

Grade Six Computer Science

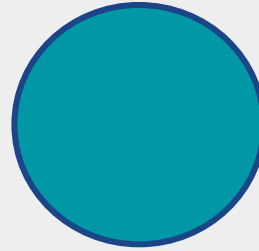


IF you teach THEN they will learn

2013 Over the next 10 years...

Electrical and Computer Engineering

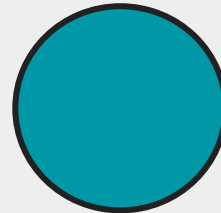
The most promising and
profitable jobs of now and the
future.



There will be
1.4 million
Programming jobs to



With only
400 000
Graduates in
computer science



Leaving
1 million
empty jobs!

Computing jobs are the #1 source of new wages in the US

500,000
current openings

These jobs are in every industry and every state, and they're projected to grow at twice the rate of all other jobs.



There are technology jobs in every field:

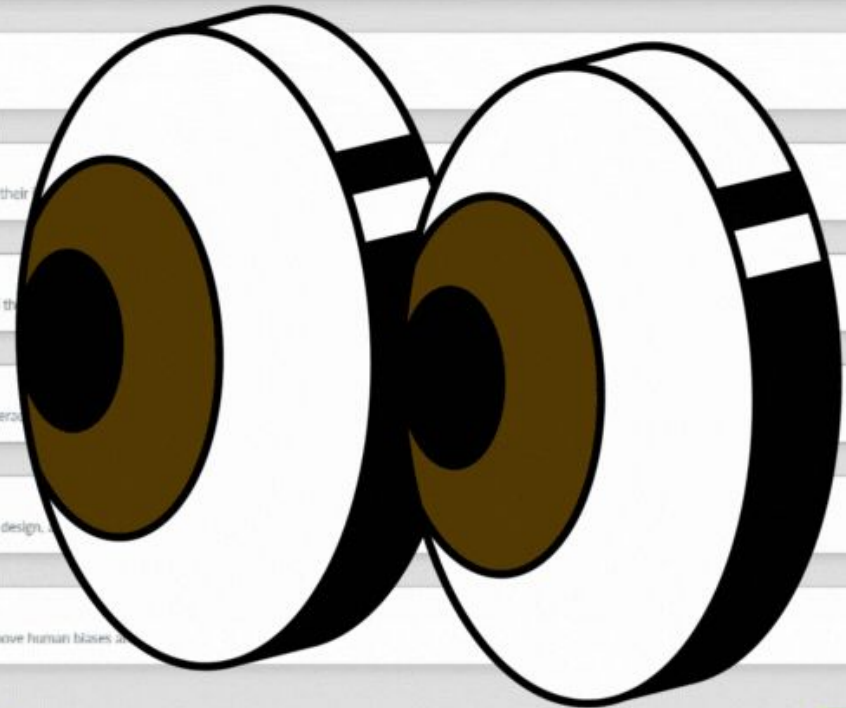


- Home
- Alberta's K-6 Curriculum
- Explore Resources
- Curriculum Implementation Information Hub
- Student Learning Hub
- Printable Curriculum
- Support
- e-Tutoring Hub
- Contact Us

Science Change Subject

Prev Grade 4 Grade 5 Grade 6 Next

- ORGANIZING IDEA
Matter: Understandings of the physical world are deepened through investigating matter and energy.
- ORGANIZING IDEA
Energy: Understandings of the physical world are deepened through investigating matter and energy.
- ORGANIZING IDEA
Earth Systems: Understandings of the living world, Earth, and space are deepened through investigating natural systems and their interactions.
- ORGANIZING IDEA
Living Systems: Understandings of the living world, Earth, and space are deepened through investigating natural systems and their interactions.
- ORGANIZING IDEA
Space: Understandings of the living world, Earth, and space are deepened through investigating natural systems and their interactions.
- ORGANIZING IDEA
Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.
- ORGANIZING IDEA
Scientific Methods: Investigation of the physical world is enhanced through the use of scientific methods that attempt to remove human biases and errors.



Organizing Idea	Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.	
Guiding Question	In what ways are abstraction, design, and coding related?	
Learning Outcome	Students examine abstraction in relation to design and coding, and describe impacts of technologies.	
Knowledge	Understanding	Skills & Procedures
<p>The process of abstraction includes</p> <ul style="list-style-type: none"> • determining what details to keep and what to ignore • removing unnecessary details • identifying important information • generalizing patterns <p>Information is data that is organized to be more useful.</p> <p>An abstraction is a simplified version of something complex.</p> <p>Abstractions can make daily life easier; e.g.,</p> <ul style="list-style-type: none"> • simple controls on appliances • light switches • steering wheels • apps <p>Computational artifacts can be designed to address societal needs and wants; e.g.,</p> <ul style="list-style-type: none"> • weather modelling • communications • automotive controls • medical research • apps 	<p>Abstraction is used in design and coding of computational artifacts to make problems easier to think about.</p>	<p>Apply abstraction during the design process.</p> <p>Identify examples of abstractions encountered in daily life.</p> <p>Discuss the role of design and coding in society.</p> <p>Use a visual block-based language to design code that includes relevant design structures.</p>

Organizing Idea	Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.
Guiding Question	In what ways are abstraction, design, and coding related?
Learning Outcome	Students examine abstraction in relation to design and coding, and describe impacts of technologies.

Knowledge	Understanding	Skills & Procedures
<p>Structures used in coding include</p> <ul style="list-style-type: none"> • sequences • conditionals (if-then-else statements) • loops <p>Sequence structures are ordered sets of instructions within code.</p> <p>Conditional structures are statements that tell computers to complete different actions based on different situations.</p>	<p>Abstraction is used in design and coding of computational artifacts to make problems easier to think about.</p>	<p>Apply abstraction during the design process.</p> <p>Identify examples of abstractions encountered in daily life.</p> <p>Discuss the role of design and coding in society.</p> <p>Use a visual block-based language to design code that includes relevant design structures.</p>

Organizing Idea	Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.	
Guiding Question	In what ways are abstraction, design, and coding related?	
Learning Outcome	Students examine abstraction in relation to design and coding, and describe impacts of technologies.	
	Knowledge	Understanding
<p>The use of computers, coding, and technology can have impacts that are</p> <ul style="list-style-type: none"> • personal • social • environmental • economic <p>Impacts of computers, coding, or technology may be intentional or unintentional.</p>	<p>Computers, coding, and technology can be used in ways that have positive or negative impacts.</p>	<p>Discuss how computers, coding, or technology have had impacts.</p> <p>Predict possible impacts of computers, coding, or technology.</p>



Creativity

ORIGINALITY

Creative thinking often results in ideas or solutions that are unique and haven't been thought of before. It involves breaking away from established patterns and norms.

RISK TAKING

Creative thinkers are often willing to take risks by exploring unconventional ideas or challenging the status quo.

IMAGINATION

Imagination is at the core of creative thinking. It involves the ability to visualize and conceptualize ideas and scenarios that don't currently exist.

FLEXIBILITY

Creative thinkers are open to exploring various possibilities and are willing to adapt their thinking when confronted with new information or challenges. They can switch between different modes of thinking.

