MATH + CODING, Grades 6-10

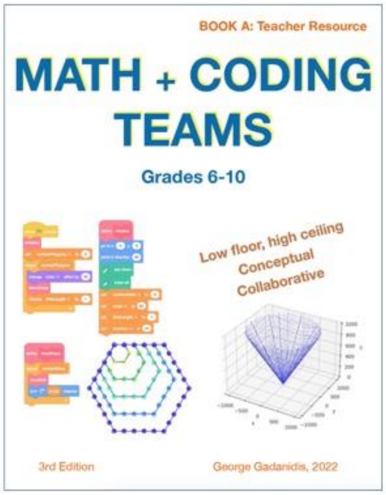
in a collaborative learning setting

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



BOOK A: Teacher Resource

MATH + CODING TEAMS

Grades 6-10

Low floor, high ceiling
Conceptual
Collaborative

3rd Edition

George Gadanidis, 2022

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.

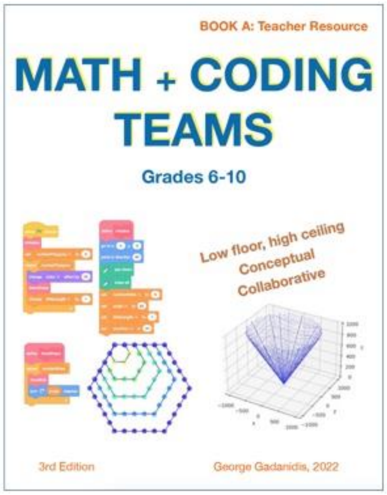
PRESENTER

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 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.

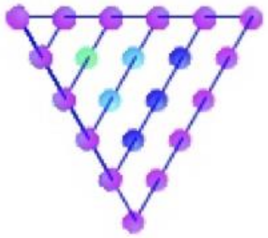
LOW FLOOR, HIGH CEILING LEARNING

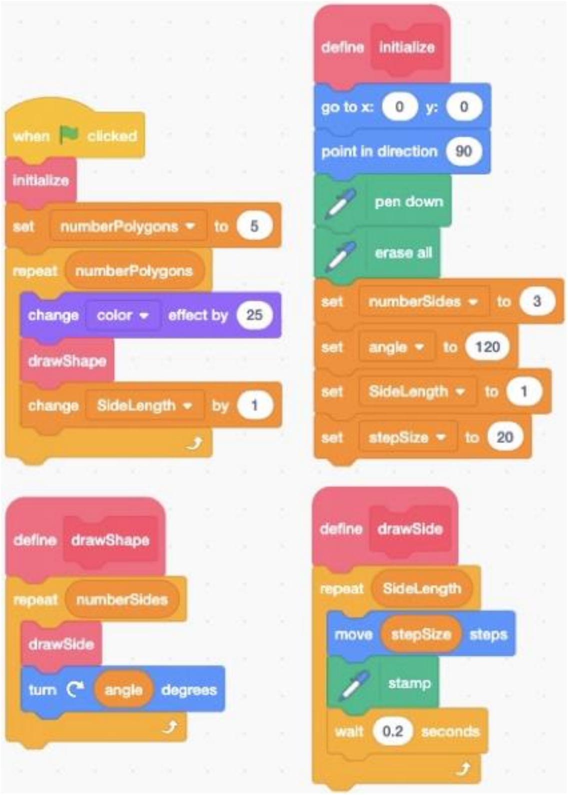
Math + Coding Teams tasks are designed to offer all students starting points for engaging mathematically and with code.

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.

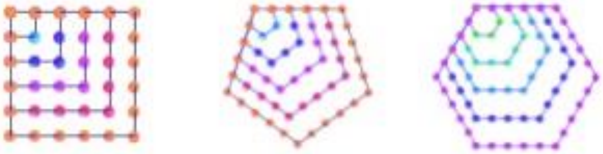
- **Code that works.** Notice that students are given some code that works.
- **Solving puzzles.** The activities often involve editing this code to create different representations, such as drawing dots for different triangular numbers, or drawing dots for different polygonal numbers (as shown on the 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.

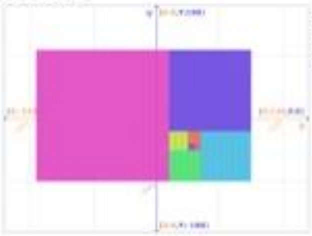


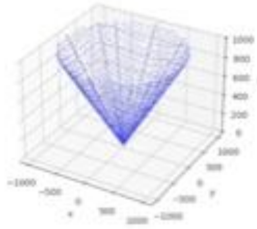


```
when ⚑ clicked
initialize
set numberPolygons ▾ to 5
repeat numberPolygons
    change color ▾ effect by 25
    drawShape
    change SideLength ▾ by 1

define initialize
go to x: 0 y: 0
point in direction 90
pen down
erase all
set numberSides ▾ to 3
set angle ▾ to 120
set SideLength ▾ to 1
set stepSize ▾ to 20

define drawShape
repeat numberSides
    drawSide
    turn ↻ angle degrees

define drawSide
repeat SideLength
    move stepSize steps
    stamp
    wait 0.2 seconds
```

CONTENTS



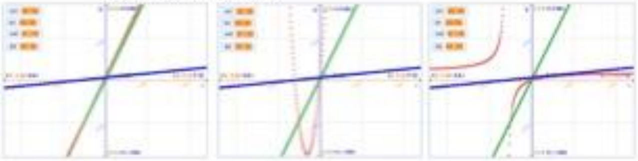




goal 400

step 0.195313

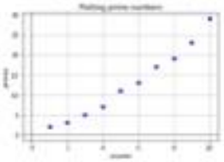
distance 399.609375

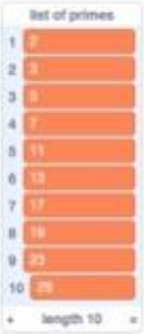


Math: Linear & non-linear relations; "operations" on linear equations; graphical representations of relations

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

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





list of primes

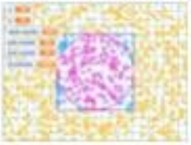
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19
9	23
10	29

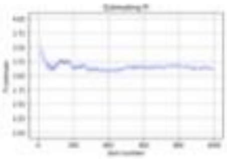
+ length 10 =

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





Math: Meaning of Pi; experimental probability

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

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