


Computer programming and problem solving: A winning combination!



February 2020

Direction de la formation générale des jeunes
Ministère de l'Éducation et de l'Enseignement supérieur

Québec 



Goals

- ◉ Develop a better understanding of computer programming as a learning tool and learning objective.
- ◉ Make connections between the creative process used in computer programming and the problem-solving process used in mathematics.



Presentation outline

1. Ministerial documents
2. Computer programming in schools
3. Programming and the Mathematics program
4. Examples of activities involving programming



Participants

*Elementary school
teachers*

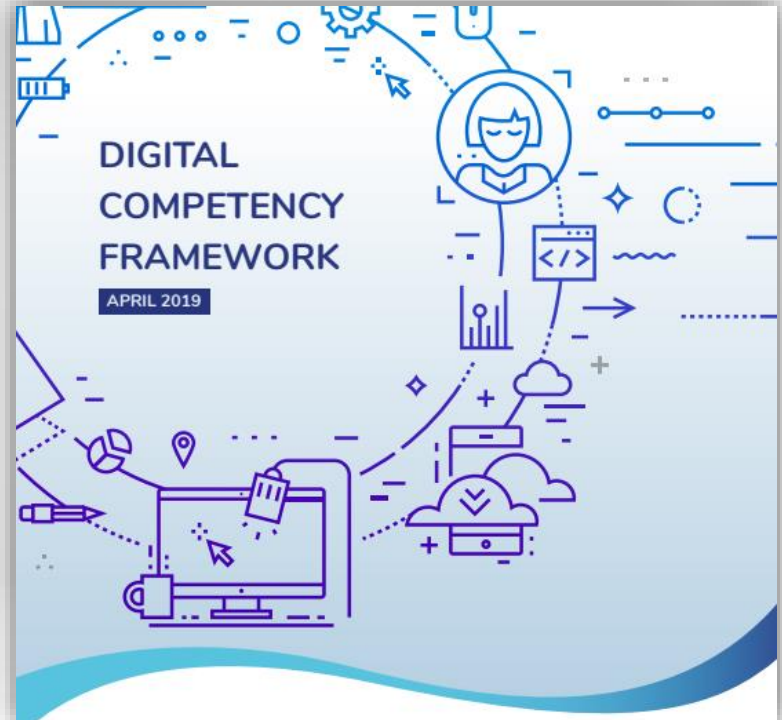
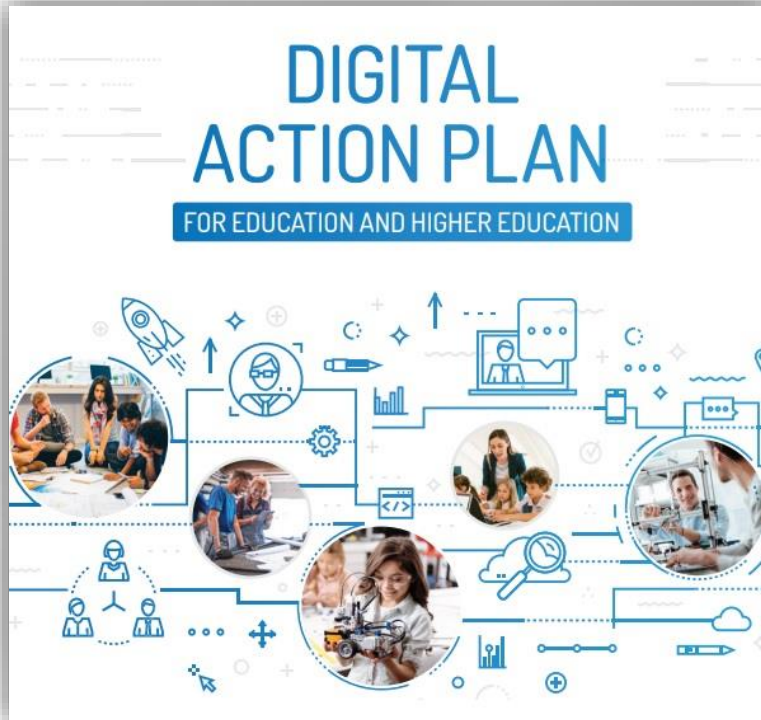
*Education consultants
in mathematics*

*Education consultants
in ICT integration*

Others

1

Ministerial documents





The Digital Action Plan

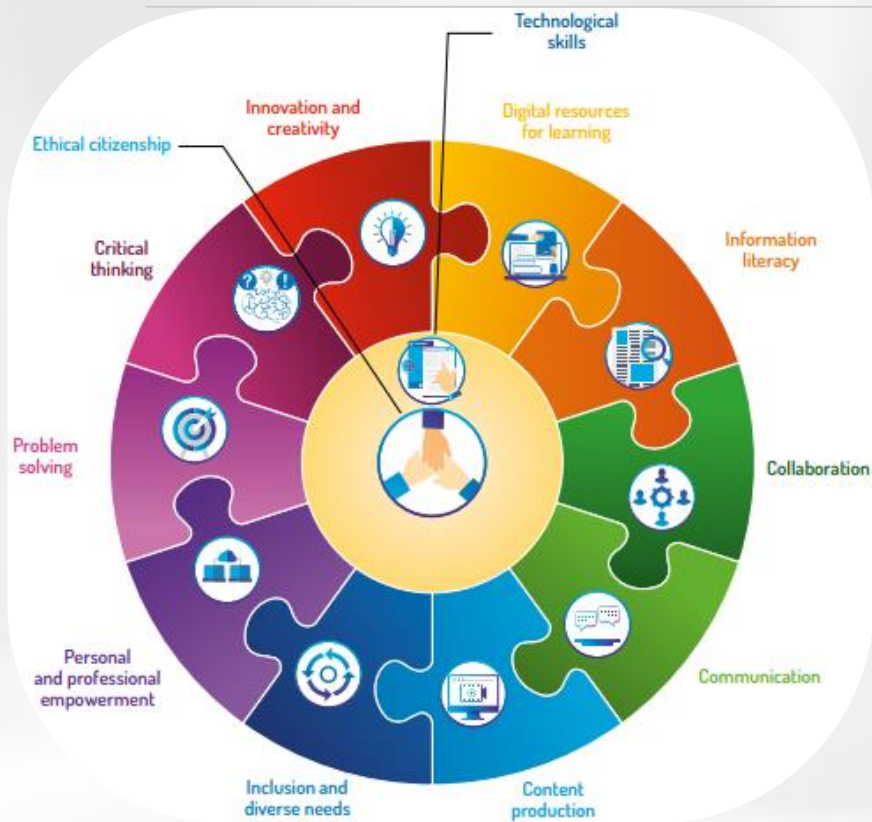
MEASURE

02 INCREASE THE USE
OF CODING IN EDUCATION

“The Ministère will encourage and support the use of coding for educational and didactic purposes in order to help students acquire the learning and competencies prescribed in the Québec Education Program (QEP).”



The Digital Competency Framework





2 DEVELOPING AND MOBILIZING TECHNOLOGICAL SKILLS

ELEMENTS:

- Developing a general understanding of artificial intelligence and its impact on education, society, culture and politics
- Developing greater awareness of emerging issues concerning digital technology and its impact on education
- Learning to use new technologies to keep one's digital skills up to date
- Developing computational thinking, particularly through the improvement of one's programming skills and enhancement of one's understanding of computer programming
- Securing personal data with the appropriate resources, taking into account the risks associated with the use of digital technology
- Mobilizing the technological skills required to use different software, digital platforms and applications in educational activities and everyday life
- Exploring how everyday devices function on a mechanical, electronic and computer level
- Implementing an appropriate solution or seeking help to solve a technological problem

2

Computer programming in schools

THE
FACTS ON
EDUCATION

12 reasons to learn coding at school

Learning how to code: what are the key benefits for students?