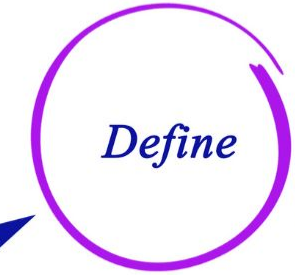
Design Thinking Process



*Learn About
Your Audience*



Empathize

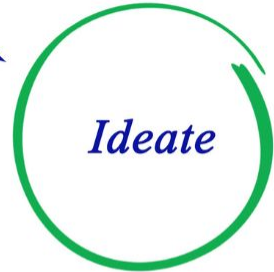


Define

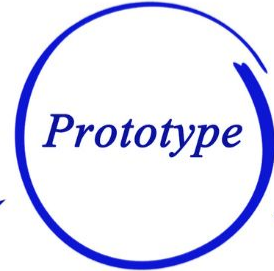
*Construct Point
of View Based
on User Needs*



*Brainstorm and
Come up with
Creative Solutions*

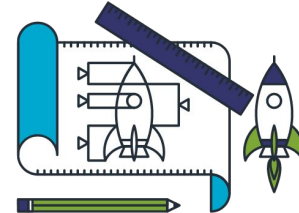


Ideate

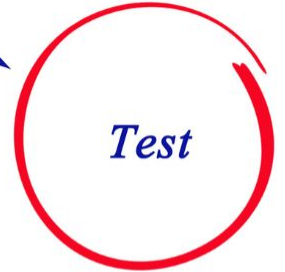


Prototype

*Build
Representation
of Your Ideas*

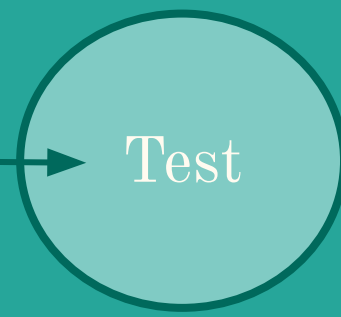
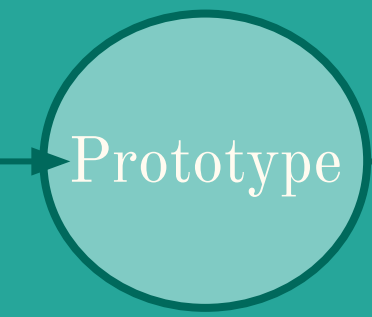
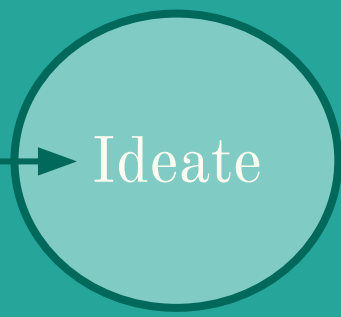
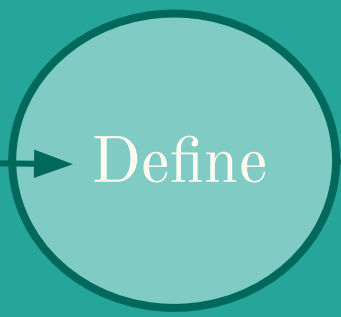


Test Your Ideas



Test





—



Learn about the audience for whom you will be designing.

Walk a mile in their shoes.

Empathize

As you watch the video, start to think about what this person needs. Also be prepared to talk about how the video made you feel.

Think about...

- How you would feel in Mandy's position?
- What kind of person does Mandy seem like?
- What qualities does the video show you about Mandy?



Define

What is Mandy's problem?

Tip: It's not that she is deaf.

This is an impairment that cannot be fixed and Mandy has figure out how to sing even though she cannot hear.

How does Mandy currently deal with her problem?

She sings in her stocking feet so she can feel the vibrations.



Earrings with wireless to sense vibrations



Shoes with no soles or holes to feel



Something to wear under clothes connected to speakers



A chair that picks up the vibrations



Bracelet



Hair accessories



Empathize

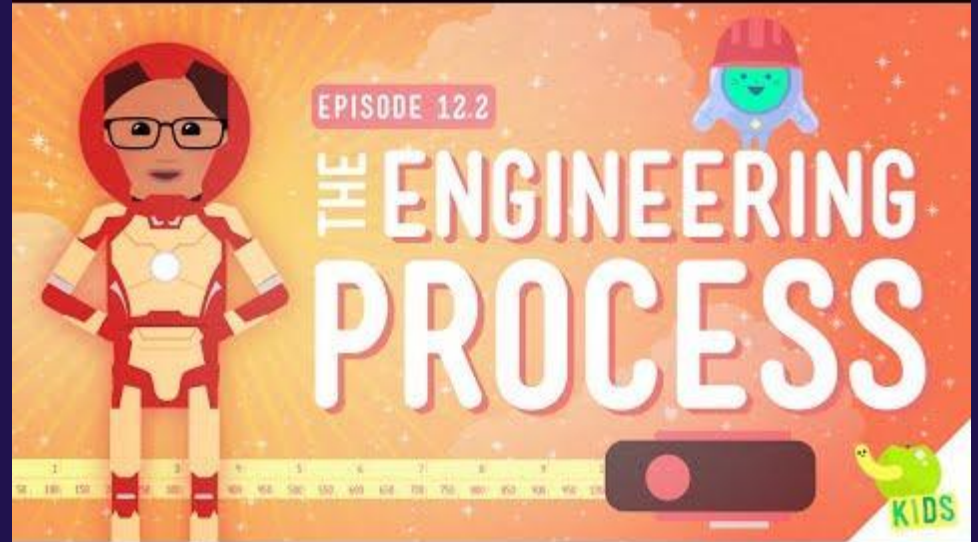
Define

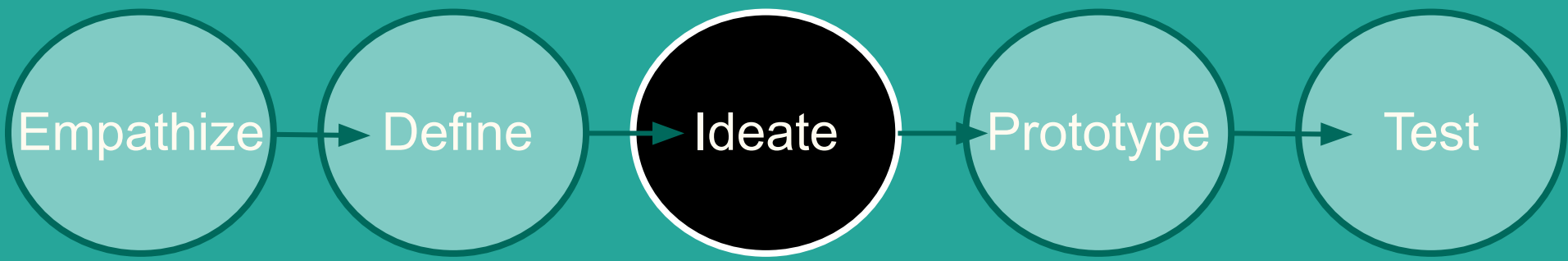
Ideate

Prototype

Test

Use what you know about your audience to determine exactly — what the problem you are going to solve is.





