8-PAGE
INSERT
ELEMENTARY
MATH

ONTARIO
ASSOCIATION FOR
MATHEMATICS
EDUCATION

Abacus

Volume 65 • Number 1 • September 2026

A MODEL FOR DOING MATHEMATICS THROUGH CODING!

TABLE OF CONTENTS

In this issue, you'll find:

1. A model for learning and teaching mathematics through coding
2. A lesson sequence for coding in grade 4-6

A Publication of the OAME/AOEM

ABACUS CO-EDITORS



MARC HUSBAND *is an associate professor at St. Francis Xavier University in Antigonish, Nova Scotia. He is a classroom-based researcher, who collaborates with teachers to better understand how mathematics looks, feels, and sounds in elementary classrooms. His work explores how teaching and learning emerge through everyday mathematical activity.*
mhusband@stfx.ca



TINA RAPKE *currently focuses on all grade levels of mathematics teaching and learning (including the education of future teachers). She is interested in how research informs classroom practice and vice versa. She has 20+ years of teaching experience and is currently an associate professor at York University. Dr. Rapke holds a joint doctoral degree in mathematics and education.*
trapke@edu.yorku.ca



CRISTINA DE SIMONE *Cristina is currently a middle school teacher, and pursuing a Ph.D. in mathematics education at York University. She is dedicated to rooting her practice in research, which is helping to inform future teaching.*
TinaDS@yorku.ca

As a new school year begins, many teachers are looking ahead to the opportunities that await them in their mathematics classrooms. If teaching coding feels like one more thing to add to an already full plate, you are not alone! This issue of the *Abacus* is inspired by Dr. Lisa Anne Floyd's research, which explored how pre-service teachers moved from feeling uncertain about coding to feeling confident teaching it. In this issue, Lisa shares an adaptation of her model that teachers can use in their classrooms. In the pages that follow, Lisa highlights experiences that can build confidence in both students and teachers as they explore mathematics through coding.

Through incorporating this model into your practice, if you're new to coding and/or teaching coding, we hope that it may also help teachers to feel more competent to teach coding.

LEARNING AND TEACHING MATHEMATICS THROUGH CODING

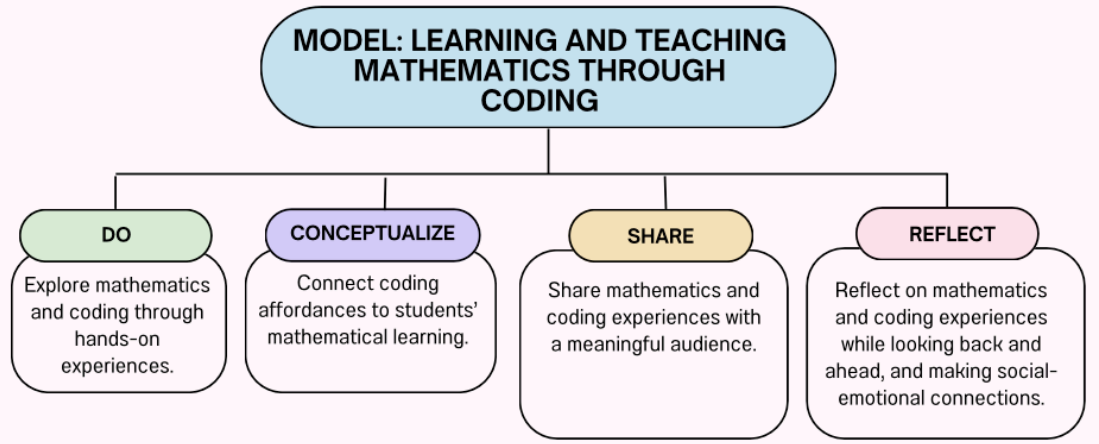
Dr. Lisa Floyd

Progression through this model, which involves four stages—*Do, Conceptualize, Witness/Share, Reflect*—might be seen as semi-linear. This means that students often move back and forth between creating, sharing, reflecting, and revising. Although it makes sense to start with the DO stage, it's important to be flexible and allow students the space to move back and forth between the stages.

