Brodie, I., Khan, S., & Youmans, S. (2023). Coding, Computational Modelling & Equity in Mathematics Education: Working Group A Early Years/Elementary Report. In Online Proceedings of the *Coding, Computational Modelling, and Equity in Mathematics Education Symposium*, St. Catharines (Canada), April 2023.

Coding, Computational Modelling & Equity in Mathematics Education Working Group A Early Years/Elementary Report

Working Group Leaders
Iain Brodie[1], Steven Khan[2], & Sandy Youmans[3]
[1]Ontario Tech University; [2]Brock University; [3]Queen's University

Working Group Members
Priscila Correa
Brent Davis
Jeanne Dobgenski
Zeynep Gecu-Parmaksiz
Anjali Khirwadkar
Brian R. Lawler
Irina Lyublinskaya
Kelly McKie
Alexandria Middlemiss
Siri Krogh Nordby
Martine Rekstad
Marisol Santacruz-Rodriguez
Pinar Sen
Beyza Hatice Sezer
Yimei Zhang

Figure 1. Working Group participants and co-leaders on last day.

Abstract

Coding and computational Modelling are increasingly becoming curricular requirements across all age-ranges (Floyd, 2022). This working group (WG) explored this phenomenon by focusing on elementary and early years education with an emphasis on mathematics while acknowledging and respecting the interdisciplinary nature of the work of teachers during this dynamic and diverse period of child development. The working group's focus included:

- discussion of affordances and limitations of different metaphorical frames and descriptions for coding used in EEYME such as 'literacy' (diSessa, 2017), recipe, puzzle, playground (Bers, 2020), creative expression, and architecture (low floor, high ceilings, wide walls).
- developing and evaluating instructional materials and approaches for teaching coding and computational thinking in elementary and early years math classrooms (earlymath.ca)
- examining the impact of coding and computational thinking on students' math skills and attitudes
- exploring the use of technology and software tools to support coding and computational thinking in EEYME.
- investigating the ways in which coding and computational thinking can be used to enhance problem-solving and critical thinking skills in EEYME.

- sharing research findings, learning trajectories, and emerging best practices for teaching coding and computational thinking in EEYME.
- critical discussion about under-articulated or less explicit goals such as computational participation (Kafai & Burke, 2017), productive computational disposition (Pérez, 2018) as well as opportunities for considerations of ethics, equity, and the development of empathy (Bers, 2022).

The WG was dedicated to examining how and why opportunities to code across the early years through elementary school provokes mathematics learning with a combination of hands-on activities and critical discussions. Participants had the opportunity to try out some unplugged and plugged tasks that have been tested in the mathematics classroom and explored the affordances of the Micro:bit (microbit.org) as a potentially equitable tangible coding platform.

Introduction

Coding holds great promise for elementary and early years mathematics education because it fosters computational thinking, problem-solving, mathematical modelling, and creativity. By incorporating coding into mathematics education, we can deepen mathematical understanding and broaden instruction with a more playful and accessible approach (Gadanidis, 2015; Papert, 1990). However, most teachers at this level have not had formative and foundational experiences necessary to develop sufficient knowledge and confidence in coding and to use it to teach mathematics (Weber et al., 2022). Elementary teachers require opportunities to develop an understanding of coding in the curriculum and the knowledge needed to use it effectively with children, including enriching their learning of mathematics (Angeli et al., 2016). This working group provided an opportunity for members to share and build their coding capacity, while considering ways to develop coding abilities in elementary educators.

Organizing Framework (YALL)

In preparation for the working group the leaders chose to adopt a framework previously developed and used by Lorraine Godden and Sandy Youmans originally called "Affirming, Yearning, and Learning". The framework order was flipped to begin with Yearnings (What do we want to know/learn? Or why come to this working group? And followed by Affirming - What do we want to know? And what do we already know? The final part of the framework led into action - learning at large group and smaller sub-group levels (LL). The framework structured our engagements over the three days.