*There are
NO bad
ideas!*

—
*Think about
your
audience's
needs!*



*Quantity
over
Quality!*

*Ideas!
Ideas!
Ideas!*

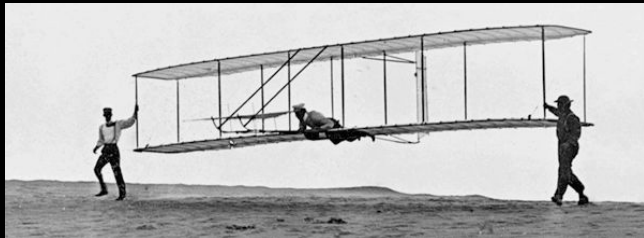
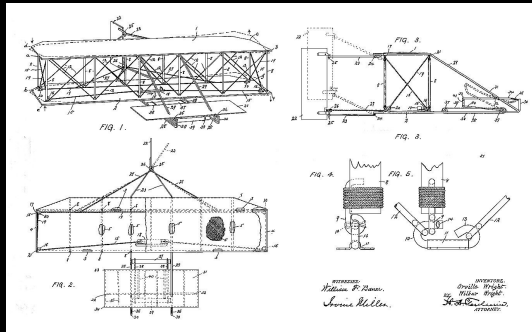
Empathize

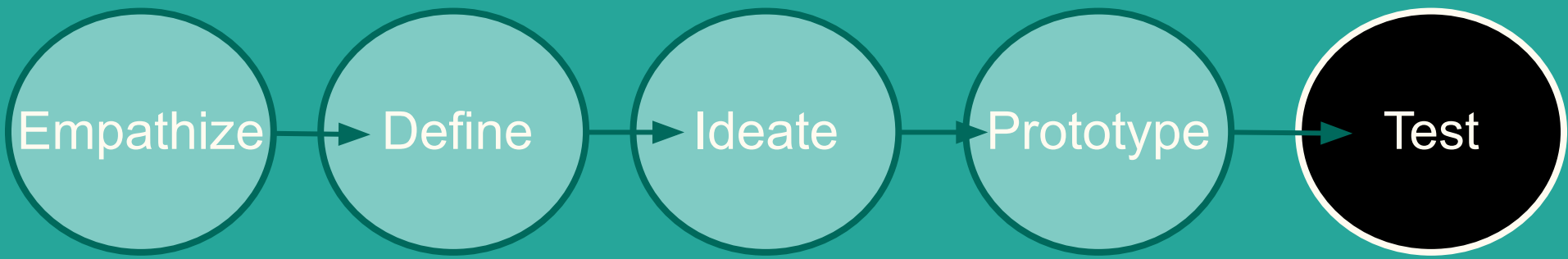
Define

Ideate

Prototype

Test





If at first you don't
succeed,
TRY
TRY
Again...and again...
and again.