During the Do and Conceptualize stages, students explore mathematical ideas by creating, testing, and revising their own work. These ideas draw from the work of Seymour Papert's (1980) theory of constructionism, which involves students building and constructing their knowledge through doing mathematics. Learning-by-making draws from the work of Jean Piaget's (1970) theory of constructivism, with whom Papert worked for many years.

Papert's (1980) vision was for students to experience pleasure from working in a "mathematics-speaking" (p. 47) environment, in which students would be immersed. For Papert, he saw the coding environment he used in his work as "Mathland," which he compared to children spending time in France learning French. In this way, coding can potentially open new possibilities for children to produce complex objects that are personally meaningful and may have previously seemed too difficult. The pre-service teachers in the study noticed that coding seemed to elicit excitement about mathematics, rather than groans, and they observed powerful learning taking place.

The Witness/Share and Reflect stages, while also drawing from Seymour Papert's work, encompasses attributes of computational participation, as described by Yasmin Kafai and Quinn Burke. Kafai and Burke (2014) expanded upon Papert's views about coding, and focus on coding in socially connected ways—which can help to make use of computers in schools to become more creative and collaborative. The pre-service teachers in the study noted that coding in math class resulted in students sharing with one another and building off one another's work. The pre-service teachers themselves appreciated that coding and sharing their applications made them feel they were creating with a meaningful purpose, but also engaging deeply with complex math ideas. Finally, the pre-service teachers noticed that coding naturally helped them to be more flexible and adaptive while teaching. They saw that they could better support a sense of belonging in mathematics, as students had something in common to share, which could then be enjoyed by others.



This model is meant to be flexible. While starting with "DO" is ideal, students may move back and forth between different stages as they engage with mathematics and coding.

Stages in the Model Described with Exemplars

DO. This model involves starting out with having students DO mathematics through coding. Recommendations within this stage include immersing students in rich mathematics through coding in a playful and tangible way, and making mathematics come to life while connecting to existing curriculum. Fostering a supportive and flexible environment that encourages mathematical thinking and making with coding is key.

An Initial Coding Experience for Grades 4–8

A great place to start is creating geometric images through coding. Rather than focusing on the user interface of the selected coding platform, or spending time talking about coding commands and blocks, angles, vertices, side lengths, exterior angles, supplementary angles, etc., throw students into the coding environment with a starter application, a few challenges, and allow them time to make sense of the mathematics and code. Students in Junior grades already have prior knowledge about polygons, so they are starting with something with which they are familiar, but also engaging in a new experience through coding 2-D shapes. As students alter the code and run

their applications, they start to see the connection among the coding, the mathematics, and the output.

In his book, *Mindstorms* (1980), Papert describes students creating similar images with the Logo coding platform. The pre-service teachers involved in my study were taken aback by the fact that students were doing mathematics and coding decades ago, yet they were not afforded the opportunity to do so in their own experiences as young learners.

Below is a series of starter activities, images, and links to code that might be created by a student in Grades 4 to 6, with connections to the Ontario Mathematics Curriculum (2020). The examples below have been created using pseudocode, and the links provided are the same as the pseudocode, but adapted in Scratch, which was used extensively in the study on which this article is based.

About Scratch
Scratch is a web-based application that incorporates blocks, rather than having students type their code, which can be rate limiting.
Students do not need to log into Scratch or create an account, unless they wish to save their work. Check with your school district to determine if having students log into Scratch is an option.

Pseudocode

Pseudocode involves code written in user-friendly language, but can be adapted for use in any coding language. Without actual coding, pseudocode can seem boring and useless at first for students, but it makes more sense once they have some experience with coding.

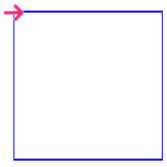
Computer programmers will use pseudocode to plan out their applications, so they may focus on the logic initially, rather than worrying about syntax.

Syntax involves colons, semi-colons, and other structures that specific programming languages use.

In the Ontario Ministry of Education curriculum Teacher Support examples, pseudocode is used, so that teachers may apply the code to the coding application of their choice.

Pseudocode was never meant to replace actual coding. It provides ideas for incorporating the pseudocode into actual applications and is used in Education Quality and Assessment Office (EQAO), since not all students will use the same coding platform.

By sharing corresponding pseudocode during consolidation or at other points along the way, students will become familiar with pseudocode. This may aid them in answering assessment questions that incorporate pseudocode, such as those on EQAO assessments.

