Journal of Research, Policy & Practice of Teachers &
Teacher Education (ISSN 2232-0458 / e-ISSN 2550-1771)
Vol. 15, No. 1, June 2025, 101-107

Exploring the impact of micro:bit on pre-service teachers' professional development

Asif Mushtaq[1*], Kalus Peter Eichler[1,2] and Moritz Seibold[2]
[1]NORD University, Department of Mathematics, Faculty of Education and Arts, Realfag, 8049 Bodø, Norway
[2]University of Education Schwäbisch Gmünd, 73525, Schwäbisch Gmünd, Germany.
**Corresponding author: asif.mushtaq@nord.no*

To cite this article (APA): Mushtaq, A., Eichler, K. P., & Seibold, M. (2025). Exploring the impact of micro:bit on pre-service teachers' professional development. *Journal of Research, Policy & Practice of Teachers and Teacher Education*, *15*(1), 101-107. https://doi.org/10.37134/jrpptte.vol15.1.9.2025

Received: 02 July 2024; **Revised:** 02 June 2024; **Accepted:** 03 June 2025; **Published:** 25 June 2025

Abstract

Many countries are integrating programming into their official curricula, recognizing its increasing importance. However, changing curricula alone is not enough. Essential are qualified and motivated teachers who can select and implement suitable learning environments. It can be assumed that the experiences gained by pre-service teachers (PSTs) during their own learning significantly influence the way they design their lessons later. Contributing to improving the quality of PSTs education, we developed workshops introducing block-based programming with micro:bit. This paper focuses on the theoretical foundation, practical implementation, and evaluation of the workshops. We investigate how PSTs benefit from our workshops. We examine the impact across four dimensions: problem-solving skills, motivation, collaboration and teamwork, and learning through discovery. We used a mixed-method approach to gather data from the PSTs: An online questionnaire combining questions with Likert-scaled answer options and open questions was used. The results indicate a strong positive trend in these four dimensions, suggesting that the micro:bit can enhance the personal and professional development of PSTs in programming.

Keywords: Block-based programming, micro:bit, pre-service teachers, professional development.

Introduction

Demands for using information processing technology in teaching and developing computational thinking are not new. Papert (1980) already emphasized: "And in teaching the computer how to think, children embark on an exploration about how they themselves think. The experience can be heady: Thinking about thinking turns the child into an epistemologist, an experience not even shared by most adults". Wing (2006) defines *computational thinking (CT)* as an approach that builds on the fundamental concepts of computer science to promote system design, problem-solving, and understanding of human behavior. Many recent studies (Bati, 2021; Grover & Pea, 2013; Rahul Simha, 2016; Wong & Cheung, 2020) have confirmed the importance of CT. Computational thinking is regarded as a cross-sectional competence because the skills subsumed under this concept are necessary for solving a wide range of problems. Such abilities are discussed, for instance, in Hsu et al. (2018):

- Sjödahl & Eckert (2023) discuss decomposition in programming environments. Decomposition is the ability to break down a complex problem or system into smaller, more manageable parts.
- Pattern recognition is the ability to recognize and use patterns when solving problems. According to Alexander et al. (1977) "Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."
- Abstraction (Grover & Pea, 2013) is the ability to focus on important information and ignore irrelevant information when solving a problem or generalizing.
- Algorithmizing is the ability to develop a sequence of steps to solve a problem or to apply a known sequence of steps when solving a problem.
- Debugging is the ability to find and correct errors in a solution path. Therefore, debugging is particularly important as it allows children to reflect on their solution processes, deal productively with mistakes, and learn from them.

Solving appropriate tasks is the only way to develop computational thinking, and programming tasks are especially suited to this. Recent research highlights programming as a valuable teaching resource, emphasizing that it fosters computational thinking and enhances mathematical learning and teaching (Sacristán et al., 2022). Because the five components of computational thinking mentioned above are essential prerequisites for successfully solving a wide range of tasks, especially mathematical tasks, computational thinking is crucial for success in mathematics lessons.