Research shows that teaching computer coding starting in Kindergarten generates many benefits for students. Here are 12 key benefits of learning to code at school:

1. Increased academic motivation
2. Acquisition of mathematical skills
3. Ability to problem solve
4. Acquisition of computer skills
5. Development of autonomy
6. Teamwork, collaboration, and mutual assistance
7. Development of critical thinking
8. Improved self-esteem and sense of competence
9. Development of creativity
10. Ability to find information
11. Increased resilience in the face of challenges
12. Enhanced reasoning, organization, and planning skills

For online resources and references please visit:
www.edcan.ca/facts-on-education

About the Author
Dr. Thierry Karsenti, Full Professor,
Université de Montréal

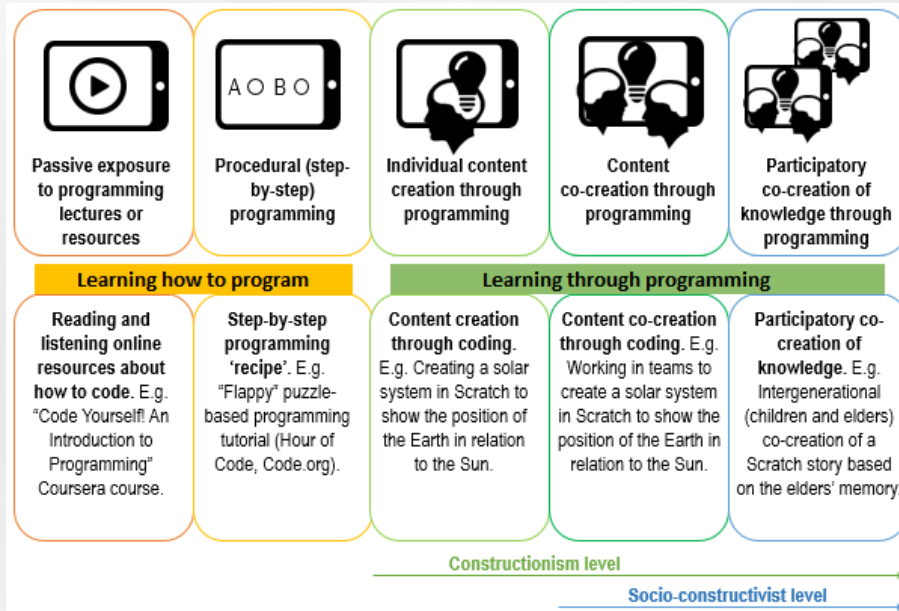


Computer programming in a school setting

“Programming is more than just coding, for, it exposes students to computational thinking which involves problem-solving using computer science concepts like abstraction and decomposition. Even for non-computing majors, computational thinking is applicable and useful in their daily lives.”

Source: Sze Yee Lye & Joyce Hwee Ling Koh, “Review on teaching and learning of computational thinking through programming: What is next for K-12?,” *Computers in Human Behavior*, vol. 41 (2014), 51-61.

Learning how to program or learning through programming



Margarida Romero "De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative," *Formation et profession*, vol. 24, no. 1 (2016), 87-89, <http://dx.doi.org/10.18162/fp.2016.a92>.

Image source:

https://www.researchgate.net/publication/305699926_Learning_to_code_from_procedural_puzzle-based_games_to_creative_programming



Computer programming: Learning tool or learning objective?

Learning tool

- ◉ May be used in **all subject areas**
- ◉ **Simple** to use, with a **short training phase**

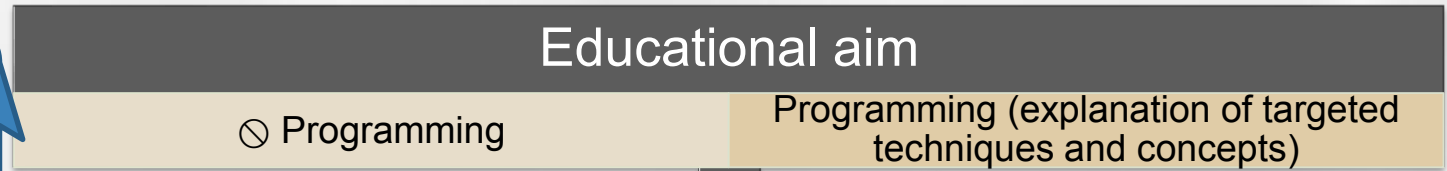
Learning objective

- ◉ More commonly used in **Mathematics, Science and Technology** and **Computer Science** (as well as Arts Education)
- ◉ More **complex** to use, with a **long training phase** (e.g. conditional constructs, types of variables, repetition structures, object-oriented programming)

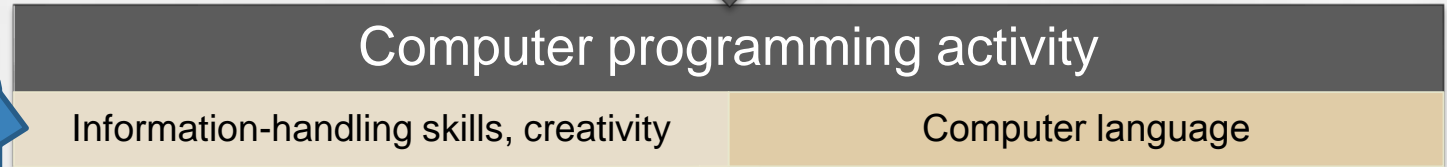


Computer programming: Learning tool or learning objective? (cont.)

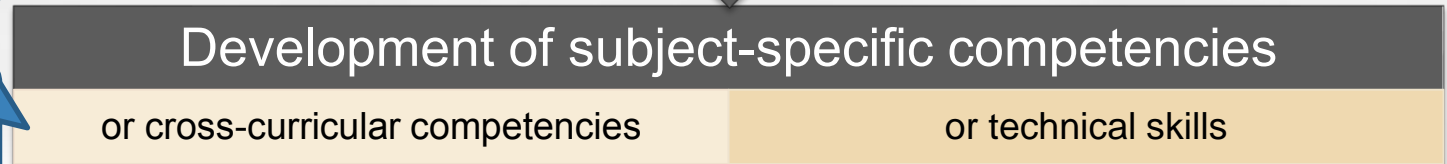
The educational aim
may be learning a
mathematical concept
or process.



How students learn the
mathematical concept
or process.



Students develop
mathematical
competencies.