Day 1

Yearning (Motivations)

After introductions, table groups were asked to identify their personal goals for attending this working group and to begin to articulate potential shared collective goals. Following the idea of Simon Sinek (2009) the orienting question in the first session was, *"Why (teach) coding in the elementary grades and early years?"*

Responses to this prompt included:

Coding is easier to learn in early grades; Coding opens doors for learners; Kids like robots and will like math; Coding encourages creativity and allows kids to create and problem solve while having fun; Coding normalizes mistakes and debugging as important disposition during problem solving; Coding provides multiple representations and strategies; Mathematics and CT are both human activities but also privileged human activities in our context and so *important to be able to use; Coding as providing opportunities for learners to develop and demonstrate voice, agency, identity and belonging; "Computational science is an offspring of mathematics, so its elements tend to involve sophisticated consolidations of ranges of mathematical concepts, contexts and representations; Coding as a 'silent' or third teacher drawing on Guy Brousseau's ideas; Coding as part of the bridging connections between STEM and Mathematics Education allowing for interdisciplinary and contextually relevant learning (Figure 2).*

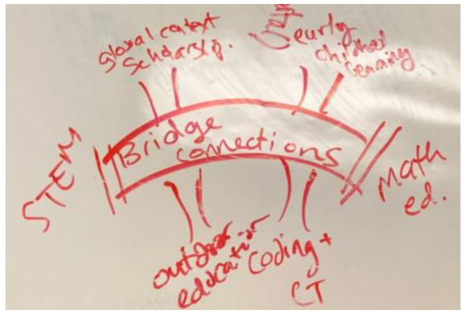


Figure 2. Participants' rendering of relationship between STEM and Math Ed with coding and CT part of the connections along with outdoor education, early childhood learning and global contexts.

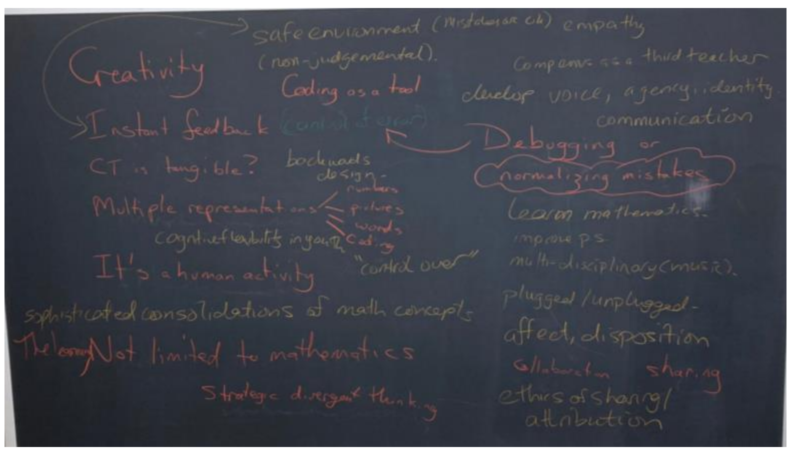
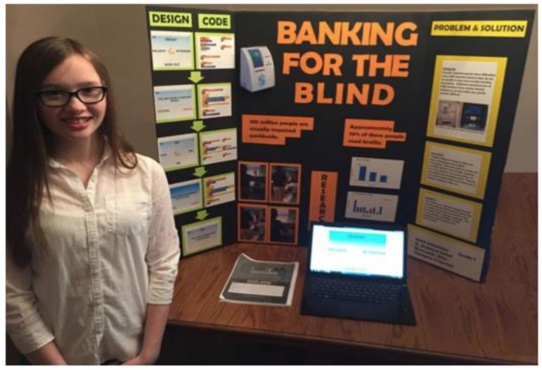


Figure 3. Summary of some of the discussion about why teach coding.

Gadanidis' (2017) five affordances of coding for K-6 math were discussed. These include the following: Agency (opportunity to be in control of one's learning), Access (by differentiation and ability to go explore more complex relationships and representations), Abstraction, Automation (allows automation of some processes), and Audience (shareable). Participants were also encouraged to consider whether they experienced any of these and where they had occurred during the working group.