This fact is one reason programming has been integrated as an essential part of the new mathematics curriculum in Norway (Norwegian Directorate for Education and Training, 2019). In general, curricula worldwide reflect the importance of computational thinking and programming. The European Union encourages its member countries to integrate computational thinking and programming into their curricula. Other countries have integrated coding into science and mathematics education as a separate subject or as a cross-curricular competence (Kaufmann & Stenseth, 2021; Weintrop et al., 2015). Moreover, during the MEDA3 conference, a panel discussion delved into the curricular implications of computational thinking, highlighting its growing significance. The discussion featured insights from multiple countries, emphasizing the relationship between computational and mathematical thinking.

The extent to which the objectives set in the curricula are achieved depends heavily on the teacher and how they design their lessons. Regarding teaching programming, it must be considered that this area is new and that pre-service teachers (PSTs) were not confronted with it during their school careers. Therefore, PSTs must first learn programming themselves while at the same time understanding its potential for the education of students and exploring didactical opportunities for teaching programming.

Education can always be viewed from two aspects. The material aspect reflects that students acquire educational content, which they can apply as a tool in current or future situations. The formal aspect reflects that learners develop their personalities while acquiring the educational content. This statement is generally true and particularly applicable when teaching programming, where there are countless suggestions for more or less complex projects using different hardware and associated programming environments (Hsu et al., 2018). The necessary material and time expenditure is only justified if it is reflected in the educational result. Especially the introduction of specific hardware, the conduct of a project with this hardware, and the choice of a lesson design are only justified if the goal is not only to acquire programming skills and solve programming tasks but, above all, the student's personal development. As Papert (1980) discusses, programming is not only about syntax and algorithms but also about fostering problem-solving skills, creativity, and the ability to cooperate.

When addressing the pedagogical aspects of programming instruction, micro:bit (Micro:Bit Educational Foundation, n.d.) emerges as a compact yet formidable educational instrument—an indispensable resource for learners embarking on their initial programming endeavors. The device's physical computing functionalities significantly augment student engagement by allowing learners to observe the immediate and tangible consequences of their coding efforts (Hartley et al., 2024). Its inherent simplicity and adaptability empower educators to implement a diverse array of projects that encompass various degrees of complexity. The extensive range of applications pertaining to mathematics and science positions micro:bit as an exemplary ally in STEM lessons (Lu et al., 2021). Micro:bit provides the potential for experiential learning and stimulates curiosity, fostering an enthusiasm for discovery-based education. As previously articulated, the selection of specific projects to undertake with micro:bit within the educational environment cannot be determined solely by the programming challenges associated with the project. Rather, it is contingent upon a more profound objective: enhancing each student's individual growth and development.

It is essential to examine teachers' perceptions of this emerging technology and their understanding of its potential. The perspectives of PSTs, who will educate future learners on micro:bit, are particularly significant. Analyzing these perspectives on micro:bit engagement is noteworthy. Furthermore, it is imperative to explore how practical experiences with micro:bit shape attitudes towards it and enhance the recognition of its potential in facilitating discovery learning, problem-solving, teamwork, and motivation. We organize workshops where PSTs gain practical experience and reflect on it theoretically. These workshops aim to qualify the PSTs to work successfully with micro:bit themselves. In addition, we like to motivate the PSTs to continue working with micro:bit and arouse joy, interest, and curiosity. Such experiences allow PSTs to make didactic decisions and design their lessons.

Research objectives

In view of the importance of the teacher's work mentioned above, it is interesting to investigate the impact of our workshops. Therefore, one aim of this work is to answer the following two research questions:

RQ1: To what extent can a workshop contribute to PSTs recognizing the potential of micro:bit?