Initial strategies

Start with activities
that require little
material and
equipment

Encourage
participation in
non-digital
activities

Observe a
programming
activity in a
colleague's class

Participate in
training sessions
on the topic

Establish a group
of student experts



3

Computer programming and the Mathematics program

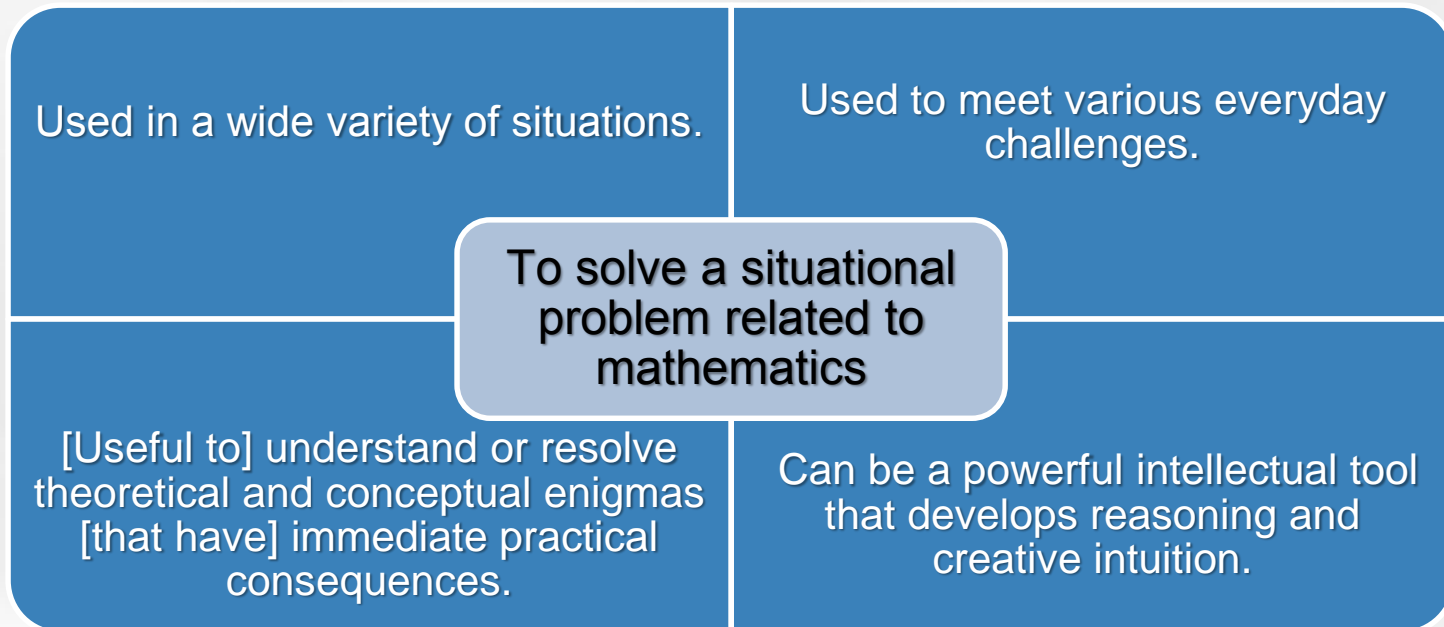
“High technology, engineering and computer programming are among the many fields requiring the use of mathematics, but it is also used in manufacturing common everyday objects, in measuring time or in organizing space.”

QEP, Elementary, p. 140.