In connecting to the Symposium theme on equity and computational Modelling and mathematics education, we shared two cases from the Scratch Team blog. The first (Nguyen, 2016) described Katie S., an 8th grade student who used Scratch to help visually impaired individuals to use ATMs using hand gestures. The second example (Singh, 2016) describes the case of Scratch user named Megaman100, a student who is neurodivergent, who developed confidence and technical skills with coding that led to the creation of his own video game characters. Both cases illustrate Gadanidis' affordances and concretely foreground the way learning to code and working towards equity travel together. The Scratch organization also presents several examples of the way the programming environment can be used by learners with disabilities (Adams, 2009). It is important to note that these examples do not include children in the earliest years and the examples are not strictly mathematical..



Katie won the Financial Award at NICEE and second place overall at the National Invention Convention and Entrepreneurship Expo.

Figure 4. Katie S. in front of her Scratch Project presentation: Banking for the blind.

The second aspect of yearning asked what participants hoped to learn in the session (the YALL framework does not imply discrete categorization). Several themes emerged. One of the strongest was focused on teachers and wondered, “How can we build the capacity of pre-service and in-service teachers and guide them in understanding the relationship between mathematics and coding in a holistic way, i.e. to make connections between our knowledge in various fields and math/coding/CT” and to tease out the connections among coding, math and socio-emotional learning. This is a recurrent theme in the research literature over at least the last two decades. The second theme is also recurrent - “How are terms such as coding, computational thinking, coding, algorithmic thinking and mathematical thinking defined, and are they used with the same understanding by different individuals or groups?” Attempting to define computational thinking and mathematical thinking seems to be a necessary step in groups such as this to have productive conversations and is an aspect we picked up on the second day. The third emphasis was on tasks and framed using Dan Meyer’s (2010) now classic analogy repurposed as “What problems can we offer that are “headaches” for which mathematics AND code are the aspirin?”

Affirming

Researchers (and teachers) have learned quite a bit about teaching and learning coding at various grade levels and the challenges and affordances of doing so in relation to mathematics. Research on coding in elementary settings and those focused on mathematics is more diverse than research on coding in the early years reflecting the different developmental and conceptual emphases. As a working group, we brainstormed what was affirmed about coding in the research and our own experiences, or essentially what the group already knew about coding. The main themes that emerged were that coding is characterized as the following:

- a vehicle for creativity
- A good source of productive failure or struggle
- marked by algorithmic sovereignty
- facilitates deep understanding about problem solving
- not computer science
- transforms objects of learning into tools for learning
- develops identity

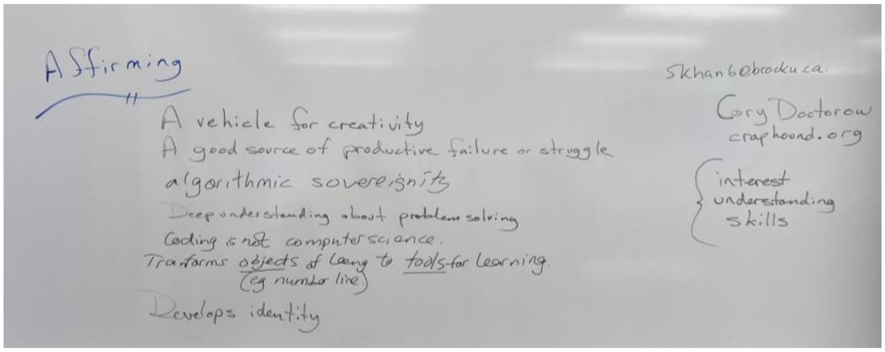


Figure 5. Working group responses about affirming what coding is based on research and our own experiences (or what we already knew about coding).

Contexts Matter

One of the clear reminders that emerged over our discussions was that contexts matter a great deal and differences in contexts are directly related to equity (Guiterrez, 200?) for learners and teachers and researchers. In this section we present outlines of the different local (provincial) and individual contexts in which we work. In coming to appreciate why we might be talking past each other at times and in opening up space for epistemological plurality by recognizing our attentional blindspots due to hyperfocusing and assumed commonality, this work is necessary.

Day 2

Learnings (Initial)

Day 2 began with working group members sharing some of the coding activities they engaged with in their professional settings. Coding activities about grids, goats, and games were shared with the group, ranging from early years to elementary level coding.

