

MATH + CODING, Grades 6-10

in a collaborative learning setting

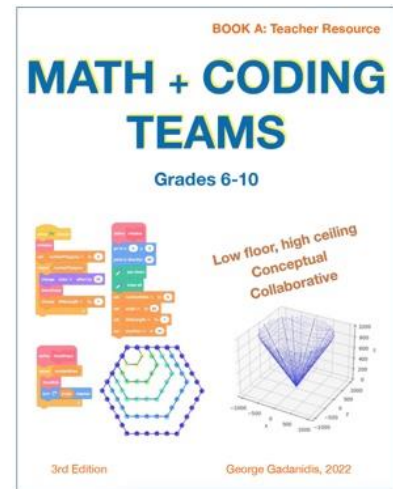
by George Gadanidis, Western University

OVERVIEW

This session will introduce a research-based resource offering:

- low floor & high ceiling tasks
- coding starting points to bring math to life dynamically
- conceptual surprises & insights
- a revitalized classroom culture

Participants will receive a free PDF copy of the resource, *Math + Coding Teams, Gr. 6-10*.



If you find this resource useful, you may book a free 45-60 minute online session for your district leads & teachers, and receive a free district licence.

PRESENTER

George Gadanidis is professor of mathematics education at Western University.

He has worked for many years in research and outreach classrooms in Ontario (as well as in Brazil), collaborating with educators to design mathematics experiences that offer students a sense of mathematical wonder.



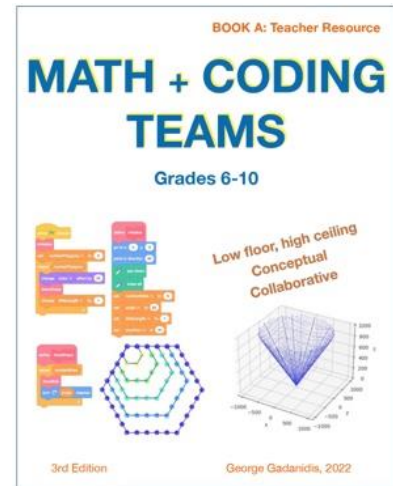
You may see more about his work at imaginethis.ca

MATH + CODING TEAMS

Math + Coding activities offer:

- Rich math + coding tasks
- Low floor, high ceiling engagement
- Conceptual understanding
- Collaborative learning

Activities are based on classroom research.



USING THIS RESOURCE

Math + Coding Teams may be used in 2 complementary ways:

- For mathematics education, to bring math concepts and relationships to life dynamically through code.
- For computer science education, to introduce coding concepts in the context of solving mathematics problems.

The tasks may be used as:

- Collaborative team activities.
- Extension or enrichment activities.
- Starting points for project-based learning.

Math + Coding Teams includes 2 Books:

- *Book A* (120 pages) is designed as a teacher resource. It contains: (a) Student Tasks & (b) Teacher Notes (and solutions).
- *Book B* (56 pages) is designed as a student resource. It contains only the Student Tasks. Book B is intended as a resource that may be shared with students.

ORGANIZING TEAMS

- Teams of 2-4 students.
- A non-competitive atmosphere.
- Collaborative problem-solving.
- A sense of common purpose.
- No team member left behind.

A culminating sharing of learnings and wonderings.

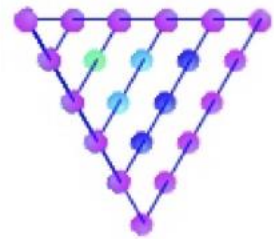
LOW FLOOR, HIGH CEILING LEARNING

Math + Coding Teams tasks are designed to offer all students starting points for engaging mathematically and with code.

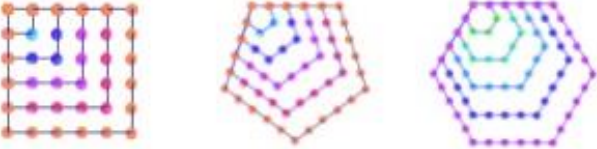
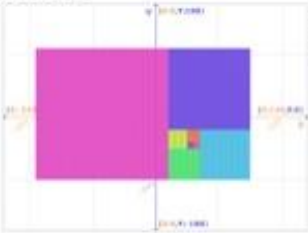

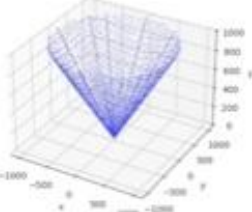



Tasks also offer opportunities for students to extend their learning by asking their own questions and using code to investigate new directions.