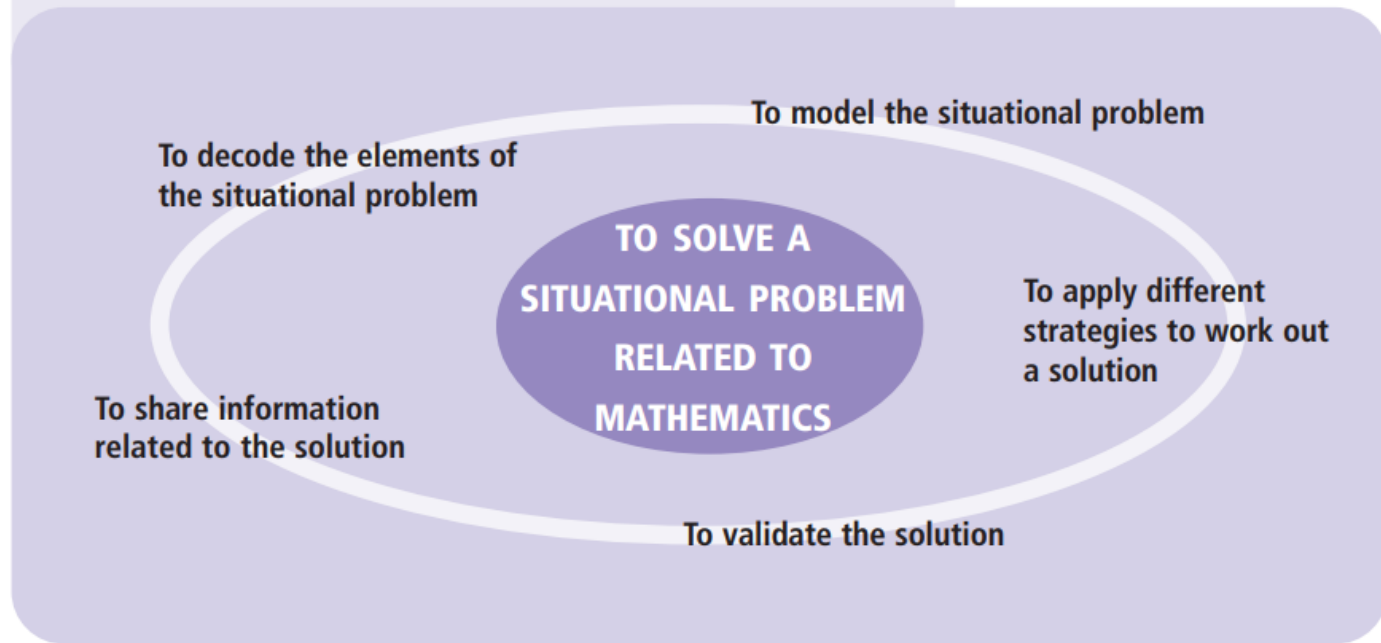




Connections with problem solving in mathematics

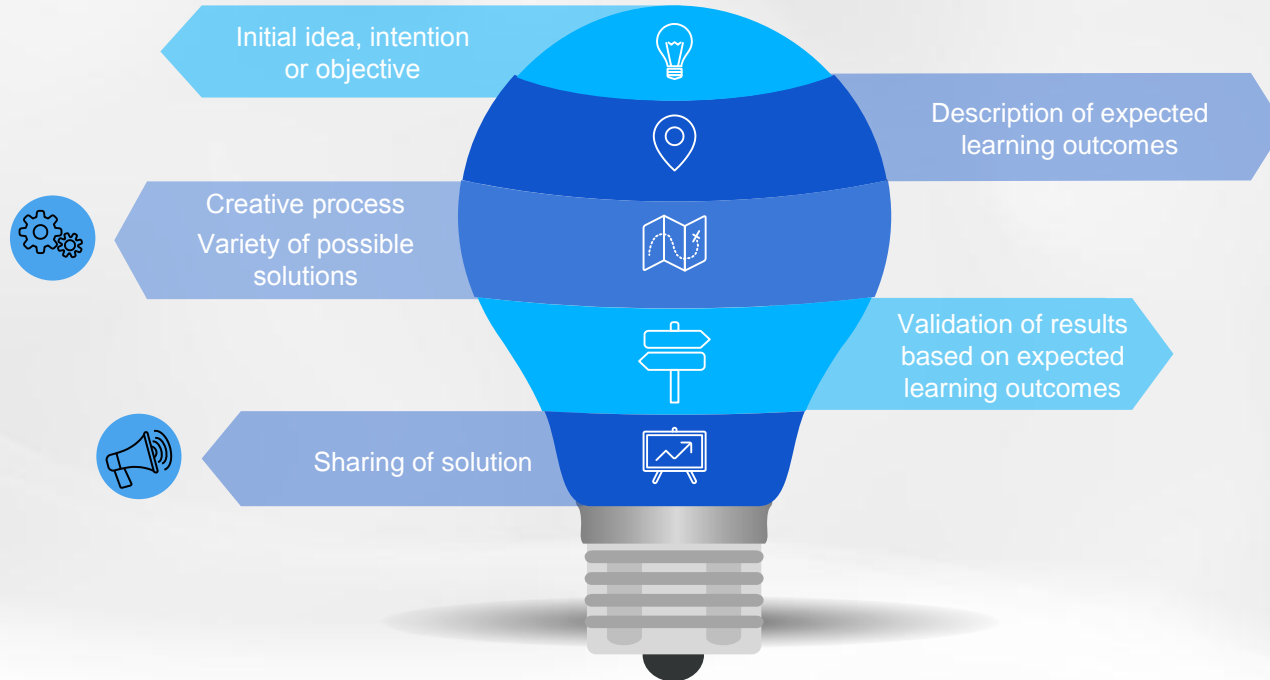


Key Features of the Competency

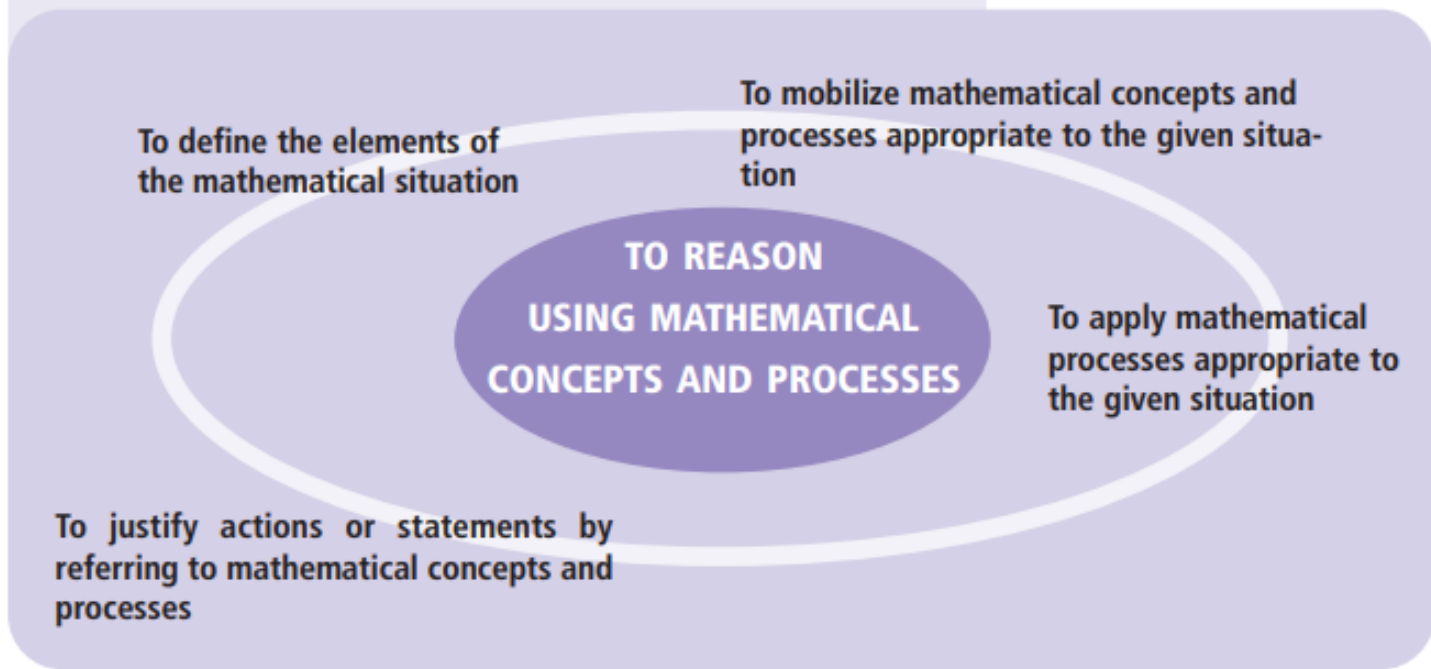




Computer programming and the problem-solving process



Key Features of the Competency



Key Features of the Competency

To become familiar with mathematical vocabulary

**TO COMMUNICATE BY
USING MATHEMATICAL
LANGUAGE**

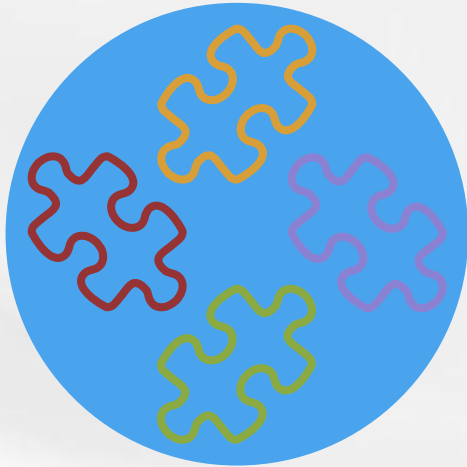
To make connections
between mathematical
language and everyday
language

To interpret or produce mathematical
messages



4

Examples of activities involving programming



- ⦿ using a spreadsheet
- ⦿ using dynamic geometry software
- ⦿ using presentation software
- ⦿ using a coding application

How to code patterns

Pattern generator		Additive patterns					
Enter a starting number:	<input type="text" value="0"/>	3	6	9	12	15	18
Pattern:	<input type="text" value="3"/>						
Pattern generator		Subtractive patterns					
Enter a starting number:	<input type="text" value="100"/>	96	92	88	84	80	76
Pattern:	<input type="text" value="3"/>						
Pattern generator		Multiplicative patterns					
Enter a starting number:	<input type="text" value="1"/>	4	16	64	256	1024	4096
Pattern:	<input type="text" value="4"/>						

Generating arithmetic sequences using a spreadsheet

How do I program my spreadsheet to generate an arithmetic sequence that follows a specific pattern?

Examples of formulas that generate arithmetic sequences

[illegible]



Mathematical concepts and processes involved in this task

Arithmetic

- Using [their] own words and mathematical language . . . describes numerical patterns
- Using [their] own words and mathematical language . . . describes series of numbers and family of operations
- Adds new terms to a series



Name	Result #1	Result #2	Result #3	Mean
Annie	8	10	6	8.0
Julie	6	7	7	6.7
Geneviève	9	7	6	7.3
Martin	9	7	6	7.3
Raymond	7	8	8	7.7
Sophie	8	7	10	8.3
Mean	7.8	7.7	7.2	

=AVERAGE(C3:E3)

=RANDBETWEEN(6;10)

Calculating the arithmetic mean

How do I program my spreadsheet to randomly generate results and calculate the mean for each student?