Table 1. Working Group Coding Activity Demonstrations: Grids, Goats, and Games

Name	Link	Description
Sandy	https://earlymath.ca/teacher-resources/roots-of-coding-series/	Early years "Roots of Coding" grid activity called "X Marks the Spot" use to develop spatial language used during coding
Iain	https://scratch.mit.edu/projects/828890799/	A beginning computational model with billy goats who eat the sweet, green grass - maybe too much. What is needed to balance this ecosystem? Could the troll help?
Iain	https://scratch.mit.edu/studios/31981209	Coin Flipping and More Oh, My! Exploring probability.
Alex	https://blockly.games/maze	Fun maze :) blockly has other activities too
Irina	https://www.birdbraintechnologies.com/	Resources for Finch robot
Irina	https://www.primotoys.com/	Cubetto

Figure 6. A Finch robot demonstration to our working group.

Day 2 included opportunities to engage in rich discussions about the relationship between mathematics and coding. Topics included the following:

- What is mathematics?
- What is mathematical thinking?
- Does all coding include math?
- What is the relationship between computational thinking and maths?

Insights from our discussions included the following:

- It is important to make a distinction between proto-mathematics vs mathematics
- Coding itself does not bring rich math by itself, math concepts can be learned (engaged with) in the process of coding, but this needs to be intentional and made explicit by educators
- You can teaching meaningful math through coding and you can teach coding while students are learning math
- The word com-putare (computer) derives from Latin and means to take apart and put back together again
- This project is a civilizational transformation project; thus be wary of that

Using Commercial Puzzle Games and Puzzles to Develop Coding Related Processes

Participants were introduced to several modern commercial puzzle games that can be used to develop coding related processes (computational thinking) across the early years through play, inquiry and guided discovery. Many of the games also depend on and develop skills related to spatial reasoning. All of the games involve elements of logical reasoning, sequencing, decomposition and provide multiple opportunities for debugging thinking, unlike screen-based coding environments.

Using games with a sense of computational awareness, i.e. with an awareness of computational thinking processes, allows teachers to work intentionally to ground the language of computational thinking with the concrete enactive experiences and productive disposition developed through game-play and puzzles. Opening up this dimension also opens the door to including many games from across cultures and time-periods - from Mancala to Hex.

Figure 7. Discovering how commercial games can be used to teach coding.

Day 3

Learnings (Sub-groups)

On the third day of our working group, participants split into sub-groups based on their interests. Three distinct sub-groups were formed: 1) Computation tools for mathematics education, 2) The role of coding and computational thinking in mathematics education, 3) Building coding capacity in elementary teachers. Key learnings and wonderings are presented for each of the sub-groups.

Sub-group 1: Computational Tools for Mathematics Education

Sub-group 1 discussed and identified some helpful computation tools for mathematics education and created a flow chart to help education decide which computation tool to use.

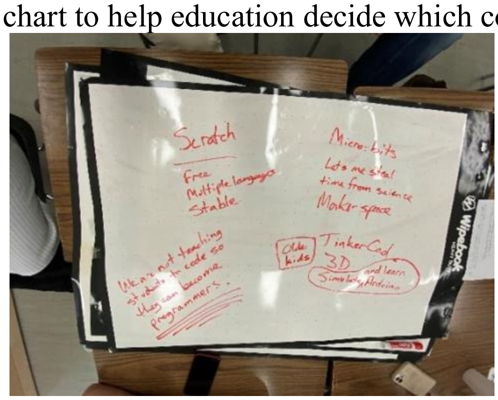


Figure 8. A list of computational tools that can be used for mathematics education.

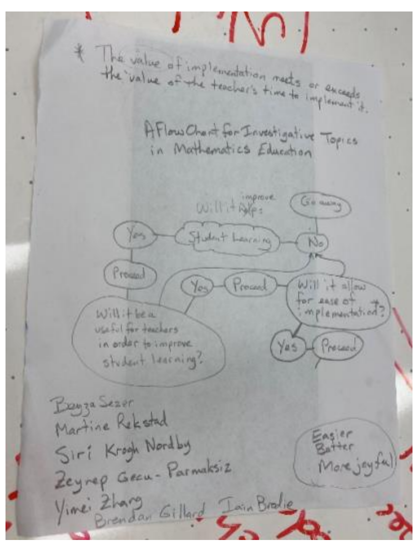


Figure 9. A flowchart to help educators decide which computational tool to use.