RQ2: How do pre-service teachers (PSTs), after a workshop, rate the effect of working with micro:bit regarding developing problem-solving skills, increasing motivation, fostering learning through discovery, and supporting collaborative work?

Methodology

We utilized a mixed-method approach to collect data from the PSTs after conducting the workshops. We gathered the participants' opinions on the aspects we aimed to explore. An online questionnaire on (Nettskjema-UiS, n.d) combined Likert scale questions with open-ended ones. Participants rated statements on a 5-point Likert scale, where 1 represented "strongly disagree" and 5 represented "strongly agree," while also having the opportunity to provide open responses. The questionnaire was divided into sections, and the participants' opinions were asked concerning the four dimensions of problem-solving, learning through discovery, motivation, teamwork, and collaboration. We received responses from 45 participants in a post-survey.

Research design

We wanted to explore the extent to which PSTs can grasp the potential of using micro:bit in just one appropriately designed workshop and are motivated to pursue further exploration. Through collaborative and discovery-based learning, the workshop aimed to sensitize the pre-service teachers and recognize the potential of working with micro:bit regarding problem-solving skills and developing social competencies. Appropriate activities and tasks in teaching programming are crucial for improving the students' competencies (Popat & Starkey, 2019). Therefore, we selected problems around micro:bit to address specific learning objectives related to programming within the mathematics curriculum. By working on these problems, participants could acquire programming skills and develop their competencies regarding the five components of computational thinking.

Respondents of the study

We organized three workshops at the University of Education (PH) Schwäbisch Gmünd, Germany. 50 PSTs with a teaching specialization in mathematics were involved, and they were highly likely to have later to teach programming in schools. All participants were previously unfamiliar with the micro:bit. Before the workshop, we asked the students briefly about their expectations and experiences regarding programming. In particular, we asked the participants to what extent they consider themselves confident in their ability to teach programming.

Planning and execution of workshops

The workshop was structured into three segments (see Table 1), guiding participants from basic to advanced levels. This study was conducted during author 1's visit to Germany. Each workshop lasted around 150 minutes. Author 1 led the workshops, assisted by Author 2. The workshop started with an introductory tutorial, followed by preliminary tasks. After completing tasks on a virtual simulator using the MakeCode editor, PSTs worked on physical devices and learned to transfer software to hardware using the *.hex* file format. After basic instructions, Author1 introduced the next phase, which focused on guided project work. Working on motivating problems such as using the micro:bit as a digital dice, students were encouraged to acquire programming skills.

Table 1

Structure of the Workshop Along with Specific Research Outcomes

Workshop / timeslots	Content and activities	Aimed skills and abilities development towards research outcomes (goals)
Segment 1: (50 minutes)	Introduction: The essence and importance of computational thinking and programming, the benefits of micro:bit, various aspects of the device's architecture and safety instructions	Critical thinking
	Exploring the makecode editor with its different features: Working with inputs, variables, loops, conditions, logical connections, and mathematics module	Exploration
		Learning by doing
Segment 2: (50 minutes)	Introduction to a virtual class with micro:bit	Exploration, teamwork
	Working on introductory project associated with string, leds and random-numbers and math-module	Learning by doing problem-solving
	Understanding of relationship between hardware and software	Learning by discovery

continued

Segment 3: (50 minutes)	Project work: *shaking* a micro:bit dice, programming a step-counter, and creating a compass.	Problem-solving, collaboration and teamwork, learning by discovery creativity and research,

