Drawing Sequences

Michael Todd Edwards, *Miami University*
Alex Mains, *Mandan High School - Mandan, North Dakota*
Steve Phelps, *Madiera High School & University of Cincinnati*

Abstract: *In the following article, the authors discuss ways programming can be used to foster creative thinking and inquiry-oriented exploration of number patterns and their properties. Using Trinket (`https://trinket.io/`), a freely available on-line coding platform, students write and run Python code to animate number sequences as they engage in mathematics aligned to Common Core standards.*
Keywords. *Programming, inquiry, geometry, pattern*

1 Introduction

The Common Core State Standards for Mathematics (CCSSO, 2010) recommends the exploration of number sequences in the fifth grade. In Operations and Algebraic Thinking (5.OA), CCSSM expects students to "generate two numerical patterns using two given rules," and to "identify apparent relationships between corresponding terms" (p. 35). Papert (1980) and others (Jang and Lew, 2011; Lewis and Shaw, 2012) have found computer programming to be a powerful vehicle to help students in the early grades make connections between numerical patterns and geometric objects. In the following paragraphs, we discuss ways programming can be used to foster creative thinking and inquiry-oriented exploration of number patterns and their properties. Using Trinket (`https://trinket.io/`), a freely available on-line coding platform, students write and run Python code to animate number sequences as they engage in mathematics aligned to Common Core standards. Trinket is freely accessible from any web browser on any device and requires no logins or software installation.

2 A First Example

Prior to engaging them in coding, we introduce students to a process that generates drawings from numerical sequences. We distribute grid paper and present the following repeating sequence in a whole class setting.

$$1, 3, 5, 7, 9, 1, 3, 5, 7, 9, \ldots$$

As we engage in the drawing process, we interpret each term of the sequence as the length of an individual line segment. We demonstrate this with students in a step-by-step manner, using grid paper to draw segments one at a time. Drawing each segment from left to right on the grid paper, we rotate our paper clockwise 90 degrees after each segment is drawn (so that our papers are always oriented in portrait or landscape format). Figure 1 illustrates grid paper with the first term of the aforementioned sequence already drawn.

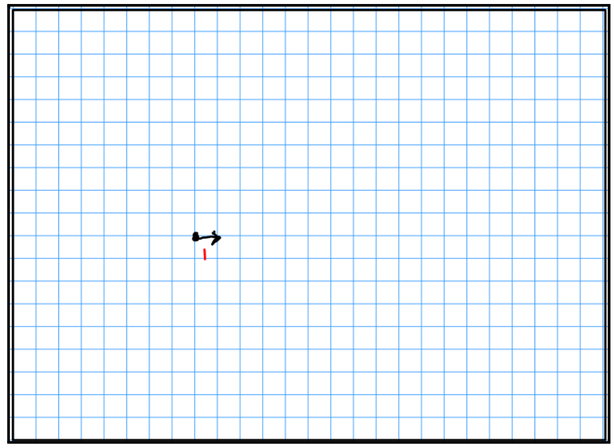


Fig. 1: *Initial grid provided to students.*

Using grid paper, students turn the paper clockwise 90 degrees (i.e., left to right), then draw the next term in the sequence (i.e., 3) as shown in Figure 2.

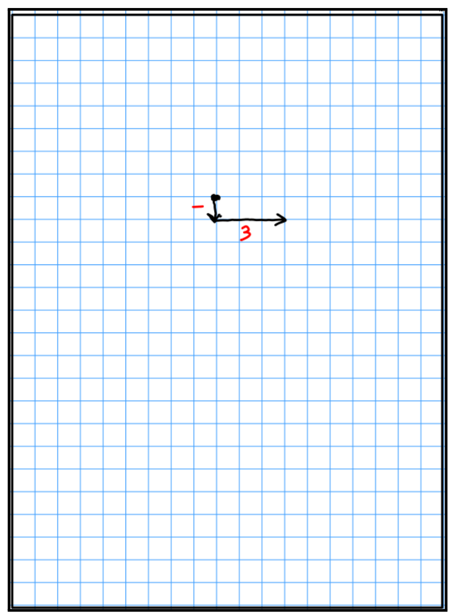


Fig. 2: *Drawing of the first 2 terms of sequence* $1, 3, 5, 7, 9, 1, 3, 5, 7, 9, 1, \ldots$.