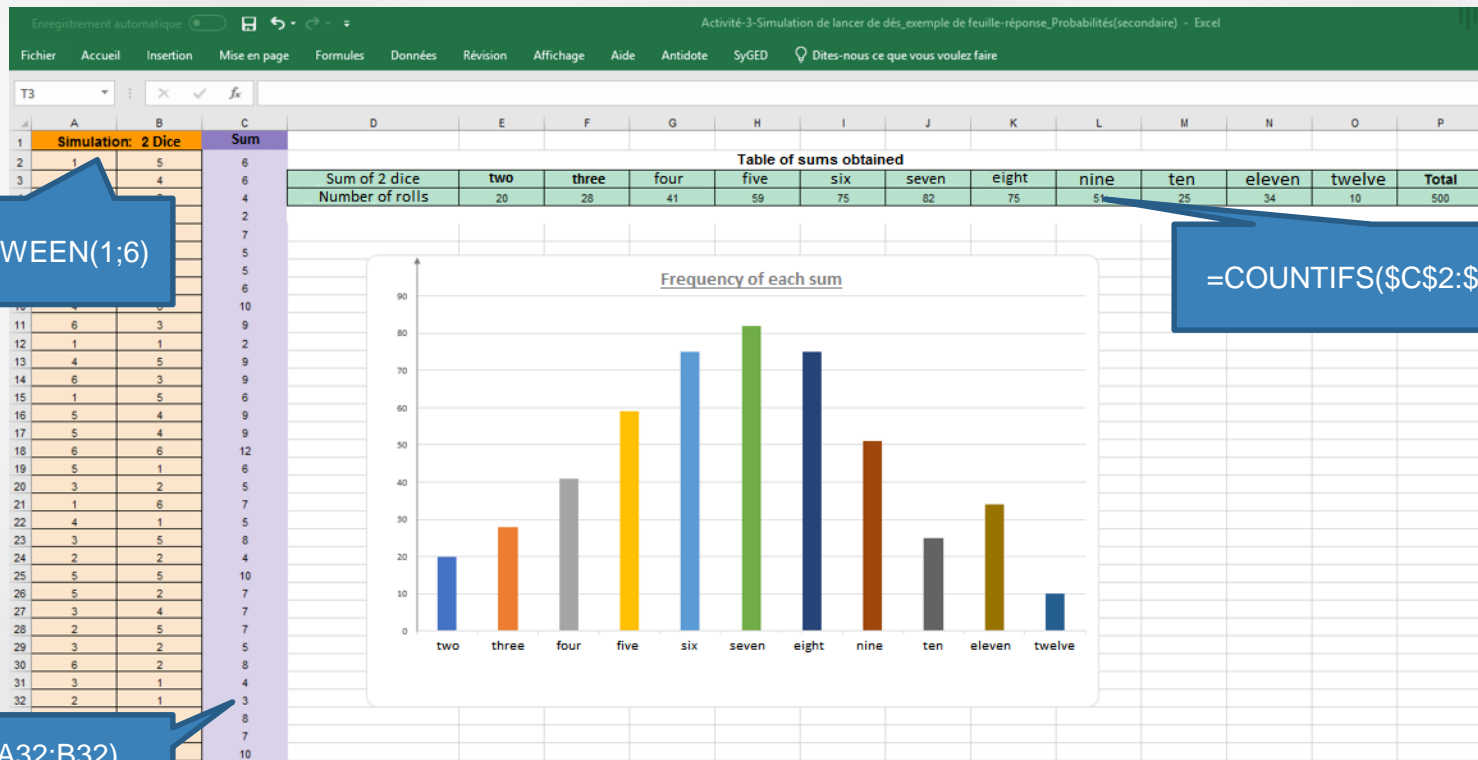


Mathematical concepts and processes involved in this task

Statistics

- ◉ Collects, describes and organizes data (classifies or categorizes) using tables
- ◉ Displays data using a table, a bar graph
- ◉ Understands and calculates the arithmetic mean





Probability and diagrams

How do I program my spreadsheet to randomly generate dice rolls, calculate the sum of two dice and display the frequency of each sum?



Mathematical concepts and processes involved in this task

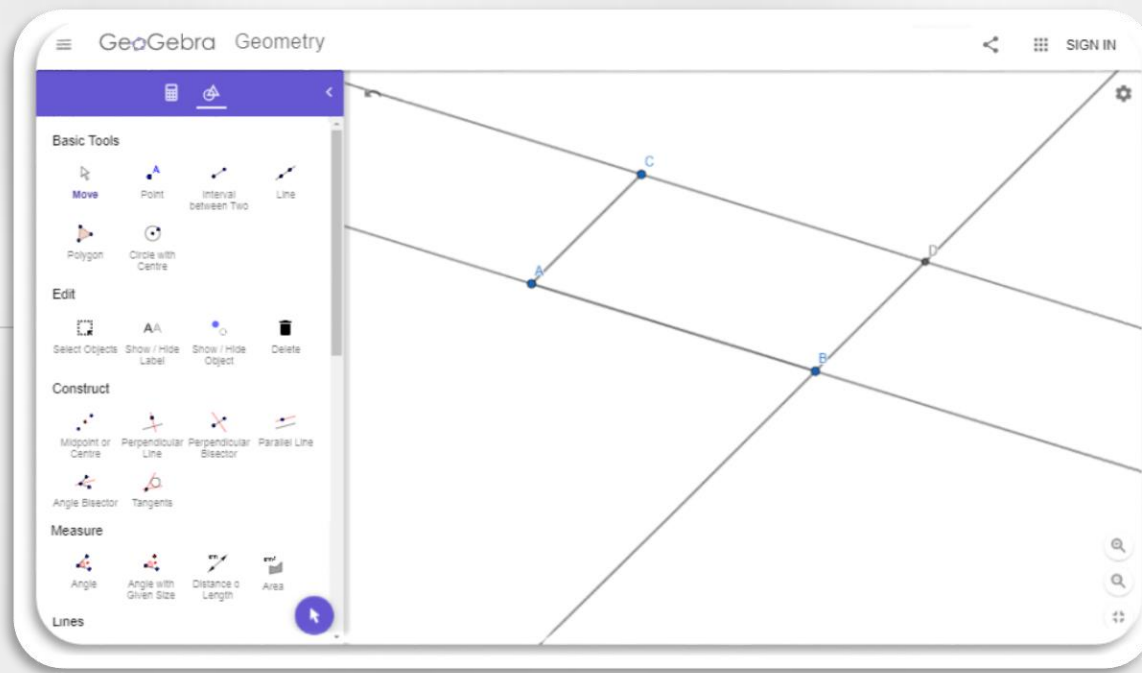
Probability

- Experiments with activities involving chance, using various objects
- Uses tables or diagrams to collect and display the outcomes of an experiment
- Compares the outcomes of a random experiment with known theoretical probabilities

Statistics

- Collects, describes and organizes data (classifies or categorizes) using tables
- Displays data using a table, a bar graph





Drawing plane figures using GeoGebra's Geometry application

How do I draw a figure using [GeoGebra](#)'s Geometry application?



Mathematical concepts and processes involved in this task

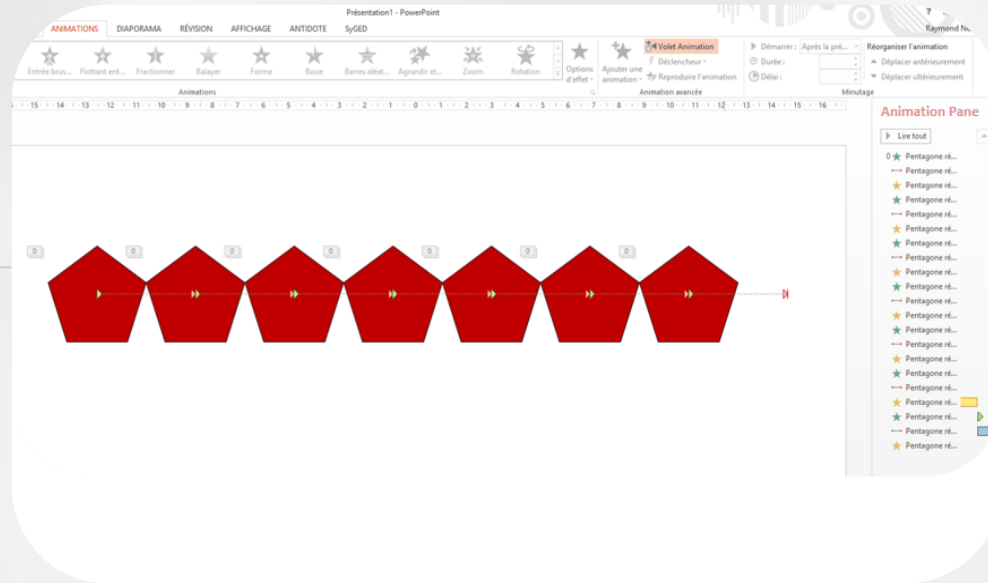
Geometry

- **Plane figures**
 - Constructs figures made with closed curved lines or closed straight lines
 - Constructs parallel lines and perpendicular lines
 - Describes quadrilaterals

Measurement

- **Lengths**
 - Estimates and measures using unconventional units
- **Angles**
 - Estimates and determines the degree measurement of angles



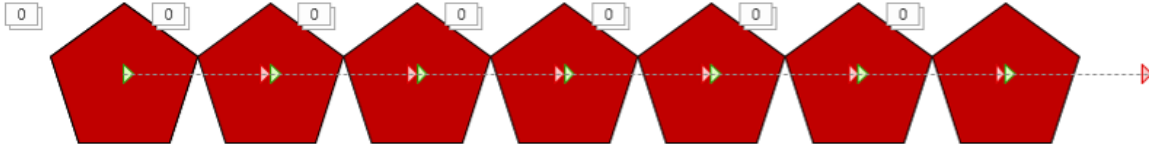


Creating frieze patterns and tessellations using presentation software

How do I program my presentation software, using animations, to create a frieze pattern or a tessellation using translations and reflections?