By solving the selected problems, the PSTs learned to break down complex problems into manageable parts. They learned to design algorithms for various tasks, including managing variables, implementing loops, and applying logic operations. In addition, they gained practical experience by programming tasks such as creating an LED display, simulating dice rolls, calculating elementary events for a given number of dice simulations performed, developing a compass display, and creating a step counter. The selected tasks allowed the PSTs to delve deeper and deeper into them, expanding and making the problems more complex. These tasks are specifically designed to target students from grades 4 to 7. The more complex problems required the acquisition of extended knowledge and skills in programming with micro:bit. For example, micro:bit was initially programmed as a digital dice, where shaking the micro:bit initiated the random generator, and the output was shown as a dice image on the display. The PSTs initially recorded the data by shaking the device, thus imitating a traditional dice. This task was deepened by shaking the device several times and afterward displaying the number of each result on the micro:bit. Through such activities, participants gained knowledge of algorithmic thinking and acquired basic programming skills, such as defining variables, working with loops, understanding basic logic, and using built-in mathematical modules. Table 1 below shows the general outline of the workshop.

Findings and discussions

Working with the physical device was particularly engaging for the participants, allowing them to witness real transformations. The PSTs worked together and experienced the dynamics of teamwork. After the workshop, the participants had a first impression of micro:bit and an initial experience of the possible uses of micro:bit. It was, therefore, possible to use a more extensive questionnaire after the workshop. Some of the direct participant responses reflecting the motivational aspects of the survey are as follows:

- *"It was perfect! I am now significantly more interested in programming."*
- *"It was a joy to learn Micro-Bit"*
- *"Thank you for the great and highly beneficial workshop on Micro:Bit."*

These statements highlight the positive impact of the workshop in enhancing enthusiasm and engagement in programming and Micro-Bit learning.

Table 2

Aspects, Statements, and Statistics Regarding the Likert-Scaled Ratings of The Statements

Aspect	Question	M	Mdn	SD	Q1	Q3
Collaboration	Micro:bit classroom offers opportunities of collaborative working	4.11	4	0.65	4	5
	Micro:bit promotes student collaboration	3.93	4	0.72	3	4
Learning by discovery	Micro:bit provides the opportunity of learning by discovery.	4.29	4	0.63	4	5
Motivation	Micro:bit can contribute to improve students' motivation.	4.29	4	0.69	4	5
	Micro:bit will inspire students about computing and coding outside the classroom.	4.00	4	0.80	3	5
	Micro:bit made lessons and activities enjoyable for students.	4.40	4	0.65	4	5
Problem solving	Micro:bit is a good device for improving problem solving skills.	4.13	4	0.63	4	5
	Programming through micro:bit can improve students' ability in problem solving.	4.09	4	0.76	4	5

Table 2 shows a selection of these statements and presents the comprehensive descriptive statistics for each statement. It shows the mean (M), the median (Mdn), the standard deviation (SD), and the interquartile ranges (Q1 and Q3).

Regarding collaboration, the micro:bit was perceived as having a positive impact: the responses regarding its effectiveness in providing collaborative opportunities were highly approved with a mean of 4.11. Additionally, the micro:bit was recognized for promoting student collaboration with a mean of 3.93. Both measures have medians of 4, which suggests that there is a strong consensus in the potential of the micro:bit to facilitate a collaborative learning environment. Furthermore, students recognized the micro:bit's potential for facilitating discovery-based learning, with a mean score of 4.29 and a median of 4. According to the responses, it appears that the micro:bit has the potential to encourage independent exploration and enhance student engagement. The data also shows that the micro:bit could increase students'motivation: it suggests that motivation was positively influenced, with a mean of 4.29 and a median of 4. The interquartile range was narrow, ranging from 4 to 5, indicating this factor's potential impact. Moreover, the micro:bit appears capable of engaging learners outside the usual classroom environment. The data, showing both a mean and median of 4, alongside quartile values of 3 (Q1) and 5 (Q3), reveals a widespread agreement among PSTs about its potential. They agreed that the approach is effective in fostering engaging learning experiences. When assessing the role of working with micro:bit to improve problem-solving skills, the mean values of the associated items are 4.13 and 4.09, with a consistent median of 4. The distribution of responses is visualized in the violin plot with nested boxplots. It shows a concentration of responses in the 4 to 5 range, emphasizing a consensus among the PSTs regarding the potential of micro:bit to promote problem-solving skills.