Keep the following in mind as you support students in their learning.

- **Code that works.** Notice that students are given some code that works.
- **Solving puzzles.** The activities often involve editing this code to create different representations, such as drawing dots for different triangular numbers, or drawing dots for different polygonal numbers (as shown on the right).
- **Temptation to explain.** There will be a temptation to explain the code to students, or to start by teaching them the different parts of code. Please let students learn by experiencing the pleasure of problem solving, surprise and insight.
- **Understanding code.** By solving such problems, students learn about the meaning and purpose of various code blocks, and start to learn how to write their own code.
- **Understanding mathematics concepts.** The code dynamically represents mathematical concepts and relationships. This brings math concepts to life, makes them manipulable, and gives them a tangible feel.
- **Extending tasks.** Encourage students to ask and investigate their own questions and, if they are interested, to work in more than one coding environment. Most tasks are presented in both Scratch and Python.



CONTENTS

Math + Coding Teams	4	
1. Polygonal Numbers + Scratch		
– Student Task	7	Math: Growing patterns; geometric patterns; algebraic expressions; polygon properties
– Teacher Notes	18	Scratch: Turtle graphics; repeat; sub-programs; variables; lists
		
2. Golden Ratio + Scratch		
– Student Task	24	Math: Growing patterns; geometric patterns; algebraic expressions; history
– Teacher Notes	28	Scratch: Turtle graphics; repeat; sub-programs; variables; lists
		
3.A Sumerian Triples + Scratch		
– Student Task	35	Math: Side relationships of right triangles; "Pythagorean" triples; graphical representations; history; [complex numbers; vectors]
– Teacher Notes	36	
3.B Sumerian Triples + Python		
– Student Task	42	Scratch: Nested repeat; conditional statements; sub-programs; variables; lists
– Teacher Notes	43	Python: Lists; nested repeat; conditional statements; 2D plots; [3D plots]
		 
4.A Infinity + Scratch		
– Student Task	49	Math: Visual representations of fractions; shrinking patterns; limit of an infinite series; history
– Teacher Notes	50	
4.B Infinity + Python		
– Student Task	53	Scratch: Turtle graphics; sub-programs; variables; sprite motion
– Teacher Notes	54	Python: repeat; variables; 2D plots; tabular output
		  

5.A. Linear relations + Scratch

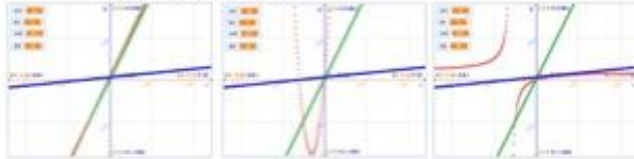
- Student Task
- Teacher Notes

59
63

5.B Linear relations + Python

- Student Task
- Teacher Notes

69
73



Math: Linear & non-linear relations; “operations” on linear equations; graphical representations of relations

Scratch: Nested repeat; conditional statements; sub-programs; variables

Python: Lists; nested repeat; conditional statements; 2D plots

6.A. Finding Primes + Scratch

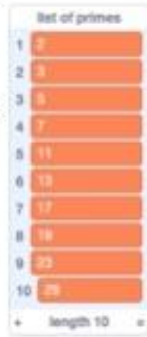
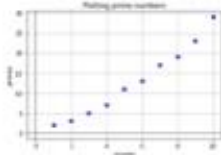
- Student Task
- Teacher Notes

80
86

6.B Finding Primes + Python

- Student Task
- Teacher Notes

92
98



Math: Prime numbers; number theory; algebraic expressions; modulo arithmetic; history

Scratch: Repeat; conditional statements; sub-programs; variables; code efficiency; modulo arithmetic

Python: Lists; repeat; conditional statements; 2D plots; code efficiency; modulo arithmetic

7.A Estimating Pi + Scratch

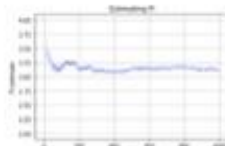
- Student Task
- Teacher Notes

102
111

7.B Estimating Pi + Python

- Student Task
- Teacher Notes

115
119



Math: Meaning of Pi; experimental probability

Scratch: Repeat; lists; conditional statements; sub-programs; variables

Python: Lists; repeat; conditional statements; 2D plots



Mathematics Knowledge Network
Réseau de connaissances en mathématiques