Teacher Notes	Starter Code Link/Pseudocode	Image Produced from Code
Provide link to the starter code. Students can click on the green flag to run the application. Then suggest challenges such as those listed below. Some students will alter the code to create their own images. Students will start to notice each other's images and try to recreate those images. It's a beautiful thing!	Scratch Code Link **when green flag clicked…** **pen down** **repeat 4** **move forward 100 steps** **right turn 90 degrees** **end repeat** NOTE: In the link, you will see other blocks of code that: • start the arrow pointing in the right direction • move the arrow to the centre of the stage (0,0) • erase whatever was on the stage previously • wait 0.25 seconds—which is used to slow the arrow's animation down so that students can see each step	

Below are some ways students might alter the starter code. The teacher might list some of these suggestions on a slide, like the samples shown. The order in which the challenges are attempted does not matter, and some students tend to naturally work on a challenge that pushes their own individual thinking.

Altering the code...

bit.ly/simplesquarefloyd

- ✓ Make a square with larger side lengths
- ✓ Make an L shape
- ✓ Make a triangle, pentagon, hexagon...

Read and Alter Existing Code...

bit.ly/simplesquarefloyd

Challenge 1 – Make the side lengths smaller.

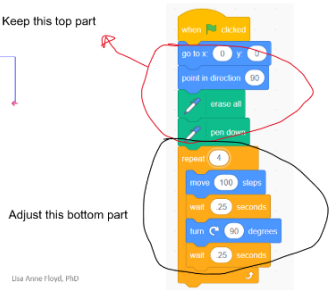
Challenge 2 – Make the shape show.

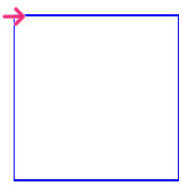
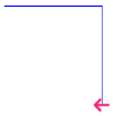
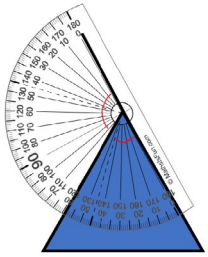
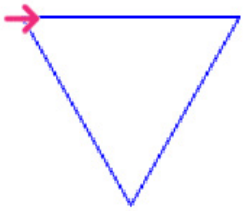
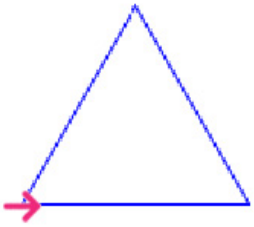
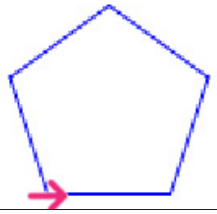
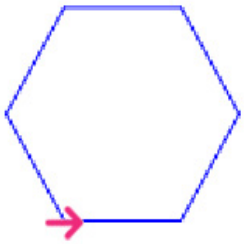
Challenge 3 – Change the shape to a triangle.

Other challenges:

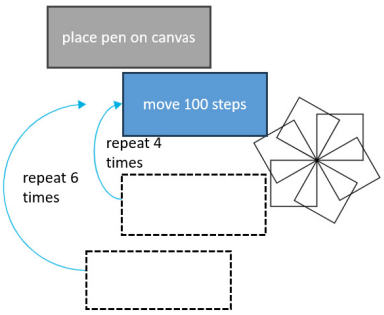
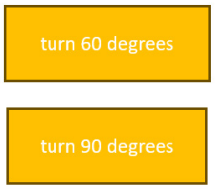
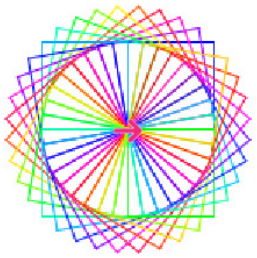
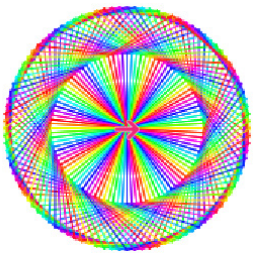
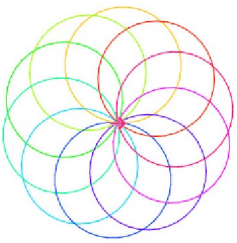
Draw a...