**Did you know? The
Wright Brothers had 2
failures before they were
successful with flight.**



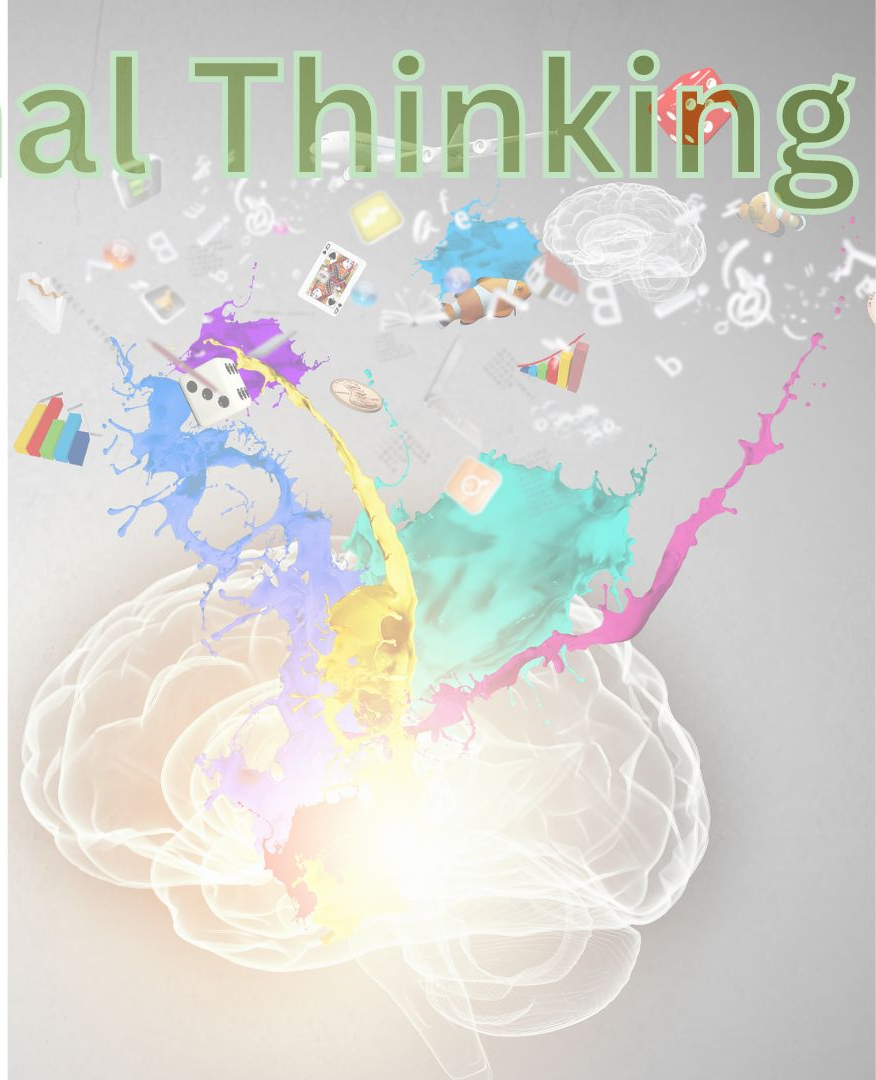
Computational Thinking

Decomposition

Pattern Recognition

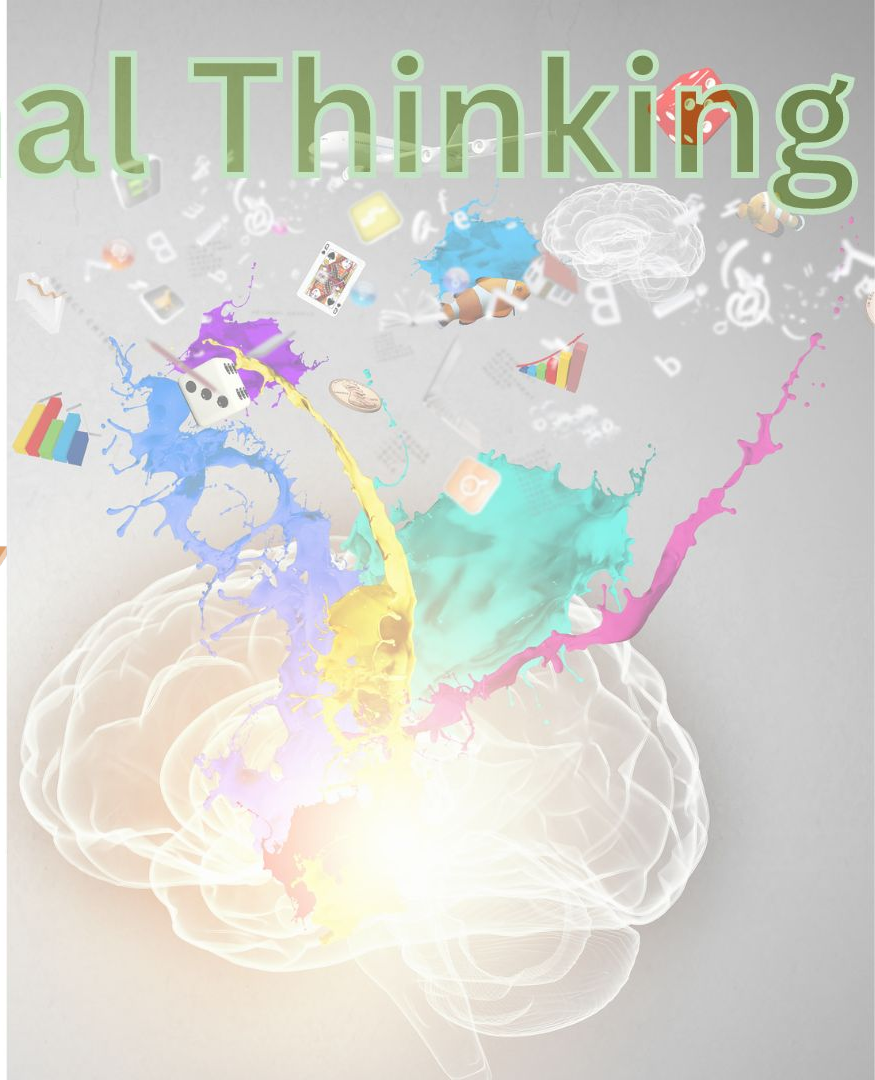
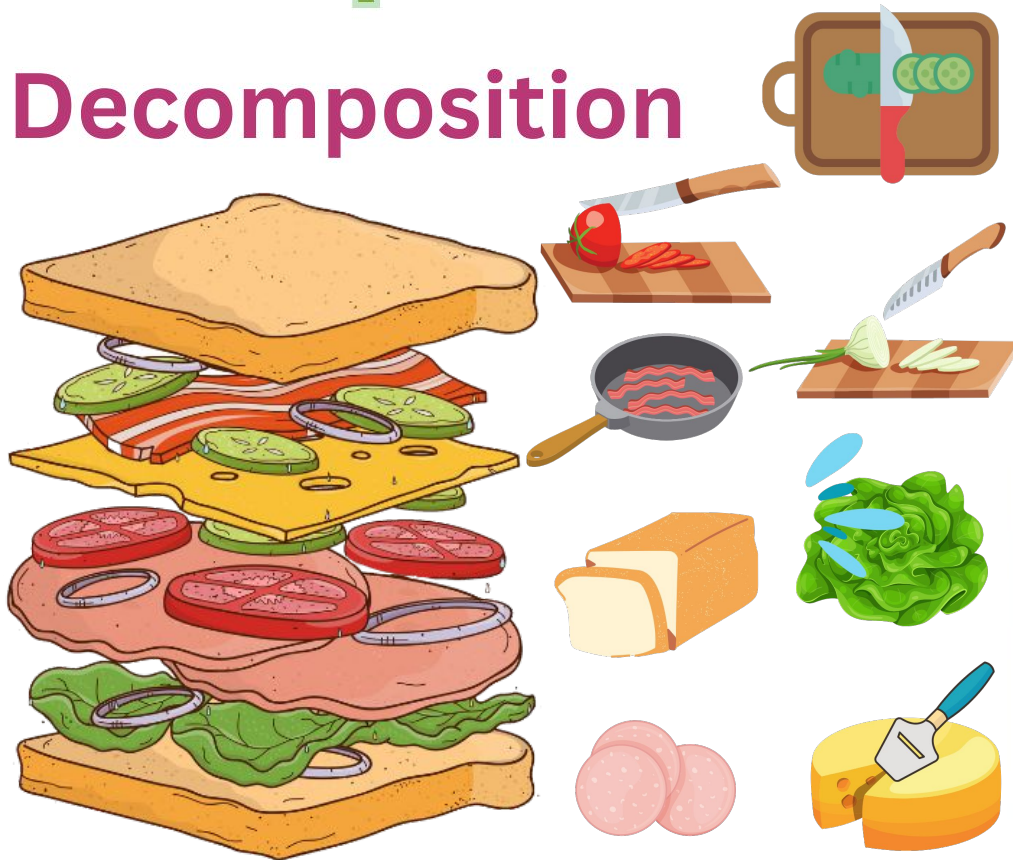
Pattern Abstraction

Algorithm Design



Computational Thinking

Decomposition

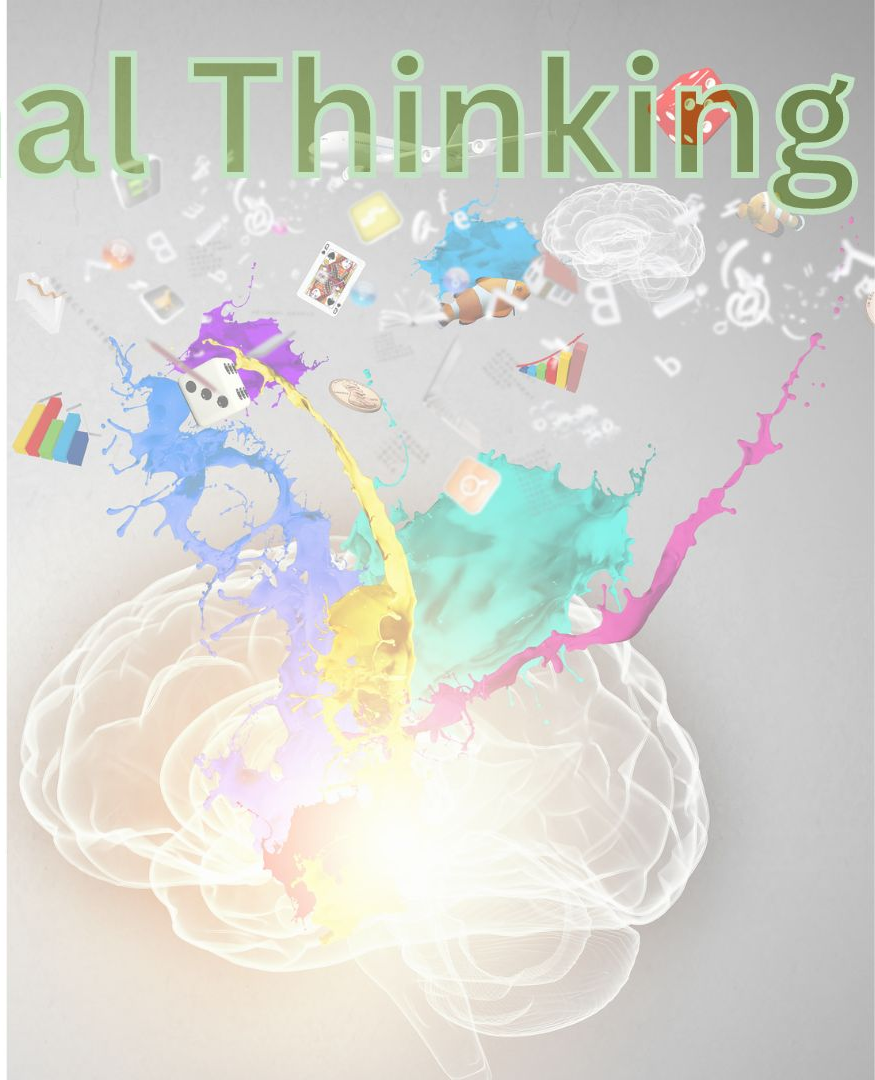
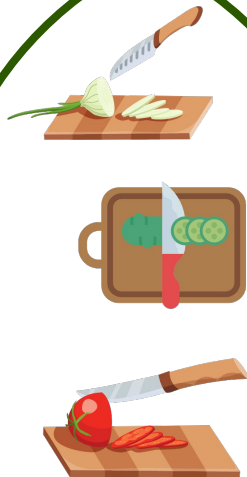
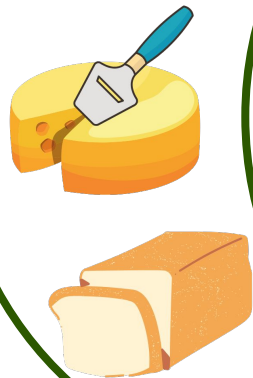


