for semi-structured or self-guided exploration increased the students' confidence and interest. The ability to make mistakes without negative consequences encouraged the students to be curious and open-minded to new challenges and tasks of higher complexity. In our opinion, the school curriculum often fails to develop the skill of learning from one's own mistakes. We believe that CS allows for safe exploration and hands-on experimentation as programs can be arbitrarily often changed and adapted, and IDEs like the MakeCode editor prevent certain errors.

However, despite the many benefits of teaching with the Calliope Mini, it is essential to acknowledge that some teachers may face challenges when adopting new technologies and teaching methods. Therefore, teachers must have an open mind and be willing to learn and experiment with the device themselves [5,10]. Thus, delegating the implementation of this concept to teachers initially unfamiliar with Calliope Mini might result in sub-optimal teaching results. Only with proper support and resources, teachers can successfully integrate the Calliope Mini into their teaching practices and provide their students with a valuable and engaging learning experience.

Organizational Aspects. When teaching with the Calliope Mini, a frontal instruction approach is infeasible due to the technical nature of the device and the potential for individual issues to arise. As our and other collegial experiences show, having more than one instructor in the classroom is recommended to assist with troubleshooting and individual support.

While experiments in natural sciences already involve careful planning and preparation of material, this effort is increased with such a cross-curricular and interdisciplinary lesson. However, with more frequent employment of the Calliope Mini platform, the initial overhead of acquainting the students with the programming environment, sensors, etc., is reduced and the system can be used productively after a very short amount of time.

Technical Setup. As mentioned in Sect. 4.1, the workshop was conducted twice. In the first iteration, the school's router broke on day one which rendered our technical setup unusable as the programming app on IOS requires server access for compiling the block-based programs to executables for the Calliope Mini. Therefore, the students could not work independently on the tasks which resulted in them not gaining a deeper understanding of the topic and unfortunately, losing interest. For us, however, it was a valuable experience from which we could deduce crucial technical conditions that need to be met in order for the lessons to succeed.

To effectively use the Calliope Mini, it is crucial to have a stable WiFi connection or a device that can compile programs. While it is possible to set up a local compilation server[2], it turned out that the programming app cannot be redirected to a different server. And though the IDE can also be used in the browser, it is not possible to upload the programs using Bluetooth as via the app. Instead, a USB wiring is required which requires in turn additional hardware and adapted worksheets with different instructions. Given the complexity of the workarounds, we recommend having an alternative ready in case of unforeseen network problems. The need for conservative approaches in a school setting is crucial for the successful integration of CS concepts in natural science topics.

5.2 Generalization of the Results

Stemming from our experience, we were able to derive the above-mentioned limitations and dependencies. Despite these, we believe that this workshop can be modified into a lesson series as part of the school curriculum for natural sciences or physics. In fact, this has been an ongoing work in GeNIUS project over the last months. The duration of the series depends on the pre-knowledge of CS concepts, especially block-based programming, and micro-controllers of the students. Furthermore, the age of the students is also a factor to be considered. We reckon that the older the students are, the less supervision and time they would need. Therefore, we believe that it is feasible for an experienced teacher to lead this experiment with up to 30 students of an appropriate age.

6 Conclusion and Outlook

In the rapidly digitalizing world, it is of utmost importance that children are introduced to Computational Thinking (CT) from an early age. Our work explores incorporating CT into a natural sciences lesson using the Calliope Mini.

---

[2] https://github.com/calliope-mini/pxt-calliope-static.

Through an experiential workshop, we observed the potential benefits of this approach. From fostering creativity and collaboration to improving digital literacy, and graph reading skills, the combination of CT with natural sciences opened up new avenues of learning. We believe our workshop plays a part in demystifying the notion of technology being an enigmatic black box, providing the first steps towards enabling children to comprehend and participate in the digital world around them.

It is crucial to acknowledge that the successful execution of the workshop hinges on the presence of technical and educational prerequisites. Stable internet access, availability of sufficient technical support, and teachers' openness to engaging with new technologies are all vital factors. Through this experience, we have found that even in failure, there are lessons to be learned and opportunities for refining our promising methodology.