Figure 1

Micro:Bit's Potential to Foster Educational Outcomes:

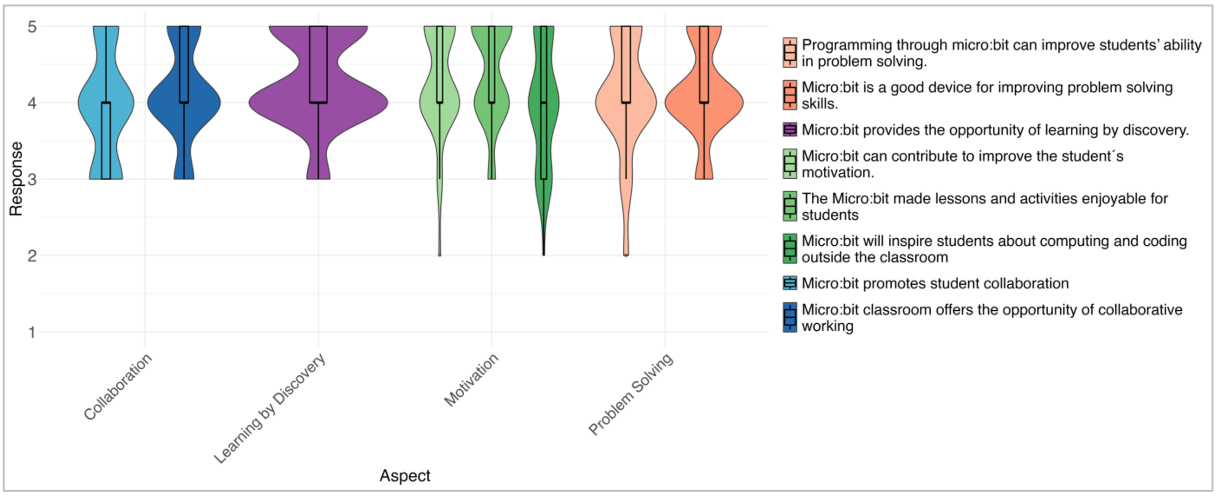

Figure 1 presents a visual representation of the data originally provided in Table 2, which includes the aspects, specific statements, and their corresponding statistical results based on Likert-scaled ratings. In this figure, the information has been reorganized and categorized according to the research questions guiding the study.

The data also shows that the PST grasped the potential of micro:bit and evaluated it positively after the relatively short workshop. In our opinion, the good results are mainly due to the tasks we carefully selected and the way we worked with them. All the tasks were chosen so that the initial problem, for example, programming a digital dice in which shaking hands leads to a number being shown, could be grasped relatively easily and solved with a short code. In doing so, the PST gained initial experience in block-based programming. After that, the problem was expanded more and more: For example, the six different events by rolling a dice were counted, and later on, the entire process of rolling the dice was even simulated. Our investigation into micro: bit's potential educational impact provides a nuanced understanding of its role in various didactic dimensions. The results of the post-workshop surveys indicate that using micro:bit effectively encourages PSTs to pursue programming beyond their initial experience and tackle additional problems.

The consistency of median values around four across different categories suggests that PSTs strongly recognize the benefits of the micro:bit. These encouraging indicators of positive perception by PSTs should be viewed within the broader context of educational objectives and the specific challenges of teaching programming.

Block-based programming and interactive engagement with appealing tasks help demystify the initial complexity of programming, making it more accessible. Therefore, it's encouraging that, following the first workshop, most PSTs reported increased self-confidence in teaching programming.

Additionally, it's important to note that, based on their experiences in the workshop, the PSTs placed high value on the potential of the micro:bit to facilitate exploratory learning, enhance problem-solving skills, and positively impact motivation to learn. The PSTs rated the influence of using micro:bit on promoting collaborative learning slightly lower than the effect it had on the other items. This rating would likely improve if PSTs experienced in future workshops how complex problems can be decomposed into subroutines and collaboratively programmed.