Computational Thinking

Pattern Recognition

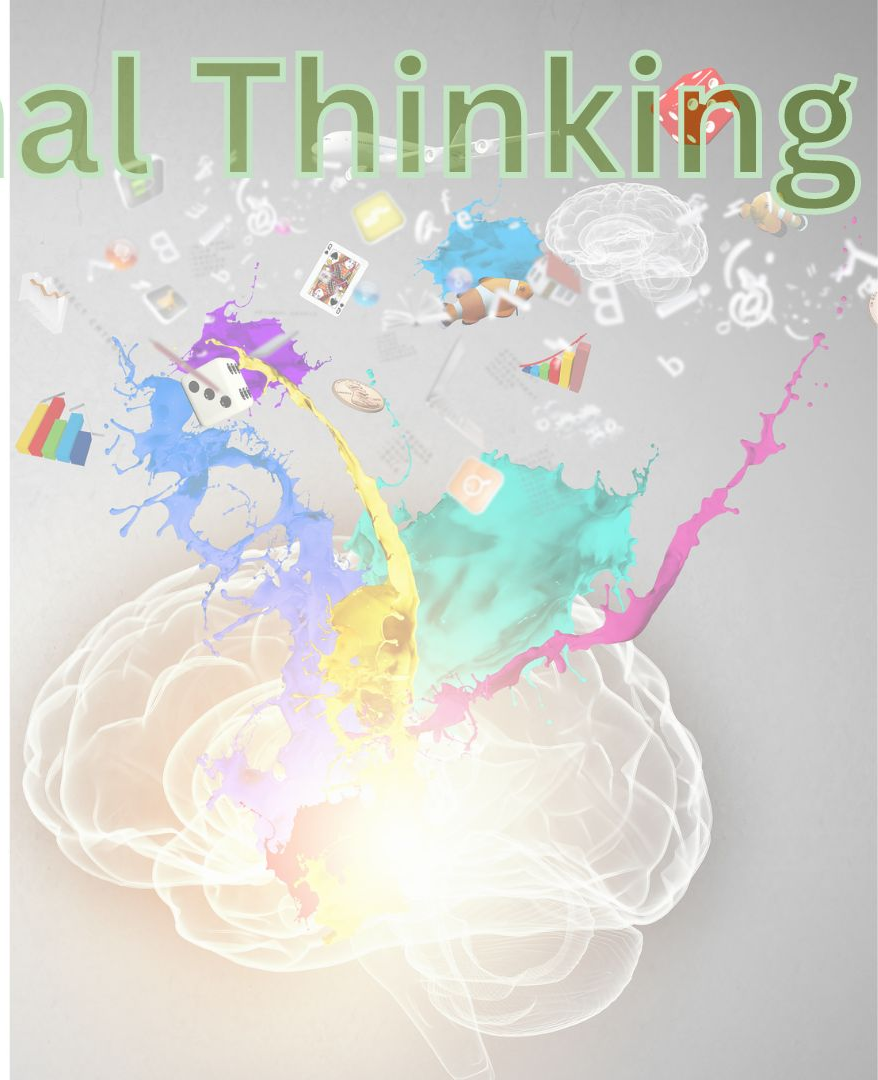
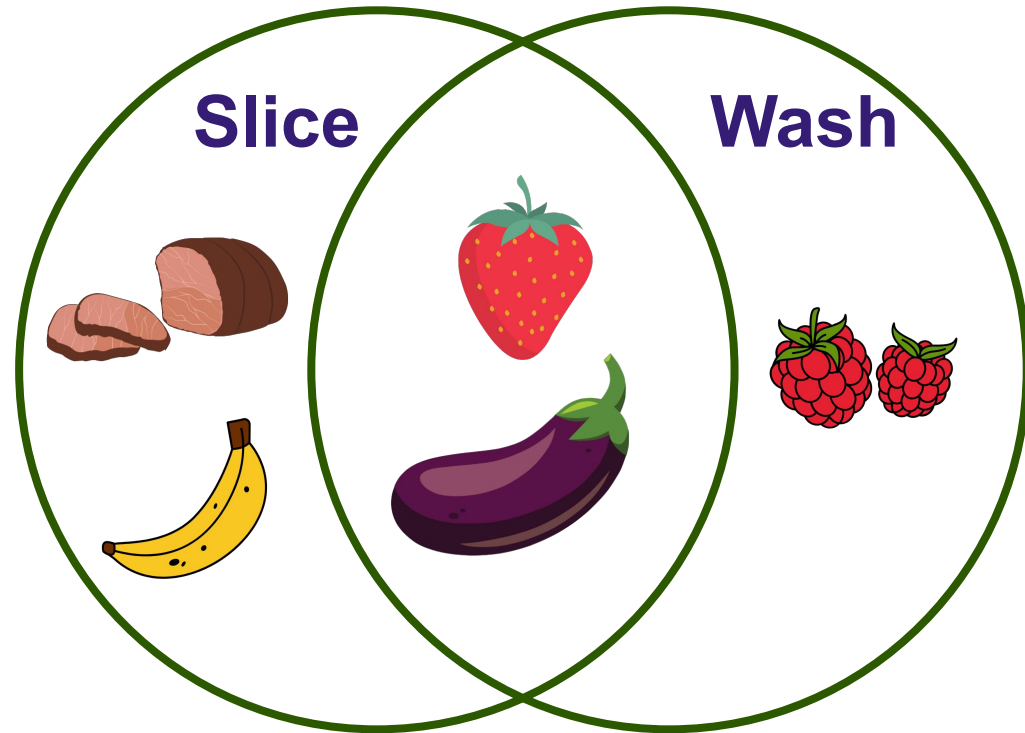
Slice

Wash



Computational Thinking

Pattern Abstraction



Computational Thinking

Algorithm Design

1. First prepare ingredients
 - a. If it is a fruit or vegetable then wash it
 - b. If it is raw meat then cook it
 - c. If it is in a package then open it
 - d. If it is big then slice it
2. Layer the ingredients
 - a. Start with a piece of bread
 - b. Repeat adding next ingredient until done
 - c. End with a piece of bread



Ears

To listen to the ideas of others

Eyes

To make observations

Mouth

To collaborate & share your conclusions with others

Hands

To do experiments & record observations

Curious Mind

To make predictions & hypothesis. To think deeply about the world.

Strong Heart

To be brave and take chances!