Sub-Group 1 concluded their time together by developing the action item and research question identified below.

Action: supporting teachers to use the tools effectively and value the benefits of the tools.

Research Question: How can we motivate students to use computational tools (i.e., especially robotics) to learn mathematics along with play (e.g., joy, engagement)?

Sub-Group 2: The Role of Coding and Computational Thinking in Mathematics Education

Subgroup 2 wrestled with the role of coding and computational thinking in mathematics education and the philosophical question about why we teach coding. Coding was discussed as a new literacy that can be used to promote culturally relevant pedagogy and promote greater equity among students.

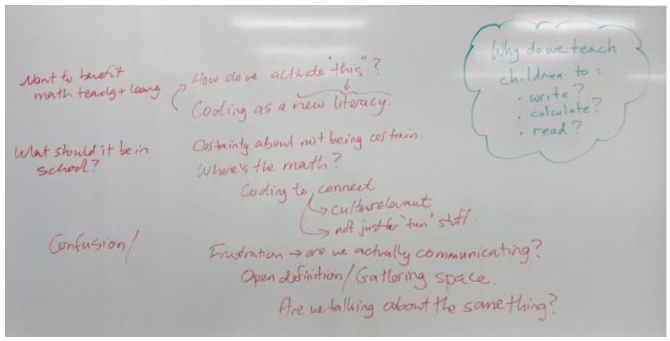


Figure 10. Guiding questions about the role of coding in mathematics education.

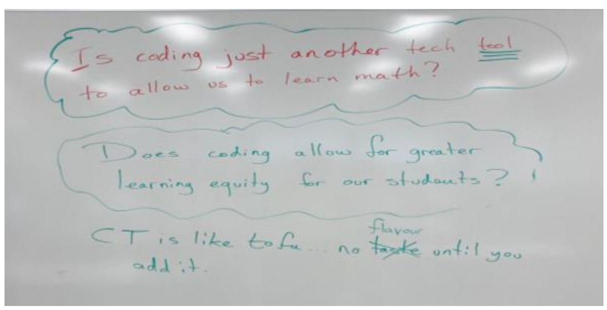


Figure 11. Further questions about coding and mathematics education.

Rather than coming to a consensus about the role of coding and computational thinking in elementary mathematics, Subgroup 2 developed the following questions for consideration:

- What do (interdisciplinary) rich mathematics look like with modern computational thinking in elementary school?
- What mathematics does modern computational thinking make irrelevant? (we commented about physicality of paper-pencil computation)
- What mathematics does modern computational thinking make important?
- What ways of mathematical knowing change as a result of modern computational thinking?
- How does modern computational thinking shift the time spent on relevant mathematical topics?

Subgroup 3: Building Coding Capacity in Elementary Educators

The building coding capacity in elementary educators brainstormed important ideas and strategies for strengthening teacher capacity. Based on these ideas, the sub-group identified the following insights:

- We need to develop as teacher educators ourselves and help pre-service educators become comfortable with learning with and from students (and us)
- We need to move beyond surface level professional development and coding tasks to equip students to be competent coding educators
- We can use the strands of mathematical proficiency to guide the development of a coding course for pre-service teachers

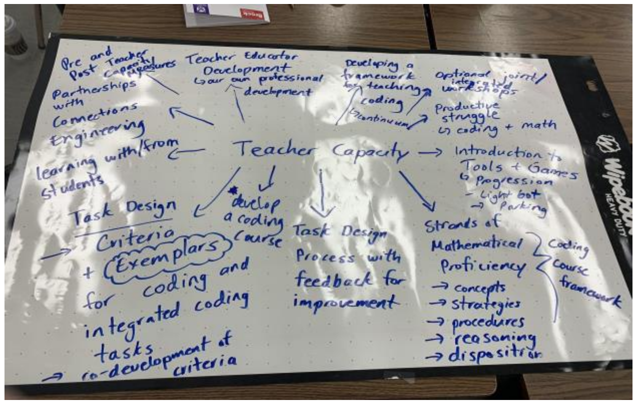


Figure 12. A list of Important ideas and strategies for building teacher capacity in coding.