Overall, the significant potential of using micro:bit to transform the working methods of PSTs in ways that can overcome traditional learning paradigms is evident.

Figure 2

Self-Reported Confidence in Teaching Programming Before (Left) and After the Workshop

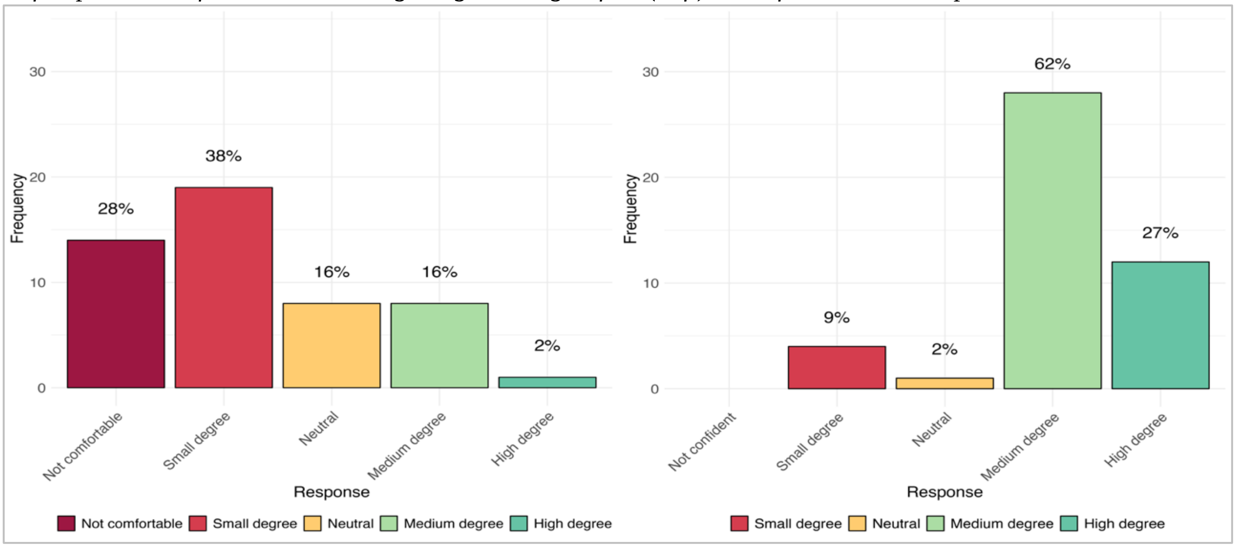

Figure 2 shows the students' self-assessment of their ability to teach programming. After a surprisingly short time, there is a considerable change in their self-assessed confidence.

Conclusions and recommendations

The engagement in programming activities with micro:bit during workshops has significantly influenced the self-reported levels of confidence in the instruction of programming (refer to Figure 2). We assert that the implementation of micro:bit is certainly a key factor in this transformation; however, a more substantial contributing factor is that the preservice teachers (PSTs) adeptly address didactically well-structured challenges. Consequently, the PSTs not only acquired programming competencies but also obtained pedagogical insights, identified educational opportunities through problem-solving, and experienced the effects of collaborative effort. Despite the limited duration of our workshop, we identified favorable outcomes across all four dimensions analyzed, highlighting the profound impact of thoughtfully designed didactic methodologies. The findings presented herein are derived from a workshop that is currently accessible in its preliminary final iteration. This workshop has undergone continuous refinement over a span of two years, and its framework can be utilized to replicate the observed outcomes. This research is recommended for educators who seek to incorporate programming into teacher education. It illustrates how the utilization of micro:bit, in conjunction with meticulously crafted didactic challenges, markedly enhances pre-service teachers' confidence, pedagogical insight, and collaborative skills. The reproducibility of the workshop concept adds further value.