Examples of student processes

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



Animation P...

Play From

0 ★ Pentagon...

★ Pentagon...

— Pentagon...

★ Pentagon...

★ Pentagon...

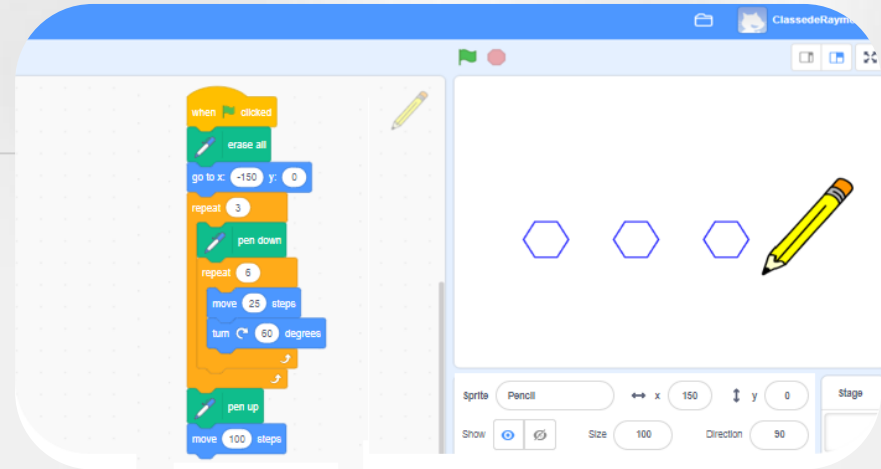
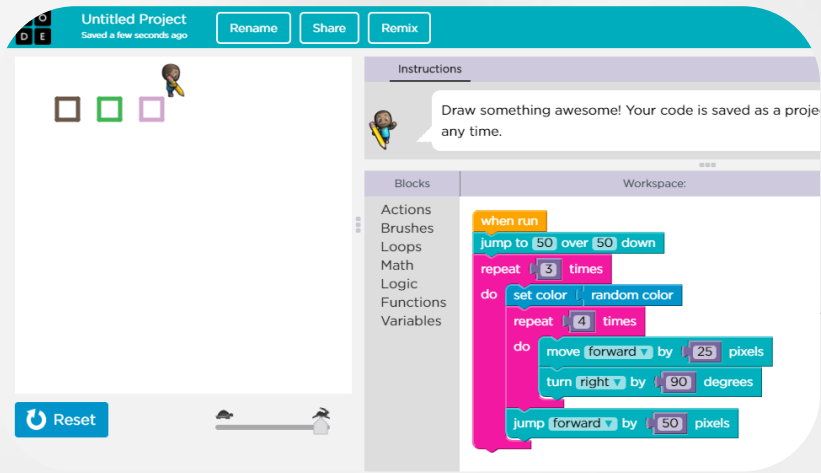
— Pentagon...

★ Pentagon...

★ Pentagon...

— Pentagon...

★ Pentagon...



Creating frieze patterns and tessellations using a coding application

How do I program a coding application to draw a frieze pattern using translations and reflections?



Mathematical concepts and processes involved in this task

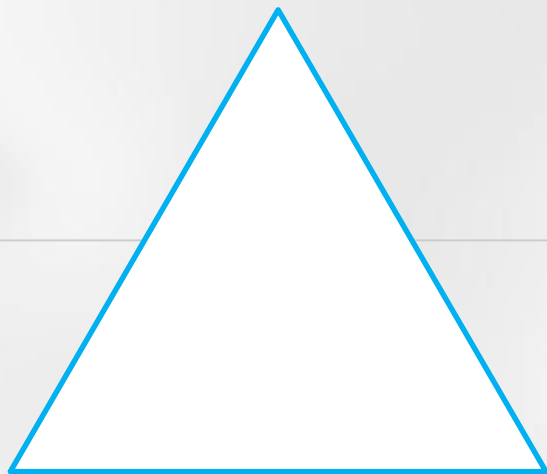
Geometry

- Observes and produces patterns using geometric figures
- Observes and produces frieze patterns and tessellations
 - using reflections
 - using translations

Measurement

- Estimates and measures time using conventional units

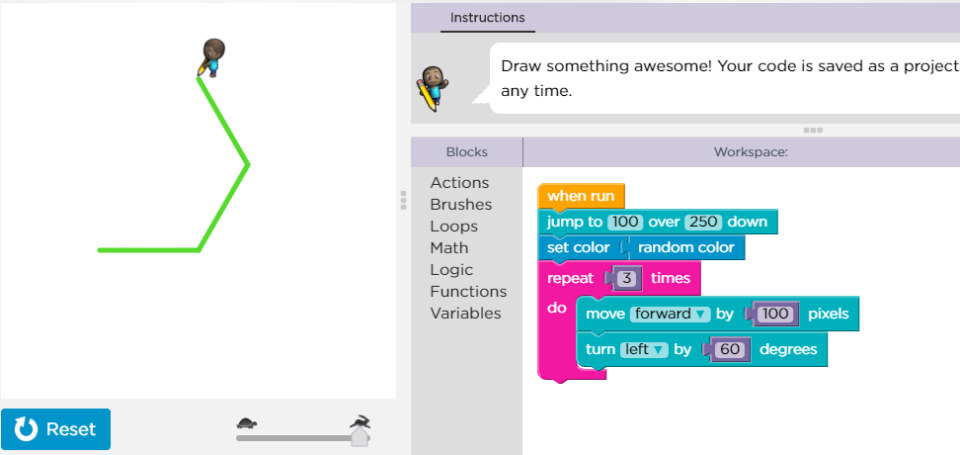




Drawing plane figures using a coding application

How do I program a coding application
to draw an equilateral triangle?

Examples of student processes



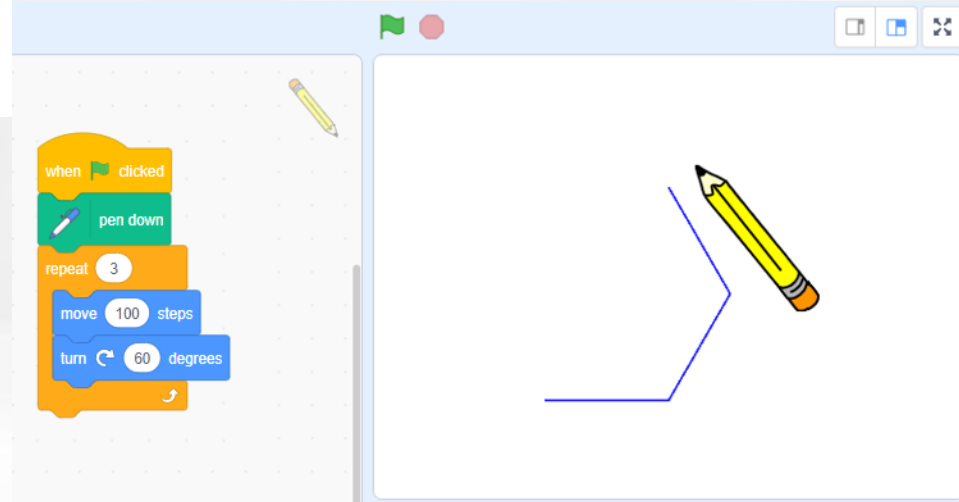
The image shows a Scratch workspace with a green line drawing of a stylized 'Z' shape. The drawing is composed of three segments: a horizontal line to the right, a vertical line down, and a horizontal line to the left. A small Scratch character is positioned at the top of the vertical segment. The interface includes a 'Reset' button at the bottom left and a 'Workspace' area on the right containing code blocks.