Students continue this process, rotating their grid paper 90 degrees clockwise then drawing a new segment with length corresponding to the next term. Figure 3 illustrates a drawing of the first 6 terms of the sequence.
After drawing the first few steps, we ask students to predict what will happen as more terms are drawn. For instance,

- Q1: Will a pattern emerge as we continue to draw terms?
- Q2: If we draw the first 10 terms, will we need another sheet of paper?
- Q3: As we draw more terms, will the segments eventually cross?

At this point, we ask students to continue drawing terms of the sequence on their own to explore their predictions. As students draw, a pattern exhibiting four-fold rotational symmetry emerges, as

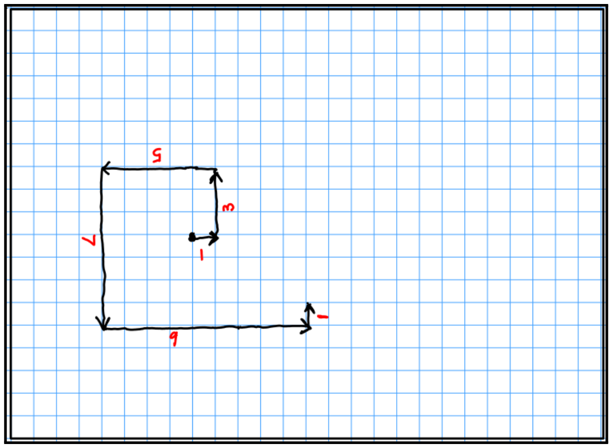


Fig. 3: *Drawing of the first 6 terms of sequence* $1, 3, 5, 7, 9, 1, 3, 5, 7, 9, 1, \ldots$.

shown in Figure 4 (i.e., Q1 is answered). Students see that the drawing fits on a single page (i.e., the answer to Q2 is "no"), tracing the same path after the repeated terms 1, 3, 5, 7, and 9 are drawn 4 times. Additionally, students recognize that the segments corresponding to terms 5 and 9 cross (i.e., the answer to Q3 is "yes").

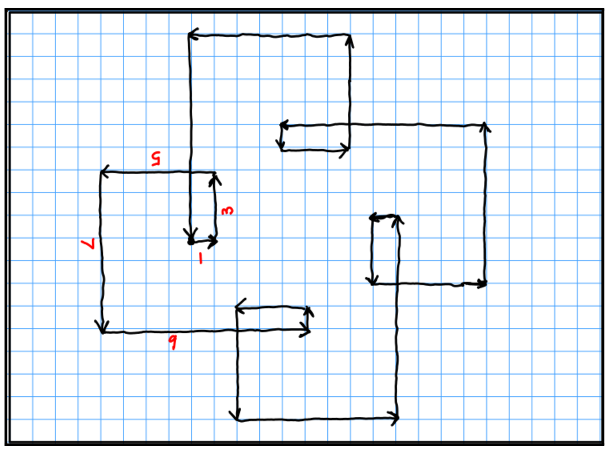


Fig. 4: *A complete drawing of the periodic sequence* $1, 3, 5, 7, 9, 1, 3, 5, 7, 9, 1, \ldots$.

This first example encourages students to generate more questions, more sequences, and more drawings. For instance, consider the following.

- Q4: Is it possible to construct sequences with drawings that are familiar, closed shapes? For instance, is there a sequence that generates a rectangle? A square? A plus sign?
- Q5: Is it possible to construct a sequence that generates a pattern that never repeats?

At this point, after students have a solid understanding of the drawing process, we encourage them to explore questions as they generate their own examples.

3 Student Exploration

3.1 Types of Sequences

As students create their own number patterns, we remind them that sequences can include patterns from one term to the next. For instance, sequences are arithmetic when one can add a fixed value (i.e., fixed difference) to any term to obtain the next term. For instance, the terms $2, 6, 10, 14$ suggest a fixed difference of 4 (i.e., a "+4" sequence). Sequences are geometric when one can multiply any term by a fixed value (i.e., fixed ratio) to obtain the next term. For instance, the terms $2, 6, 18, 54$ suggest a fixed ratio of 3 (i.e., a "+3" sequence). Additionally, terms of sequences can grow in different ways. For instance, terms can repeat (e.g., $1, 3, 5, 7, 1, 3, 5, 7, 1$, etc.) or diverge (e.g., $1, 3, 5, 7, 9 \ldots$).

3.2 Student Examples

As students generate their own sequences and their own drawings, they share their findings with their classmates in whole group settings. We encourage students to look for connections between representations as they seek to prove why certain patterns appear within and between drawings. For instance, students find that multiplying terms of a sequence by a fixed value creates scaled copies of their original drawings (i.e., similar shapes). This is illustrated with drawings generated by repeating sequences $1, 3, 5, 1, 3, 5, \ldots$ and $2, 6, 10, 2, 6, 10, \ldots$ depicted in Figure 5.