Conflict of interest

The authors hereby confirmed that there is no conflict of interest concerning the publication of this article.

Author contribution

All authors contributed well to the development of this study. Asif Mushtaq was responsible for the conceptualization, research design, conducting the workshops, and contributing to the writing process. Klaus-Peter Eichler contributed to data collection and the writing of the manuscript. Moritz Seibolt was involved in data analysis and the interpretation of the results.

Data availability statement

The data supporting this study's findings can be requested from the corresponding author. It is not publicly available due to ethical restrictions.

Acknowledgement

The authors gratefully acknowledge the partial support provided by the Science and Technology in Teaching and Learning Research Group.

References

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). A pattern language: Towns, buildings, construction. Oxford University Press. https://arl.human.cornell.edu/linked%20docs/Alexander_A_Pattern_Language.pdf

Bati, K. (2021). A systematic literature review regarding computational thinking and programming in early childhood education. Education and Information Technologies, 27(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43. https://doi.org/10.3102/0013189X12463051

Hartley, K., Rubegni, E., Underwood, L., Finney, J., Ball, T., Hodges, S., De Halleux, P., Devine, J., Anderson, E., & Moskal, M. (2024). Meet MicroCode: A live and portable programming tool for the BBC micro:bit. IDC '24: Proceedings of the 23rd Annual ACM Interaction Design and Children Conference, 24(1), 355-370. https://doi.org/10.1145/3628516.3656995

Hsu, T. -C., Chang, S. -C., & Hung, Y. -T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education, 126, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. International Journal of Mathematical Education in Science and Technology, 52(7), 1029-1048. https://doi.org/10.1080/0020739X.2020.1736349

Lu, S. -Y., Lo, C. -C., & Syu, J. -Y. (2021). Project-based learning oriented STEAM: The case of micro–bit paper-cutting lamp. International Journal of Technology and Design Education, 32(5), 2553–2575. https://doi.org/10.1007/s10798-021-09714-1

Micro:bit Educational Foundation. (n.d.). Micro:Bit. https://www.microbit.org/

Nettskjema-UiO. (n.d). *Nettskjema*. https://nettskjema.no/

Norwegian Directorate for Education and Training. (2019). Curriculum for Mathematics Year 1–10 (MAT01-05) (MAT01-05). https://data.udir.no/kl06/v201906/laereplaner-lk20/MAT01-05.pdf?lang=eng

Papert, S. A. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc. https://dl.acm.org/doi/pdf/10.5555/1095592

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. Computers & Education, 128, 365-376. https://doi.org/10.1016/j.compedu.2018.10.005

Rahul Simha. (2016). Balancing symbolic and computational thinking: Reinforcing engineering math via programming. *In Spring 2016 Mid-Atlantic ASEE Conference.* American Society for Engineering Education (ASEE).

Sacristán, A. I., Santacruz-R., M., Buteau, C., Mgombelo, J., & Muller, E. (2022). Future teachers' appropriation of computer programming as a mathematical instrument and a resource for teaching. *In Proceedings of the 13th ERME Topic Conference (ETC13) on Mathematics Education in the Digital Age (MEDA3)* (256-263). Centre for Teaching and Learning Mathematics at the University of British Columbia (CTmathU). https://ctuniversitymath.ca/wp-content/uploads/2023/06/future-teachers-appropriation-of-computer-programming-as-a-mathematical-instrument-and-a-resource-for-teaching.pdf

Sjödahl, A., & Eckert, A. (2023). Abstracting and decomposing in a visual programming environment. International Journal of Child-Computer Interaction, 36, 100573. https://doi.org/10.1016/j.ijcci.2023.100573

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining computational thinking for Mathematics and Science classrooms. Journal of Science Education and Technology, 25, 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. Interactive Learning Environments, 28(4), 438-450. https://doi.org/10.1080/10494820.2018.1534245