Instructions:

Draw something awesome! Your code is saved as a project any time.

Workspace:

```
when run
  jump to 100 over 250 down
  set color random color
  repeat 3 times
    do
      move forward by 100 pixels
      turn left by 60 degrees
```

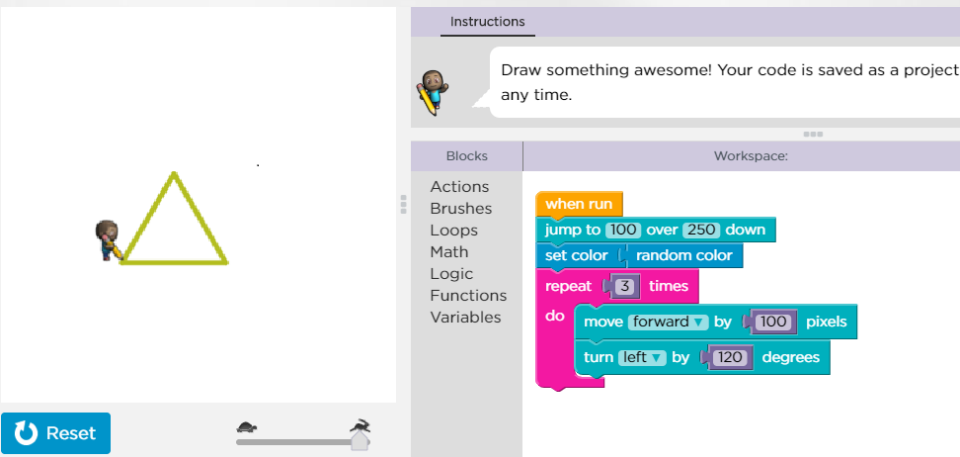


The image shows a Scratch workspace with a blue line drawing of a stylized 'Z' shape. The drawing is composed of three segments: a horizontal line to the right, a vertical line down, and a horizontal line to the left. A small Scratch character is positioned at the top of the vertical segment. The interface includes a 'Reset' button at the bottom left and a 'Workspace' area on the right containing code blocks.

Workspace:

```
when clicked
  pen down
  repeat 3
    move 100 steps
    turn 60 degrees
```

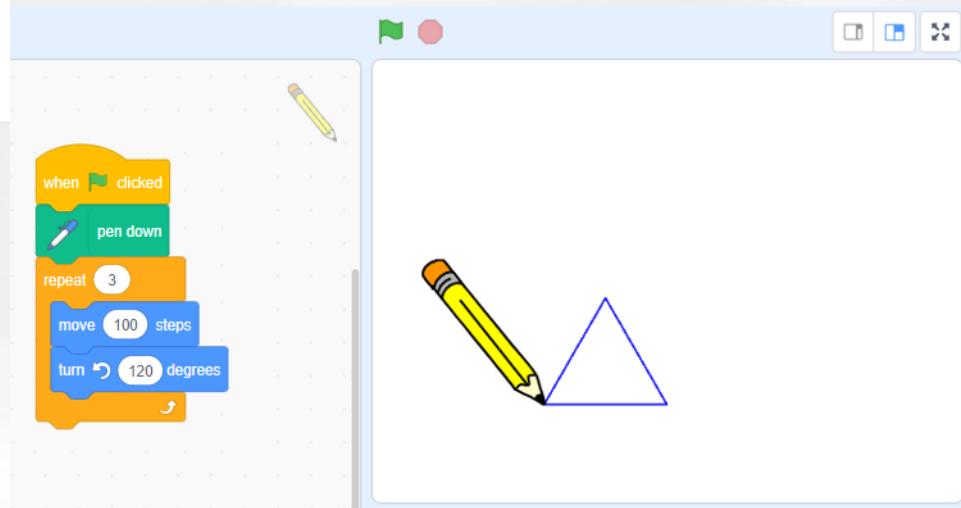
Examples of student processes



The image shows a Scratch workspace with a yellow triangle drawn on the left. A character is standing at the bottom-left vertex of the triangle. On the right, the 'Instructions' panel displays a message: "Draw something awesome! Your code is saved as a project any time." Below this, the 'Workspace' panel shows a code block with the following instructions:

```
when run
  jump to 100 over 250 down
  set color random color
  repeat 3 times
    do
      move forward by 100 pixels
      turn left by 120 degrees
```

At the bottom left, there is a 'Reset' button and a small character icon.



The image shows a Scratch workspace with a blue triangle drawn on the right. A yellow pencil character is standing at the bottom-left vertex of the triangle. On the left, the 'Workspace' panel shows a code block with the following instructions:

```
when clicked
  pen down
  repeat 3
    move 100 steps
    turn 120 degrees
```



Mathematical concepts and processes involved in this task

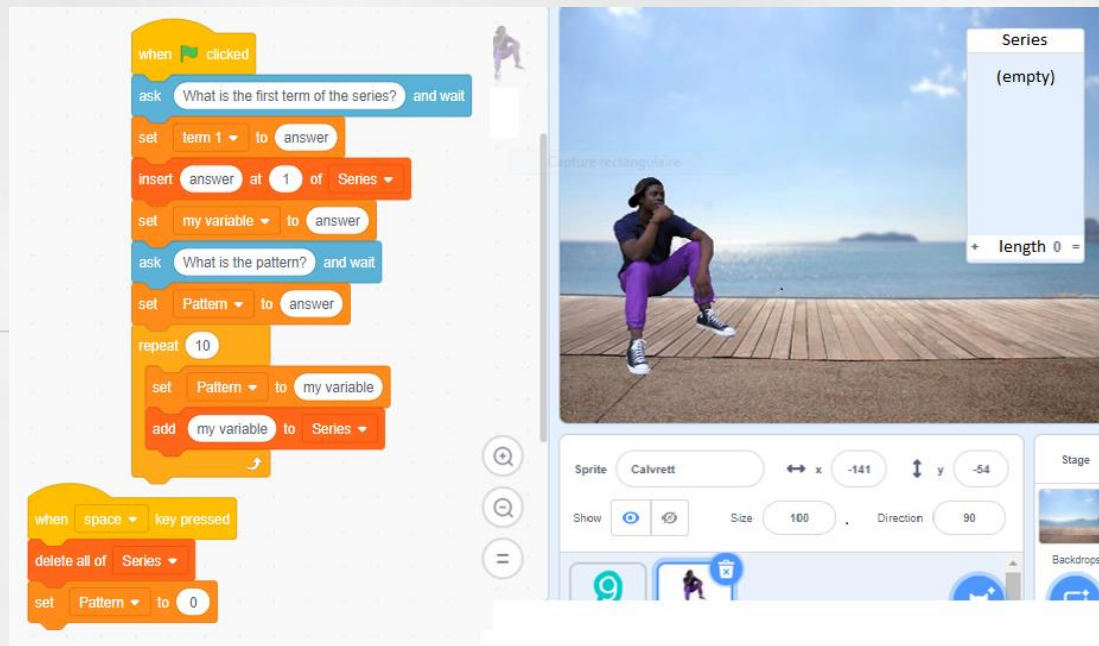
Geometry

- **Plane figures**
 - Constructs figures made with closed curved lines or closed straight lines
 - Describes triangles: scalene triangles, right triangles, isosceles triangles, equilateral triangles