Tools

Beakers, Chromebooks, pencils, magnifying glasses & more

Feet

To move safely



Ears

To listen to the ideas of others

Eyes

To make observations

Mouth

To collaborate & share your conclusions with others

Hands

To do experiments & record observations

Curious Mind

To make predictions & hypothesis. To think deeply about the world.

Strong Heart

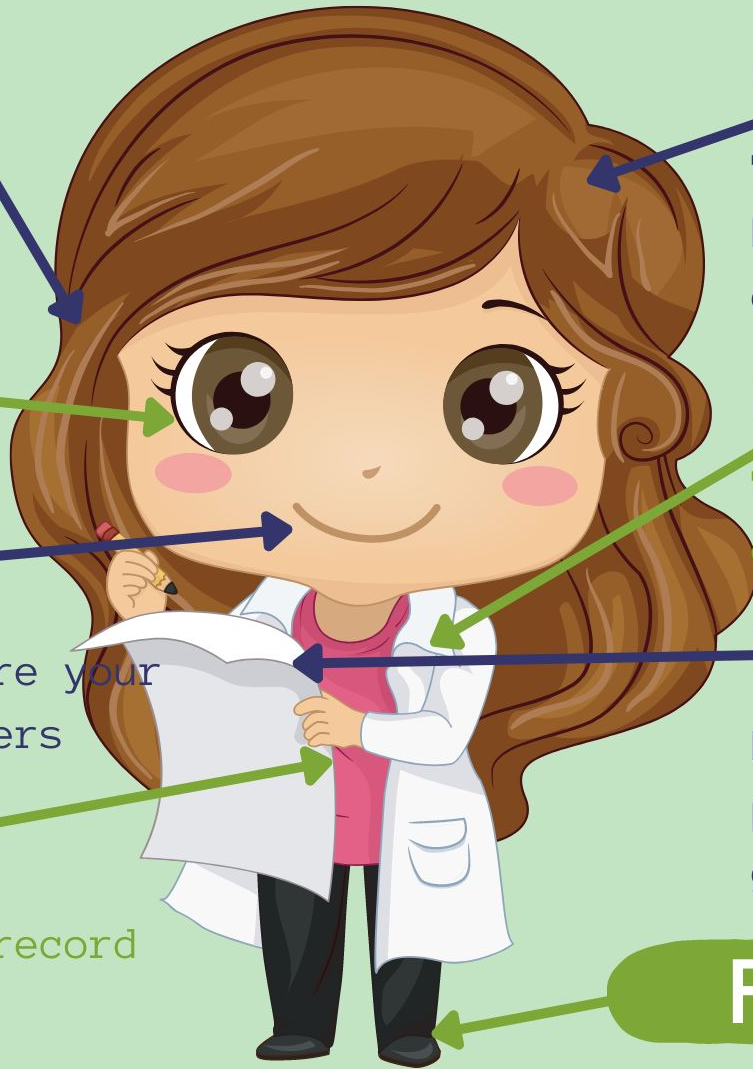
To be brave and take chances!

Tools

Beakers, Chromebooks, pencils, magnifying glasses & more

Feet

To move safely



Ears

To listen to the ideas of others

Eyes

To make observations

Mouth

To collaborate & share your conclusions with others

Tools

Beakers, Chromebooks, pencils, magnifying glasses & more

Curious Mind

To make predictions & hypothesis. To think deeply about the world.

Strong Heart

To be brave and take chances!

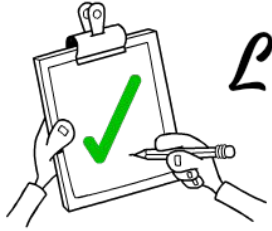
Hands

To do experiments & record observations

Feet

To move safely





Learning Outcomes

K

Children interpret instructions in the learning environment.



1

Students investigate instructions and their influence on actions and outcomes.



2

Students apply creativity when designing instructions to achieve a desired outcome.



3

Students investigate creativity and its relationship to computational thinking.



4

Students investigate and apply design in the context of computer science and technology.



5

Students create and justify a design that could be used by a human or machine to address a challenge.



6

Students create and refine computational artifacts through the use of design and abstraction.

Attack of the Kidbots

Classroom routines

Learning instructions

Other subject areas

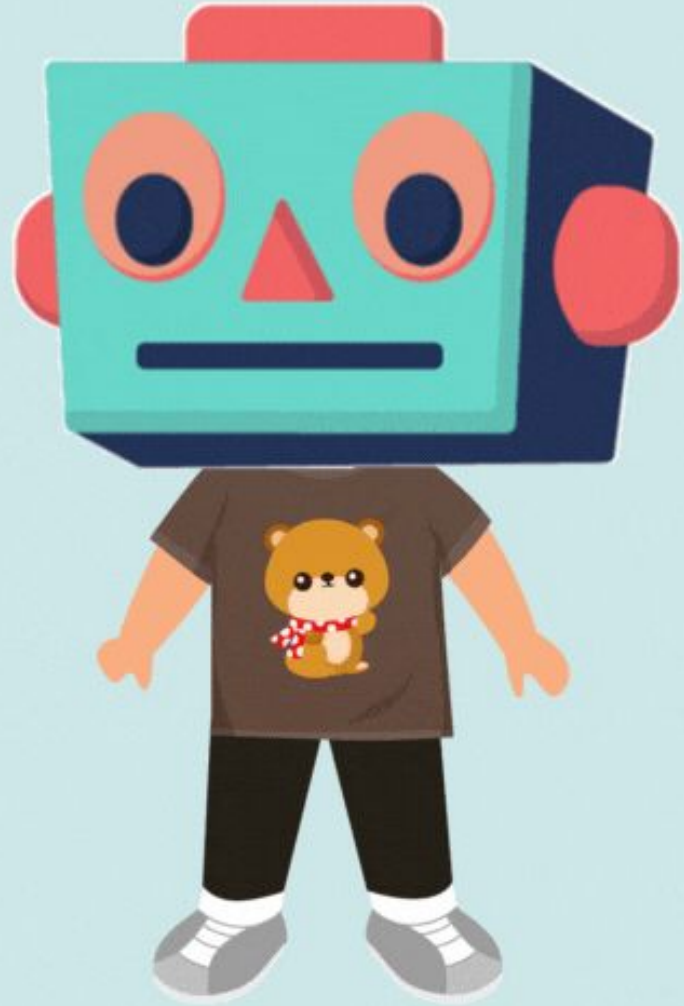
Outside of school

Community

Family

The world

In nature



Coding Knowledge

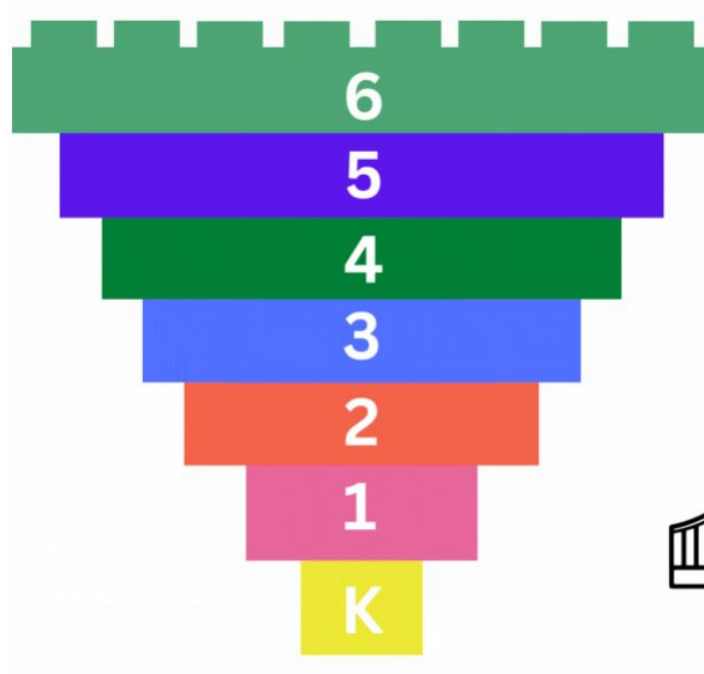
Grade 2

Debugging is the process of identifying and removing errors in a set of instructions to achieve a desired outcome.