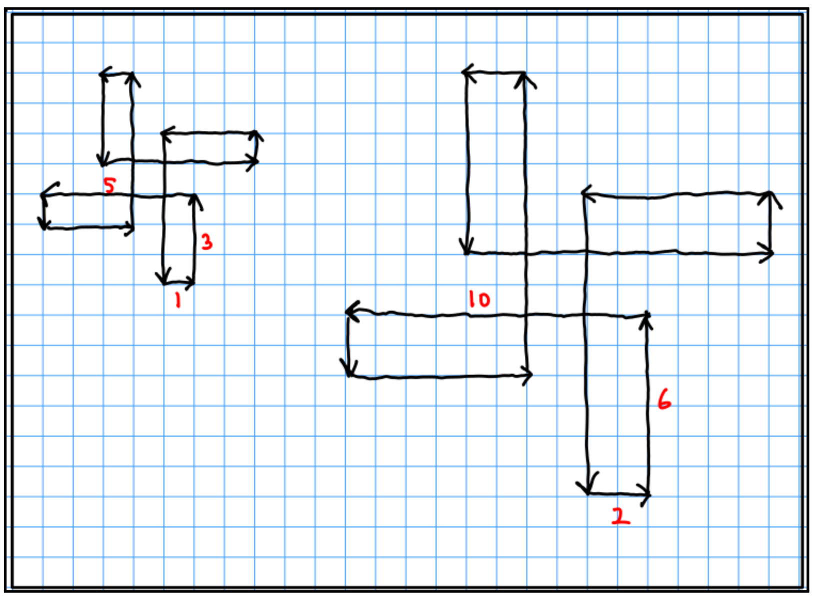


Fig. 5: *Multiplying terms of a sequence by a fixed value results in scaled drawings.*

Others find that familiar shapes can be drawn using simple, repeating patterns. Students often "work backwards" - creating an initial drawing first, then associating a number sequence to the drawing. Such an approach was used to generate the square, rectangle, and "plus" shape (i.e., dodecagon) illustrated in Figure 6.

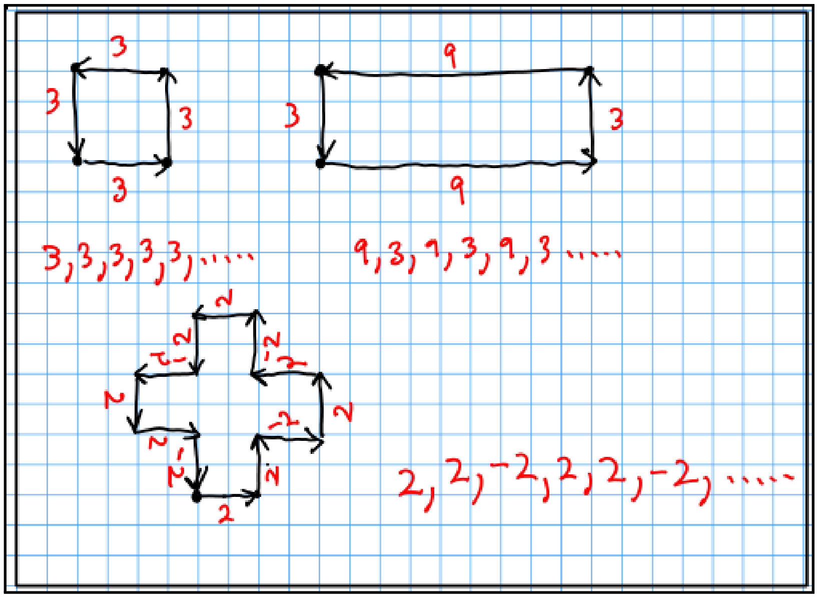


Fig. 6: *Sequences may generate square, rectangular, and plus-shaped drawings (i.e., the answer to Q4 is "yes").*

Sequences that continue to grow will quickly fall outside the bounds of their grid paper (the answer to Q5 is "yes"). By labeling lengths of line segments in such drawings, students discover additional numeric patterns as highlighted in Figure 7.