Subgroup 3 decided on the following action steps:

- Develop a comprehensive and joyful framework for teaching coding
- Use tools and games that facilitate coding progression
- Work on high quality coding tasks and accompanying rubric with relevant criteria

Note: After the Coding Symposium, members of Sub-group 3 continued to meet monthly to develop a Social Sciences and Humanities Research Council Partnership Development Grant called, "Developing Coding Capacity with Elementary Educators".

Acknowledgements: We wish to thank the Social Sciences and Humanities Research Council, the Fields Institute, and Callysto for making this Coding Symposium possible. Our working group greatly benefitted from the opportunity to meet and learn from one another.

Final Thoughts/Reflections

There is a movement in education to include coding in the elementary classroom. We know that coding allows for the learning of skills that are useful in problem-solving, modelling, and could aid in some of the so-called 21st century skills like information and technology literacy, and critical thinking skills. A great deal of attention is being paid to just what type of thinking we are engaged in. Is it computational thinking we are engaged in (see Wing, 2006) computational modelling (see Gadanidis & Namukasa, 2011), computational fluency (see Resnick, 2018), computational participation, or computational literacy?

This is a really important discussion for the future of what and how we teach coding going forward, but it misses the mark for what is needed right now, at this very minute, in the classroom by the teachers who are delivering it to their students. More than anything classroom teachers need to develop deep content knowledge about coding and be introduced to effective instructional strategies for teaching it.

This is the arc that our working group took over the course of our three days together. We debated hotly about what type of thinking was going on when we code and slowly, but surely, we moved towards what teachers needed in order to implement coding in their classrooms. We eventually broke up into sub-groups with some taking on the important thinking about thinking and others working on what supports we need, and importantly don't need, to deliver to our colleagues in the classroom.

References

Adams, J. (2009). Using Scratch to engage students with disabilities. https://scratched.gse.harvard.edu/stories/using-scratch-engage-students-disabilities.html

Bers, M. U. (2022). *Beyond coding: How children learn human values through programming*. MIT Press.

Bers, M. U. (2020). *Coding as a playground: Coding and computational thinking in the early childhood classroom (2e)*. MIT Press.

diSessa, A. A. (2018). Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, *20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

earlymath.ca (n.d.). Roots of Coding. https://earlymath.ca/teacher-resources/roots-of-coding-series/

Floyd, S. (2022). *The Past, Present, and Future Direction of Computer Science Curriculum in K-12 Education Electronic Thesis and Dissertation Repository. 8463. Https://ir.lib.uwo.ca/etd/8463*. https://ir.lib.uwo.ca/etd/8463/

Gadanidis, G. (2017). 5 straight A's for Coding + Math. *Math + Code Zine, 2*(3), http://researchideas.ca/mc/straight-a/

Gadanidis, G., & Namukasa, I. K. (2011). New media and online mathematics learning for teachers. D. Martinovic, V., Freiman, and Z. Karadag (Eds.), *Visual mathematics and cyber learning* (pp. 163-186). Springer.

Kafai, Y. B., & Burke, Q. (2017). Computational participation: Teaching kids to create and connect through code. In *Emerging research, practice, and policy on computational thinking* (pp. 393-405). Springer.

Meyer, D. (2010). Math class needs a makeover. TED Talk https://www.ted.com/talks/dan_meyer_math_class_needs_a_makeover?language=en

Microbit.org (n.d.) https://microbit.org/teach/for-teachers/

Nguyen, M. (2016). Meet the Scratcher: Katie S. Medium.com. https://medium.com/scratchteam-blog/meet-the-scratcher-katie-scheutzow-18464f757f64

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education, 49*(4), 424-461. doi:10.5951/jresematheduc.49.4.0424

Resnick, M. (Sep 2018). Computational Fluency. Medium (adapted excerpt from Lifelong Kindergarten).

Sinek, S. (2009). How great leaders inspire action. TEDx https://www.ted.com/talks/simon_sinek_how_great_leaders_inspire_action?language=en

Singh, S. (2016). How Scratch changed the life of our son with autism. Medium. Com https://medium.com/scratchteam-blog/how-scratch-changed-the-life-of-our-son-with-autism-a0b9ef0f6388

Wing, J.M. (2006) Computational thinking. *Communications of the ACM*, 49, 33-35.