Debugging can increase the reliability of instructions.

Many daily activities include repeated steps, such as

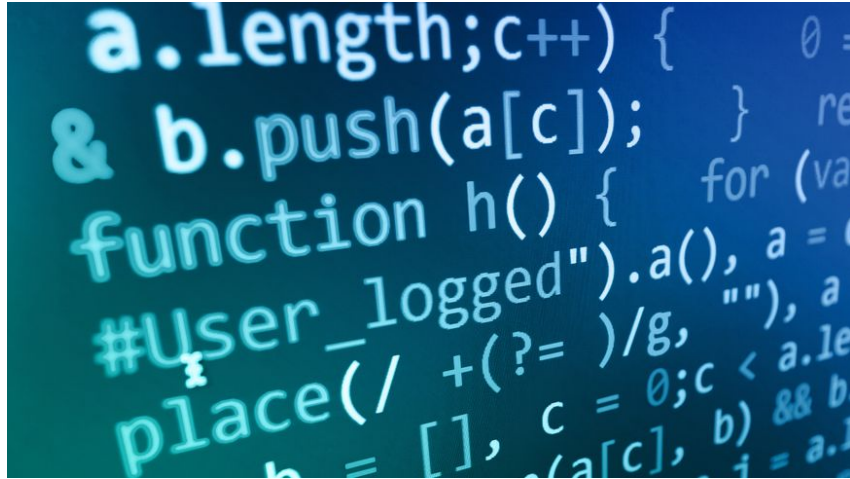
- brushing teeth
- tying one shoe and then using the same process on the other shoe

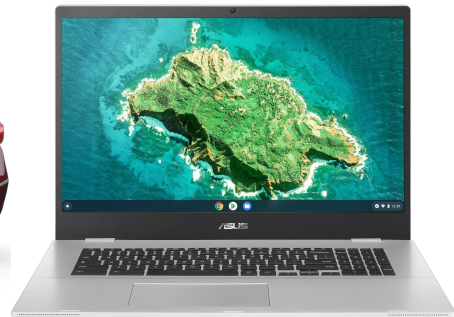


Bridging Gaps

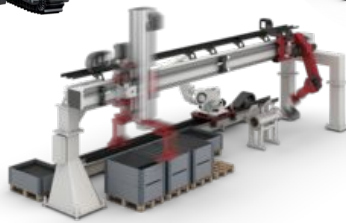
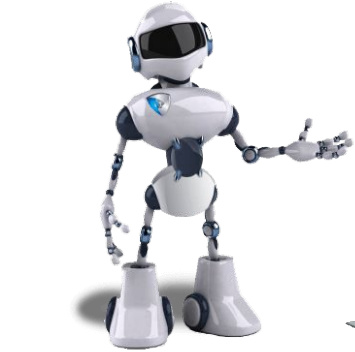
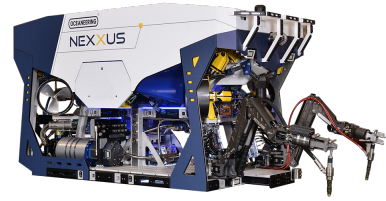
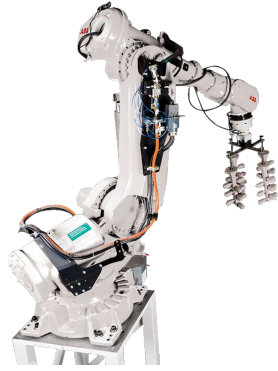


Coding consists of many languages that computers understand.



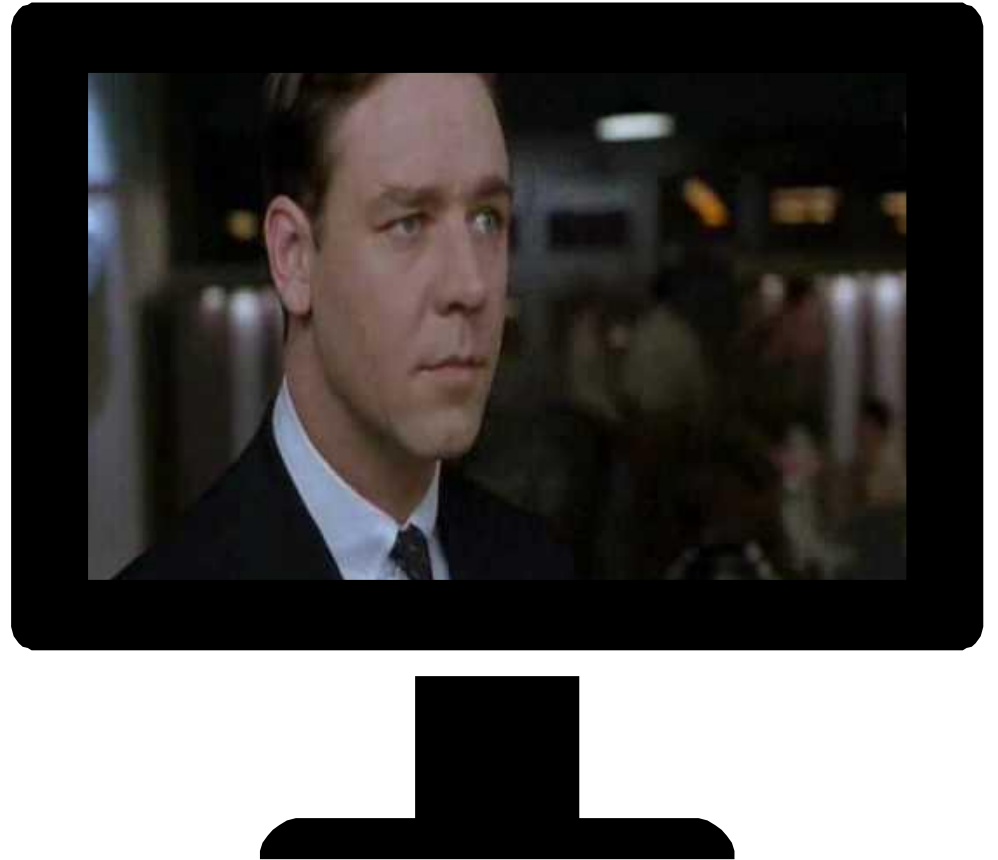


ROBOTS!!!



Coding

What we think it is...



Computer Science Vocabulary

Algorithm

A set of instructions to be followed, especially by a computer.

Event

The condition that starts or stops an algorithm.