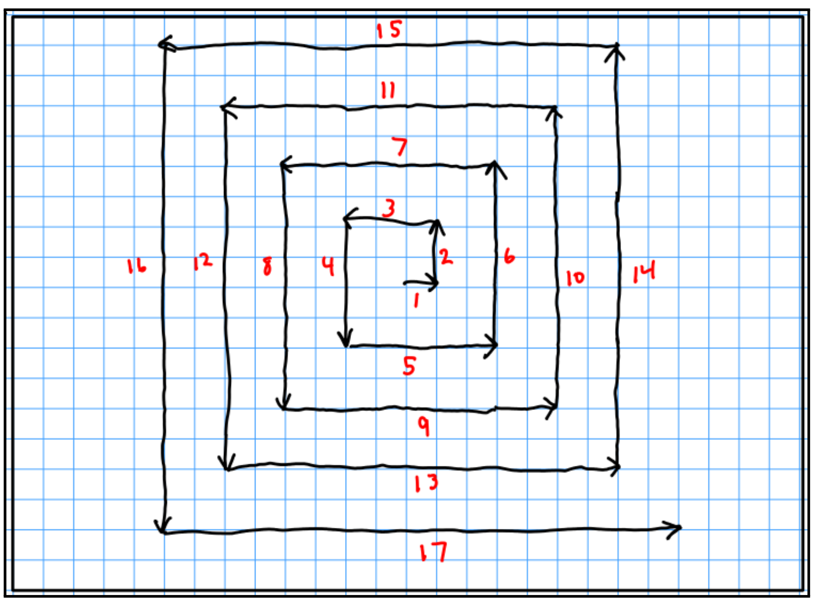


Fig. 7: *More patterns are uncovered when lengths of segments are labeled.*

4 Generalization with PYTHON Programming

After spending time exploring drawings and sequences with numerous sheets of grid paper, teachers and students recognize a need to generate drawings more quickly. Drawing sequences by-hand is error-prone and slows the process of hypothesis testing various pattern conjectures. We have found that providing students with sample PYTHON code is an invaluable tool in the sequence exploration process. Figure 8 provides sample PYTHON code we initially present to our students.

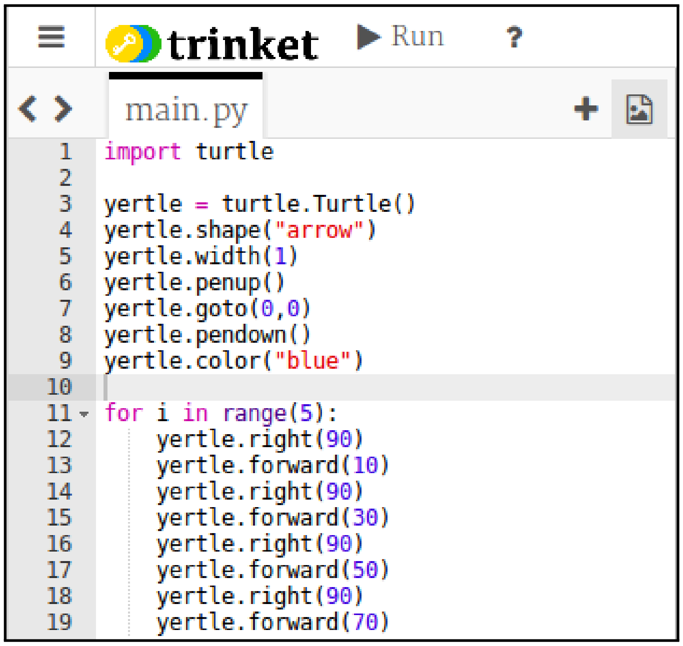


Fig. 8: *Sample PYTHON code (freely available at* `bit.ly/drawingsequences`

Clicking on the run button within the `trinket.io` environment, students run our initial PYTHON code "as-is." Doing so results in the drawing shown in Figure 9.

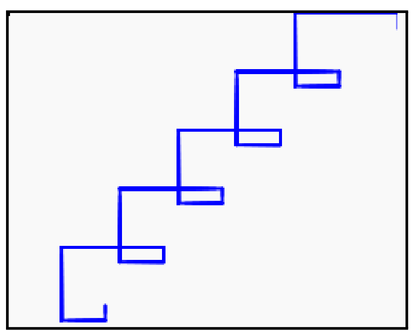


Fig. 9: *Drawing generated by initial PYTHON code.*

Rather than telling students precisely what the code does, we encourage them to explore the code by repeatedly modifying and running it. For students who need help getting started, we ask them to change various snippets of code within the program and observe changes in the resulting drawings. For instance,

- In line 5, what does changing `yertle.width(1)` to `yertle.width(8)` do?
- In line 7, what does changing `yertle.goto(0,0)` to `yertle.goto(50,50)` do?
- In line 9, what does changing `yertle.color(''blue'')` to `yertle.color(''green'')` do?