hexagon
pentagon
rectangle



Suggested Challenges Alter the Code to…	Altered Pseudocode with Highlighted Changes and Possible Solution – Code Link	Image Produced from Code
✓ make a square with shorter side lengths	when green flag clicked… **pen down** **repeat 4** **move forward 50 steps** **right turn 90 degrees** **end repeat** Possible Solution Link to Code	
✓ make an "L" shape	**pen down** **repeat 2** **move forward 100 steps** **right turn 90 degrees** **end repeat** Possible Solution Link to Code	
✓ make a triangle ✓ make other regular polygons This presents an opportunity to discuss interior versus exterior angles and supplementary angles, particularly for those students in Grade 6 in Ontario, since that is when those concepts are introduced in the curriculum. When we are walking a shape, we walk the exterior angle.  Supplementary angles can be utilized to determine what the exterior angle would be, if the interior angle of the polygon is known. Students might also notice that the exterior angle can be calculated by dividing 360 by the number of sides/ vertices.	UPSIDE-DOWN TRIANGLE when green flag clicked… **pen down** **repeat 3** **move forward 100 steps** **right turn 120 degrees** **end repeat** Possible Solution Link to Code	
	RIGHT-SIDE-UP TRIANGLE when green flag clicked… **pen down** **repeat 3** **move forward 100 steps** **left turn 120 degrees** **end repeat** Possible Solution Link to Code	
	PENTAGON when green flag clicked… **pen down** **repeat 5** **move forward 50 steps** **left turn 72 degrees** **end repeat** Possible Solution Link to Code	
	HEXAGON when green flag clicked… **pen down** **repeat 6** **move forward 50 steps** **left turn 60 degrees** **end repeat** Possible Solution Link to Code	

Shape	Number Sides and Angles	Interior Angle	Exterior Angle
Triangle	3	60	120
Square	4	90	90
Pentagon	5	108	72
Hexagon	6	120	60

Suggested Challenges Alter the Code to…	Altered Pseudocode with Highlighted Changes and Possible Solution – Code Link	Image Produced from Code
✓ create circle of polygons This directly addresses nested events, which are introduced in Grade 4. There are two repeats: the original shape code is inside of a second repeat that indicates the number of polygons that will be drawn. After each polygon is drawn, the arrow rotates in between, before drawing the next polygon. For a full circle of polygons, the value of the number of repeats multiplied by the value of the degrees turned in between each polygon drawn will need to be 360. For example, if there are 36 polygons drawn, then the degrees turned in between each polygon drawn will need to be 10 degrees (36 x 10 = 360). You will see that students will soon start experimenting with these values. You might see questions similar to this on EQAO: *Code is written to draw a circle of squares. Part of the code is missing.*  *Drag and drop the correct pieces of code to complete the program to draw the image.*  You can see that students gain the core knowledge and skills to answer this question by experimenting with actual code. Had they attempted to answer this question without real coding experience, it would have been more challenging and certainly not as fun. The beauty and joy from coding comes from playing with the mathematics and coding, and experiencing the dynamic nature coding affords (Gadanidis, 2017).	CIRCLE OF SQUARES when green flag clicked… **pen down** **repeat 36** **repeat 4** **move forward 50 steps** **left turn 90 degrees** **end repeat** **left turn 10** **change pen colour by 10** **end repeat** Possible Solution Link to Code	
	CIRCLE OF PENTAGONS when green flag clicked… **pen down** **repeat 100** **repeat 5** **move forward 50 steps** **left turn 72 degrees** **end repeat** **left turn 3.6** **change pen colour by 10** **end repeat** Possible Solution Link to Code	
	CIRCLE OF CIRCLES *Note – Ask your students what they think: Is this a true circle or just a multi-sided polygon? when green flag clicked… **pen down** **repeat 10** **repeat 36** **move forward 10 steps** **left turn 10 degrees** **end repeat** **left turn 36** **change pen colour by 10** **end repeat** Possible Solution Link to Code	

Ontario Mathematics Curriculum Expectations that the teacher can draw upon:

Spatial Sense – geometric reasoning, location, and movement

Grade	Expectation
4	E1.1 identify geometric properties of rectangles, including the number of right angles, parallel and perpendicular sides, and lines of symmetry
5	E1.2 identify and construct congruent triangles, rectangles, and parallelograms
6	E2.3 use the properties of supplementary angles, complementary angles, opposite angles, and interior and exterior angles to solve for unknown angle measures

Algebra – coding skills

Grade	Expectation
4	C3.2 read and alter existing code, including code that involves sequential, concurrent, repeating, and nested events, and describe how changes to the code affect the outcomes
5	C3.2 read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes
6	C3.2 read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code

CONCEPTUALIZE. In the CONCEPTUALIZE stage, students are more deeply experiencing the affordances of coding for mathematics. Students might start to recognize that they are getting immediate feedback through coding, as it is dynamic in nature. They should be encouraged to have agency and adjust their applications as they see fit. Students are exposed to challenging mathematics in accessible ways. Students are being supported in a low-floor, high-ceiling environment with ample opportunities for debugging and collaborating. At this stage, they also might be remixing each other's applications and adding their own creative elements.

Some teachers connect back to how students are feeling. For example, the students might recognize that testing out different approaches and learning from mistakes is an important part of the learning process. This is something that the pre-service teachers in the study noticed quite often in their course and teaching experiences.

I have shared a popular debugging activity with which teachers around Ontario and beyond have been challenging their students. Two examples from this resource are shown below:

Beetle is learning how to draw shapes in Scratch by coding them. Beetle has made a few logic errors. Click on the link to the code and help to debug Beetle's programs!

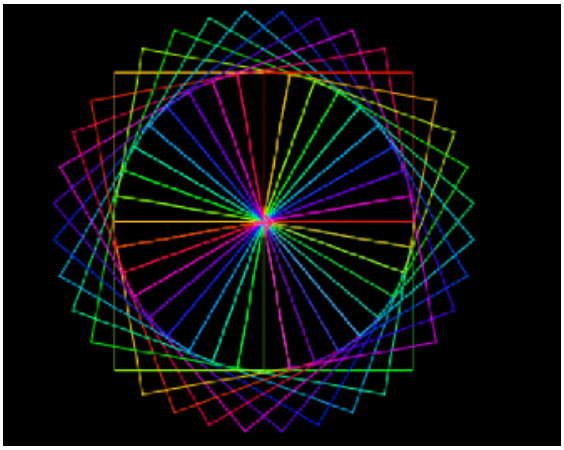
Keep track of how you debug each program on a scrap piece of paper. To reset the program if you make a mistake, just click on the link again.

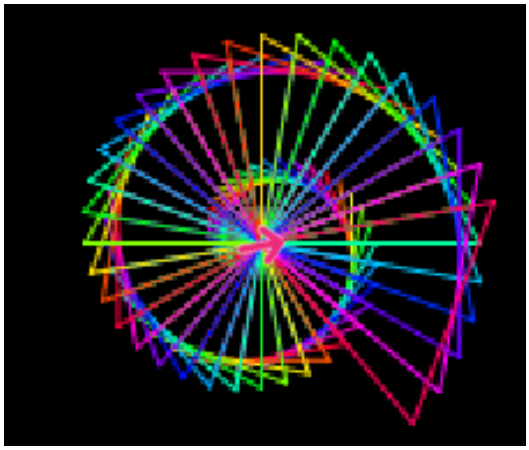
Challenge	Pseudocode Error	Pseudocode Solution
Debugging a Square: Beetle has tried to draw a square, but has an error in their code. Correct the error and click on the green flag to see if you're correct!	when green flag clicked… **pen down** **repeat 4** **move forward 50 steps** **right turn 80 degrees** **end repeat**	when green flag clicked… **pen down** **repeat 4** **move forward 50 steps** **right turn 90 degrees** **end repeat**
Debugging a House: Debug the code that is meant to draw a composite shape.	when green flag clicked… **pen down** **repeat 4** **move forward 50 steps** **right turn 90 degrees** **end repeat** **repeat 3** **move forward 50 steps** **right turn 120 degrees** **end repeat**	when green flag clicked… **pen down** **repeat 4** **move forward 50 steps** **right turn 90 degrees** **end repeat** **repeat 3** **move forward 50 steps** **left turn 120 degrees** **end repeat**