We plan on exploring further possibilities to limit the dependencies on connectivity. As part of our project GeNIUS, we are working on modifying the workshop to be a lessons series, able to be conducted in schools solely by teachers. We plan on evaluating the success rate and further dependencies in German schools.

Our outlook is to conceptualize further lessons on other topics, taking into account also biology and chemistry. The main goal of the project is to evaluate the conditions that need to be met in order to successfully integrate CS concepts into natural science lessons. We furthermore plan on evaluating the learning effects of CT and of the respective natural science fields for each lesson series. This will be done on a wider test group, including both students and teachers. Furthermore, we are preparing a training specifically for natural science teachers, in order to show how CS concepts are an essential part of the natural sciences and, also, to teach them how to use micro-controllers. We hope that one day these lessons can be integrated as part of the natural science curriculum in schools.

We also hope that our insights will spur further research, propelling our education systems into a future where digital technologies and algorithmic thinking skills are seamlessly integrated.

Acknowledgements. This work is sponsored by the BMBF project GeNIUS under funding code 16MF1011B.

Appendix A: Student Test

We conducted two tests with each student group, one administered in the morning before their respective workshop and a second one a week later. Their purpose was to assess the workshop's impact on the participants' knowledge and their preference to formulate algorithms in natural vs. semi-formal vs. graphical programming language.

Questionnaire. The questionnaire used in the tests was designed to cover two fundamentally different areas: First, understanding of the concept of an algorithm, and second, the way in which algorithmic processes are described. Accordingly, the pre-test asked dichotomously whether the concept of an algorithm was known, followed by a multiple-choice task with four options asking which of the given everyday examples corresponds to an algorithm. In the second part of the questionnaire, the participants were asked to describe paths through simple labyrinths. The questionnaires pre- and post-test were structured identically. In the multiple-choice task, the answers offered were varied at the same level of difficulty. Similarly, slightly modified mazes were used for the open-ended descriptions of the paths through the labyrinths.

Evaluation. In the first question, the first item asked the students the dichotomous question "Do you know what an algorithm is?". Initially, a third of the participants (10 out of 30[3]) claimed familiarity with the term 'algorithm'. However, upon probing their comprehension with a follow-up multiple-choice item, only a single student could accurately identify the notion that best describes an algorithm in their everyday life. Post-workshop results showed a notable improvement, with 20% of students (6 out of 30) choosing the example matching the concept of an algorithm. Intriguingly, these correct responses were concentrated within Group 2[4] (representing 42% of its participants or 6 out of 14). Group 1, by contrast, produced no correct answers in the post-workshop survey. This stark difference in outcomes between the two groups underscores the significant role of technological stability in learning environments. We assume that the issues with internet connectivity for Group 1 of the workshop were likely the reason since coding and exploration of algorithmic structures were not possible. Thus, the results emphasize the impact of reliable technology in schools on the learning outcome.

In the second part of the questionnaire, the students were asked to describe a path through a labyrinth. In a sequence of items, they were told to describe the path

1. as they choose, without suggestions;
2. using abbreviations for up, down, left, and right;
3. using counting abbreviations (e.g., "repeat 2 times up").

For the final item, the labyrinth required a repeated sequence of steps ("repeat 2 times: up and left"). Here, the students were not given any suggestions regarding the description style. To evaluate these open-ended questions, we used the qualitative content analysis established by Mayring [9] using the following research questions:

How do students preferably describe the steps of an algorithm?
Are they able to adopt a semi-formal notation, and do they stick to it?

---

[3] We report here only on the participants who submitted both pre- and post-workshop questionnaires and who agreed together with their parents on participating in the survey.

[4] See Sect. 4.1 for a description of the two groups.

We defined the following categories:

Natural Language. Students use complete sentences in German;
Directions. Students use a sequence of instructions (e.g., "right, right, up");
Directions with Counting. Students use a sequence of instructions summarizing the same instructions(e.g., "2 × right, 1 × up");
Usage of Repeat. Students use the suggested repeat abstraction (e.g., "repeat 2 times: right, then up"), akin to the block-programming language introduced in the workshop.