Students quickly note that lines 1-9 of the program control attributes of the drawing such as line thickness, initial position, and color, while lines 11-19 control the sequence to be drawn. Modifying the range parameter in line 11 from 5 to 3 results in the drawing shown in Figure 10.

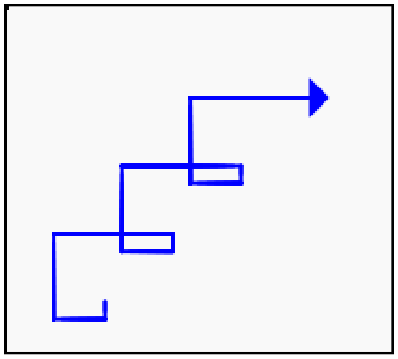


Fig. 10: *Resulting drawing when range parameter is changed to 3.*

In this manner, through exploration, students determine that the range parameter controls how many times a cluster of sequential terms (e.g., $10, 30, 50, 70$) is drawn. When the range parameter is set to 3, three loops are drawn by the code. When the parameter is 5, five loops are drawn. The original code draws the terms $10, 30, 50, 70, 10, 30, 50, 70, 10, 30, 50, 70, 10, 30, 50, 70, 10, 30, 50, 70$. Next, we challenge students to modify the original code to draw the shape generated by the first sequence provided in this paper - namely, $1, 3, 5, 7, 9, 1, 3, 5, 7, 9, \ldots$ Noting that the command `right(90)` is repeated after every forward command, students speculate that `right(90)` rotates the paper 90 degrees clockwise while the `forward(x)` command draws a segment x units long. Students copy and paste the last 2 lines of code (i.e., lines 18-19) and paste them to the end of the program, modifying the last line to read `yertle.forward(90)`. When the code is executed, as shown in Figure 11, it generates a shape similar to the one students generated by-hand in the first portion of our activity.

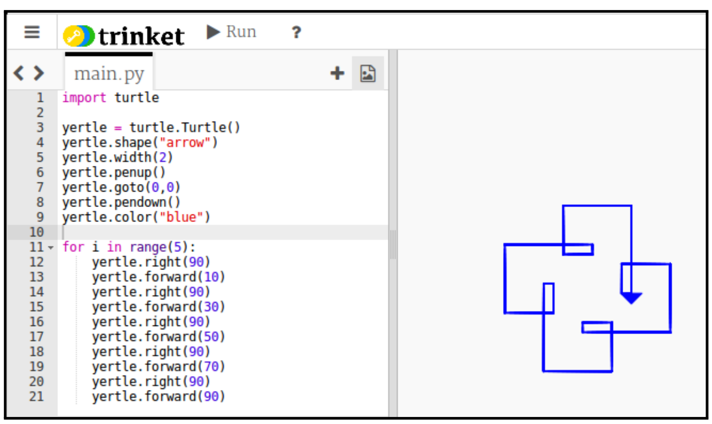


Fig. 11: *Students add two lines of code to draw the shape formed by the first example sequence.*

Once students grasp the basics of the code editing and execution features within trinket.io, they are able to explore a wide variety of questions and sequences.

5 Further Exploration Ideas

Using the "What-If Not" approach (Brown and Walter, 2005) and PYTHON, students can explore the effect of changing various attributes of the original sequence drawing process. For instance, what would happen if we rotated our paper at angles other than 90 degrees? The code highlighted in Figure 12 fixes the length of each line segment to be 100 units and rotates the grid 72 degrees after each segment is drawn.

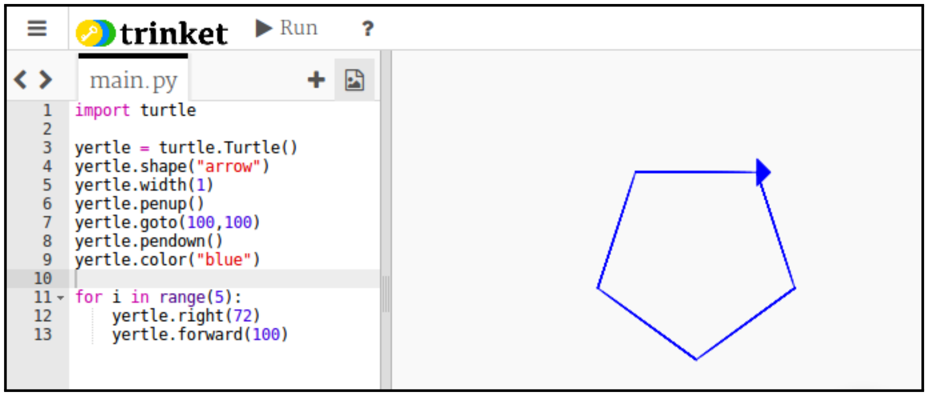


Fig. 12: *Drawing generated by drawing the sequence* $100, 100, 100, \ldots$ *with grid rotations of* 72 *degrees.*