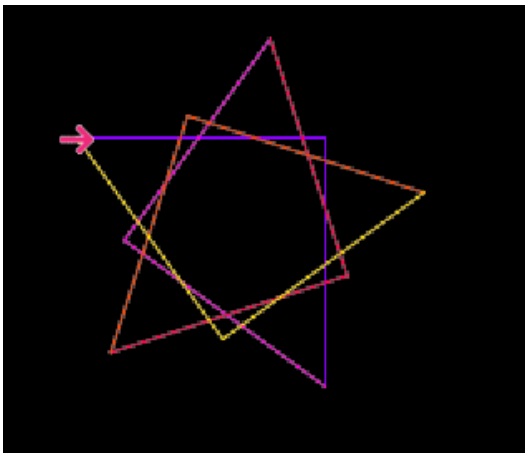
WITNESS/SHARE. While creating their applications, students should be provided with the opportunity to share with their peers, students in other classes, and perhaps family members in mathematics-engagement evenings hosted at the school. Through such experiences, you might find that students begin sharing readily, building off of one another's code, and being more playful and creative with their applications. Teachers can encourage students to explore mathematics in individual ways through examining, sharing, and valuing the works of their peers. Finally, the teachers might ask students about errors they made, how they managed to fix errors, and the learning that occurred as a result of such errors.

This stage can be as simple as students participating in a gallery walk and sharing their applications with one another. Teachers often invite younger students to the classroom for their older students to show their applications, and might encourage the younger students to play with the code and make changes to see what happens to the output.

For one Grade 6 class, I had the students write code for Geometric Mindfulness Applications, where they created a series of animations that incorporated geometric shapes by altering the original square code shared in the DO stage in this article. Many students added music and other special effects. Screen prints from a possible student project are shown below. Creating meaningful opportunities for students to share with a purpose provides students with a sense of agency.







REFLECT. At some point, it is important to allow time for students to reflect upon their mathematics and coding experiences, and to consider how their understanding of mathematics has changed, the skills they learned through playing with mathematics through coding, and if these skills may transfer to other mathematics learning and perhaps other subject areas.

Below is an example of a prompt that might elicit meaningful discussions related to the students' learning of mathematics through coding. Students might reflect upon such prompts individually, then discuss their responses in small groups, or they could create written reflections.

Sample Prompts:

Consider a situation where you experienced a bug in your program.

a) How did you feel about the error initially?
b) What steps did you take to fix the error?
c) What did you learn from the error?
d) Did you seek help from a classmate and, if so, how did they support you?
e) What tips would you give to a classmate who has an error in their code?

In Conclusion

Wherever you are with your coding journey in mathematics education is okay! The idea is that you can try these examples, iterate, and try again. That's the nature of teaching and learning mathematics. Our journey is never done, but what a beautiful view it can be along the way, when we can capture the affordances of coding for learning mathematics.

REFERENCES

Floyd, L.A. (2025). *Mathematics and coding in teacher education.* Western University.

Gadanidis, G. (2017). Five affordances of computational thinking to support elementary mathematics education. *Journal of Computers in Mathematics and Science Teaching, 36*, 143–151.

Kafai, Y.B., & Burke, Q. (2014). *Connected code – Why children need to learn programming.* Massachusetts Institute of Technology.

Ontario Ministry of Education. (2020). *The Ontario curriculum, grades 1–8: Mathematics.* King's Printer for Ontario. www.dcp.edu.gov.on.ca/en/curriculum/elementary-mathematics

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. In *Basic Books* (Vol. 1). doi.org/10.1016/0732-118X(83)90034-X

Piaget, J. (1970). *Science of education and the psychology of the child.* (Translated from the French by Derek Coltman). Orion Press.

The paper-folding directions and images in *Abacus 64*(3) & *64*(4) came from the following sources:

Duatepe-Paksu, A. (2017). Constructing a rhombus through paper folding. *International Journal of Mathematical Education in Science and Technology, 48*(5), 763–767.

Duatepe-Paksu, A. (2018). Construction of quadrilaterals with paper folding. PAÜ-BAP 2017HZDP005 Project Report.

CALL FOR *ABACUS* SUBMISSIONS

Send your teaching-strategy ideas, problems, and student solutions to the *Abacus* Co-Editors via email abacus@oame.on.ca.