Measurement

- **Lengths**
 - Estimates and measures using unconventional units
- **Angles**
 - Estimates and determines the degree measurement of angles

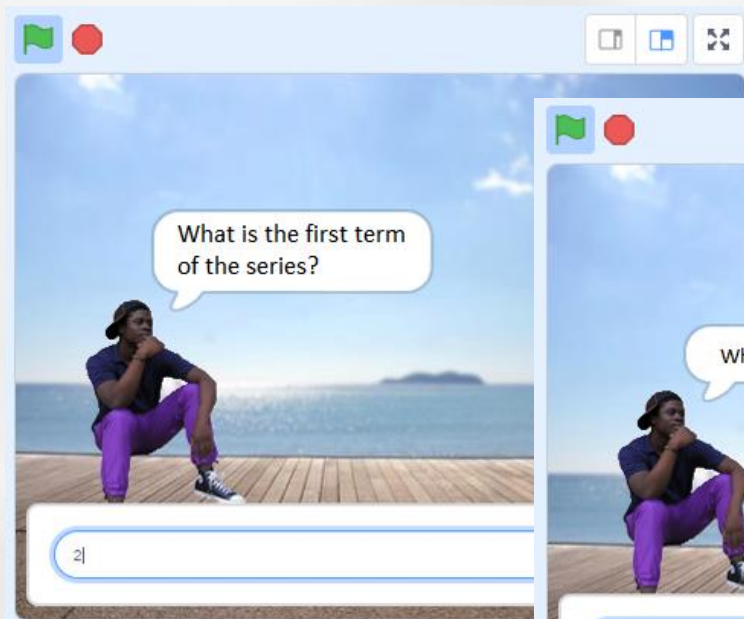




Determine a series of numbers using a coding application

How do I program a coding application to determine a series of numbers?

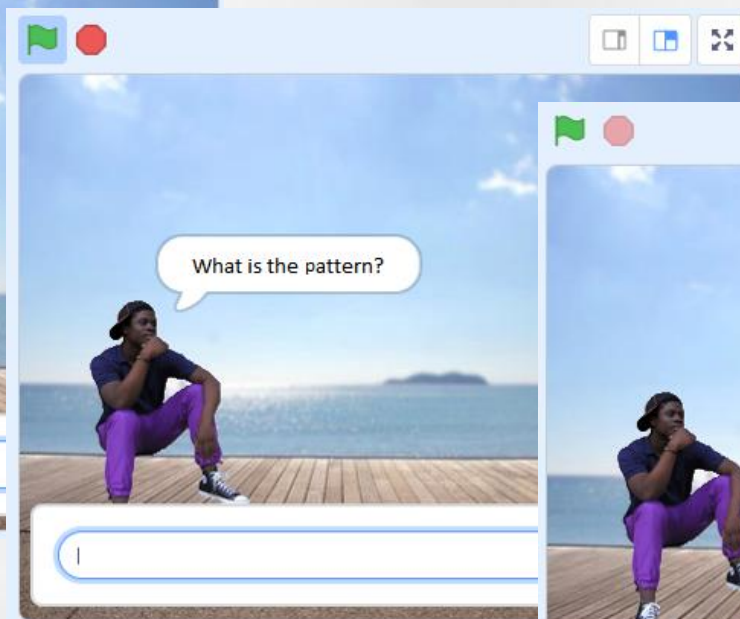
Examples of student processes



What is the first term of the series?

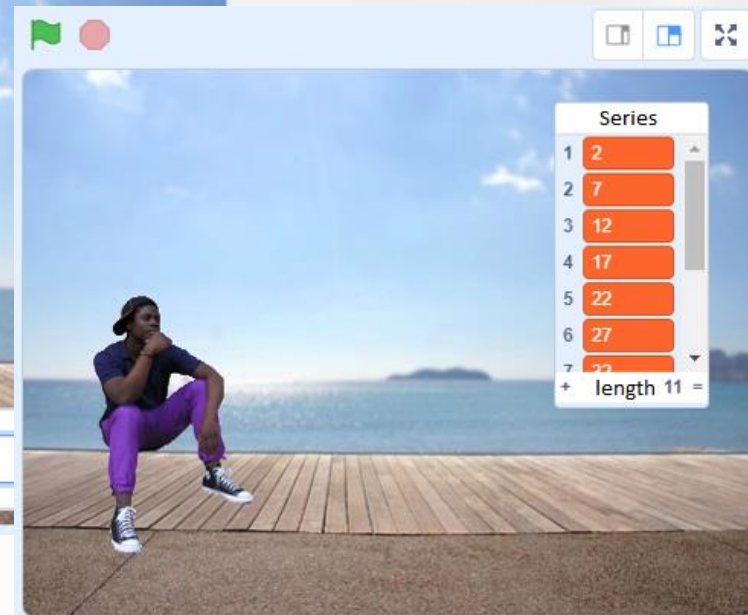
2

This screenshot shows a student interface with a background image of a person sitting on a wooden pier. A speech bubble contains the question "What is the first term of the series?". Below the image is a text input field containing the number "2". The interface includes a top bar with a green flag icon, a red stop icon, and three window control icons (minimize, maximize, close).



What is the pattern?

This screenshot shows a student interface with the same background image and top bar as the first one. A speech bubble contains the question "What is the pattern?". Below the image is an empty text input field.



Series

1	2
2	7
3	12
4	17
5	22
6	27
7	32

+ length 11 =

This screenshot shows a student interface with the same background image and top bar. A table titled "Series" is displayed on the right side of the screen. The table has two columns: the first column contains numbers 1 through 7, and the second column contains numbers 2, 7, 12, 17, 22, 27, and 32. Below the table, there is a text input field containing the text "+ length 11 =".



Mathematical concepts and processes involved in this task

Arithmetic

- Using [their] own words and mathematical language, describes numerical patterns
- Using [their] own words and mathematical language, describes series of numbers and family of operations
- Adds new terms to a series





Other ideas?



Food for thought

My knowledge

How can computer programming help students to develop problem-solving skills in mathematics?

Can computer programming help motivate students to solve mathematical problems?

My experience

Which role should computer programming activities play in mathematics courses?

Am I comfortable enough to tackle computer programming with students?



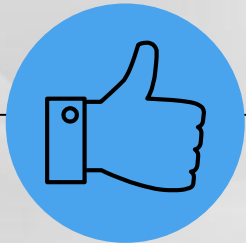
Some references*

- ◉ *Le Site du Domaine de la mathématique*: domaine.recitmst.qc.ca
 - [Networking session of January 16, 2018](#)
- ◉ *Récit MST website*: recitmst.qc.ca
 - [Presentation – Initiation à Scratch en mathématique \(Introduction to Scratch in mathematics\)](#)
 - [Spreadsheet – Planification globale : Leçons de programmation mathématique \(Overall planning: Mathematical Programming Lessons\)](#)
- ◉ *National Récit website*: recit.qc.ca
 - [Campus récit, Premiers pas avec scratch en mathématique](#)
- ◉ *Personal website of Professor Jean-François Maheux*: jfmaheux.net
 - [Some notes on three programming environments](#)

Thank You!

*If you have any questions,
please contact us!*

FGJ-math@education.gouv.qc.ca



Éducation
et Enseignement
supérieur

Québec 