In the first item of the question when no suggestion was made, 24 out of 30 students in the pre-test wrote their answers in natural language (post: 10). When asked to use abbreviations for the directions, 9 out of 30 students in the pre-test continued using full sentences mixed with the abbreviations (post: 6), 6 gave sequences of directions while 15 summarized by counting the same instructions (post: 8 vs. 14). Only 6 out of 30 students used the suggest repeat-abstraction in the pre-test (post: 12). For the final item, 3 out of 30 students used the repeated sequence (post: 6), all others used single instructions. Here, most students chose directions over natural language (pre: 18 vs. 5, post: 19 vs. 3). Notably, in Group 1 four students did not answer the question in the post-test while only two students in Group 2 dropped the question. As a trend, students were switching from the (more verbose) natural language abstractions to the shorter and more precise semi-formal descriptions when introduced to them. Their usage in programming tasks facilitates the adoption. Interestingly, in three cases, the usage of the semi-formal instructions improved the students' solution.

Limitations. Given the sample, its size, and the conditions of execution during the two iterations of the workshop, the conclusions and inferences drawn from it are hypothetical in nature and must be considered clearly preliminary. As the sample size increases, a new analysis will need to be conducted to see if any presumed efficacy and trends in response behavior (i.e., learning level) can be confirmed and generalized.

References

1. Abend, M., Gramowski, K., Pelz, L., Poloczek, B.: Coden mit dem Calliope mini. Programmieren in der Grundschule, Lehrermaterial (2017)
2. Ball, T., Chatra, A., de Halleux, P., Hodges, S., Moskal, M., Russell, J.: Microsoft makecode: embedded programming for education, in blocks and typescript. In: Proceedings of the 2019 ACM SIGPLAN Symposium on SPLASH-E, pp. 7–12 (2019)
3. Bocconi, S., et al.: Reviewing computational thinking in compulsory education: state of play and practices from computing education (2022)
4. Dong, Y., et al.: Prada: a practical model for integrating computational thinking in k-12 education. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp. 906–912 (2019)

5. Hunt, N.P., Bohlin, R.M.: Teacher education students' attitudes toward using computers. J. Res. Comput. Educ. **25**(4), 487–497 (1993)
6. Kafai, Y.B., Burke, Q.: Connected Code: Why Children Need to Learn Programming. MIT Press, Cambridge (2014)
7. Klemm, K.: Lehrerinnen und lehrer der mint-fächer: Zur bedarfs-und angebotsentwicklung in den allgemein bildenden schulen der sekundarstufen i und ii am beispiel nordrhein-westfalens. Gutachten im Auftrag der Deutsche Telekom Stiftung, pp. 1–13 (2015)
8. Knobelsdorf, M., et al.: Computer science education in north-Rhine Westphalia, Germany-a case study. ACM Trans. Comput. Educ. (TOCE) **15**(2), 1–22 (2015)
9. Mayring, P.: Qualitative Content Analysis: Theoretical Foundation, Basic Procedures and Software Solution. Klagenfurt (2014)
10. Mozelius, P., Ulfenborg, M., Persson, N.: Teacher attitudes towards the integration of programming in middle school mathematics. In: INTED2019 Proceedings, pp. 701–706. IATED (2019)
11. OECD: OECD Employment Outlook 2019 (2019). https://doi.org/10.1787/9ee00155-en, www.oecd-ilibrary.org/content/publication/9ee00155-en
12. Sabitzer, B., Antonitsch, P.K., Pasterk, S.: Informatics concepts for primary education: preparing children for computational thinking. In: Proceedings of the 9th Workshop in Primary and Secondary Computing Education, pp. 108–111 (2014)
13. Schröuder, E., Suessenbach, F., Winde, M.: Informatikunterricht: Lückenhaft und unterbesetzt (2022). www.stifterverband.org/medien/informatikunterricht
14. Staacks, S., Hütz, S., Heinke, H., Stampfer, C.: Advanced tools for smartphone-based experiments: phyphox. Phys. Educ. **53**(4), 045009 (2018)
15. Ständige Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland: Bildungsstandards fädas Fach Mathematik Primarbereich (2022). https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2022/2022_06_23-Bista-Primarbereich-Mathe.pdf
16. Weintrop, D.: Block-based programming in computer science education. Commun. ACM **62**(8), 22–25 (2019)
17. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
18. World Economic Forum, V.: The future of jobs report 2020. Retrieved from Geneva (2020)