More interesting results are generated by constructing a sequence with terms representing angle measures of grid rotations. For instance, consider the following periodic sequence.

$$10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 70^\circ, 80^\circ, 90^\circ, 100^\circ, 110^\circ, 120^\circ, 130^\circ, 140^\circ, 150^\circ, 160^\circ, 170^\circ, 10^\circ, 20^\circ, 30^\circ \ldots$$

Fixing segment lengths as 40 units and rotating by the angle measures in the aforementioned sequence, generates the drawing shown in Figure 13.

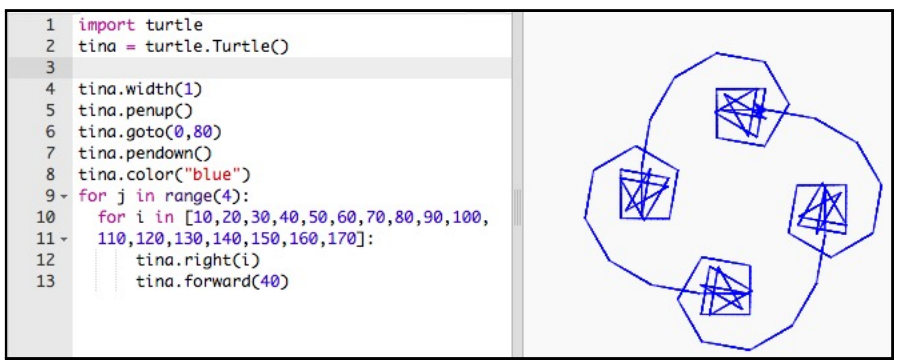


Fig. 13: *Python code (left) and drawing (right) generated by a sequence of angle measures.*

6 Summary

In the preceding discussion, we have illustrated ways in which teachers and students can make connections between numerical patterns and geometric objects by generating drawings from numerical sequences. Specifically, we've highlighted the use of Python programming as a tool to edit, write, and run code that animates number sequences. As students use the software, they are encouraged to see mathematics as an exploratory discipline, one in which experimentation and inquiry lie at the heart of understanding. When teachers provide students with opportunities to construct and analyze their own sequences, they enable students to take ownership of their own learning in a fun, creative, and engaging context while addressing curricular recommendations set forth by Common Core State Standards for Mathematics (CCSSM).

References

Brown, S. & Walter, M. (2005). *The art of problem posing*. Mahwah, NJ: Lawrence Erlbaum.

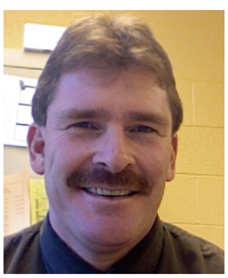
National Governors Association Center for Best Practices & Council of Chief State School Officers (CCSSO). (2010). *Common Core State Standards for Mathematics*. Washington, DC: Authors.

Jang, I. O., & Lew, H. C. (2011). Case studies in thinking processes of mathematically gifted elementary students through Logo programming. *Work 4*, 9.

Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 57-62). ACM.

Papert, S. (1980) *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Michael Todd Edwards, `m.todd.edwards@gmail.com`, teaches mathematics methods courses at Miami University in Oxford, Ohio. He is the co-editor of The North American GeoGebra Journal, The Ohio Journal of School Mathematics, and CITE-Math.



Steve Phelps, `sphelps@madeiracityschools.org`, teaches at AP statistics and computer science courses at Madeira High School in Madeira, Ohio. He also teaches teachers through the University of Cincinnati MAT program. He co-facilitates monthy Cincy Math Circle meetings and co-directs the GeoGebra Institute of Ohio.

Alex Mains, `MAINSAM@miamioh.edu`, is a high school math instructor at Mandan High School in Mandan, North Dakota, teaching Algebra 1 and Geometry. His interests include engaging students in inquiry-based teaching methods and making mathematics accessible, challenging, and fun for all learners.