Input

The instructions that are given to a human or machine.

Output

The outcome of the instructions.

Loop

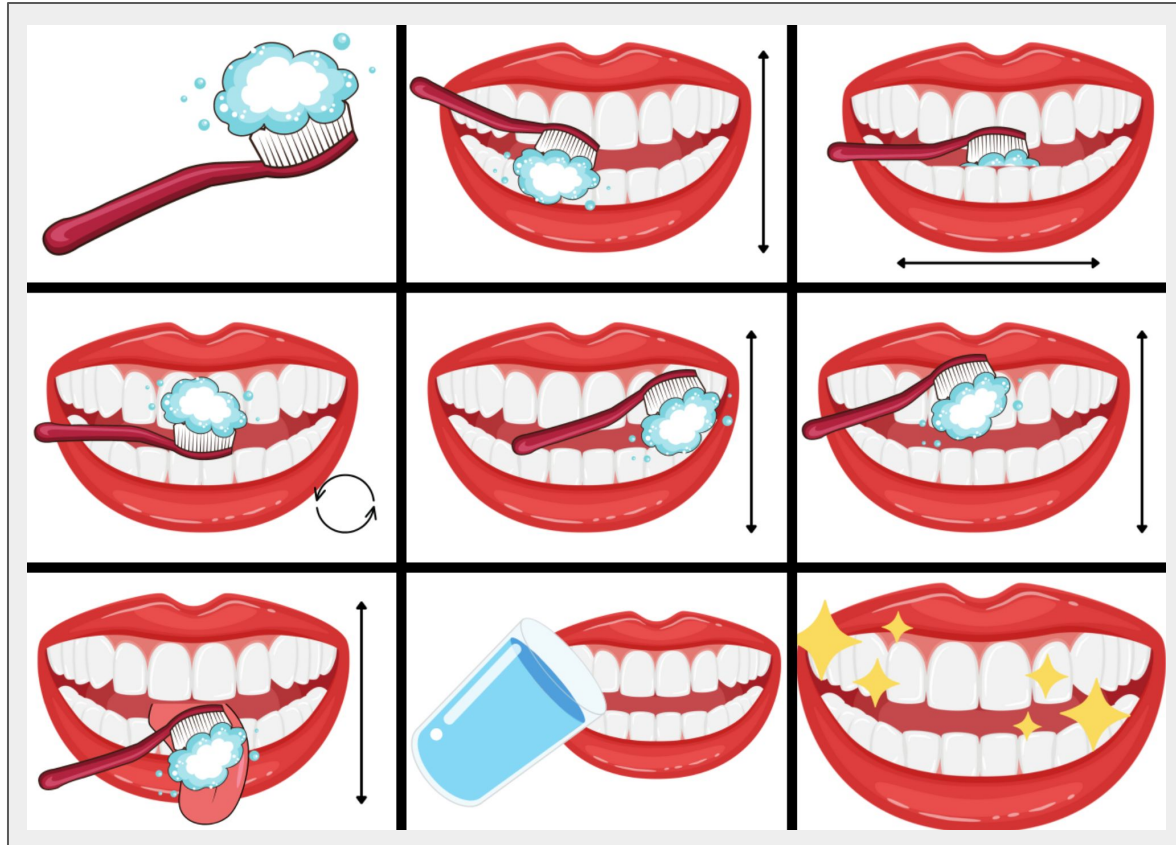
An algorithm or part of an algorithm that repeats a certain number of times, forever or until a condition is met.

Debug

Finding and fixing an error or mistake in a set of instructions (Algorithm)

grade 2

What Coding Really is



EVENT



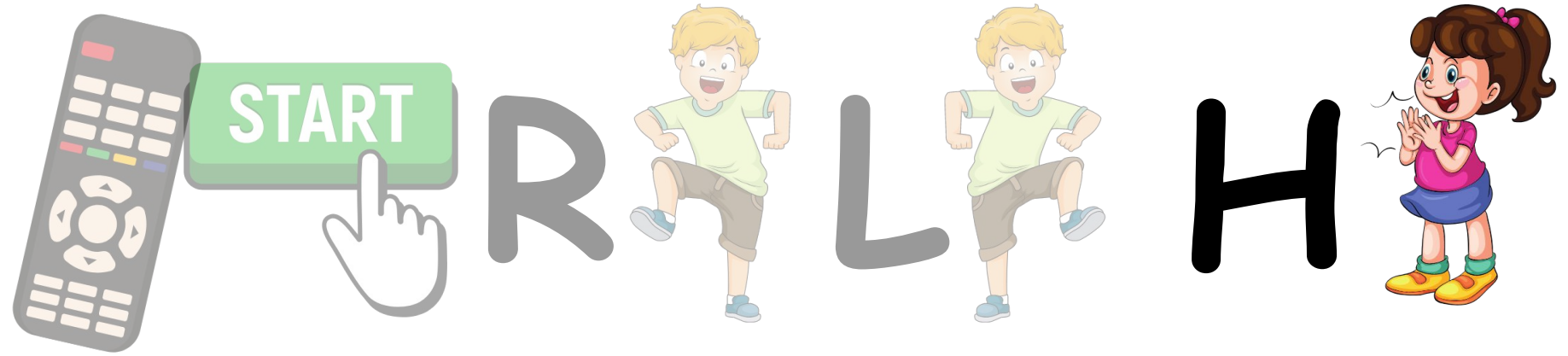
INPUT / OUTPUT



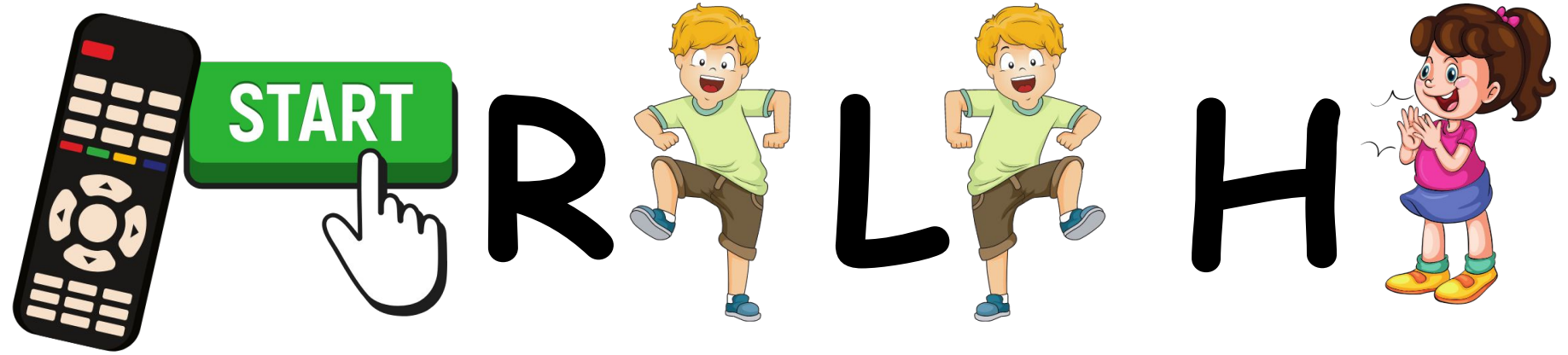
ALGORITHM



ALGORITHM



ALGORITHM



START



R

H

START



L

H

START



L

H

R

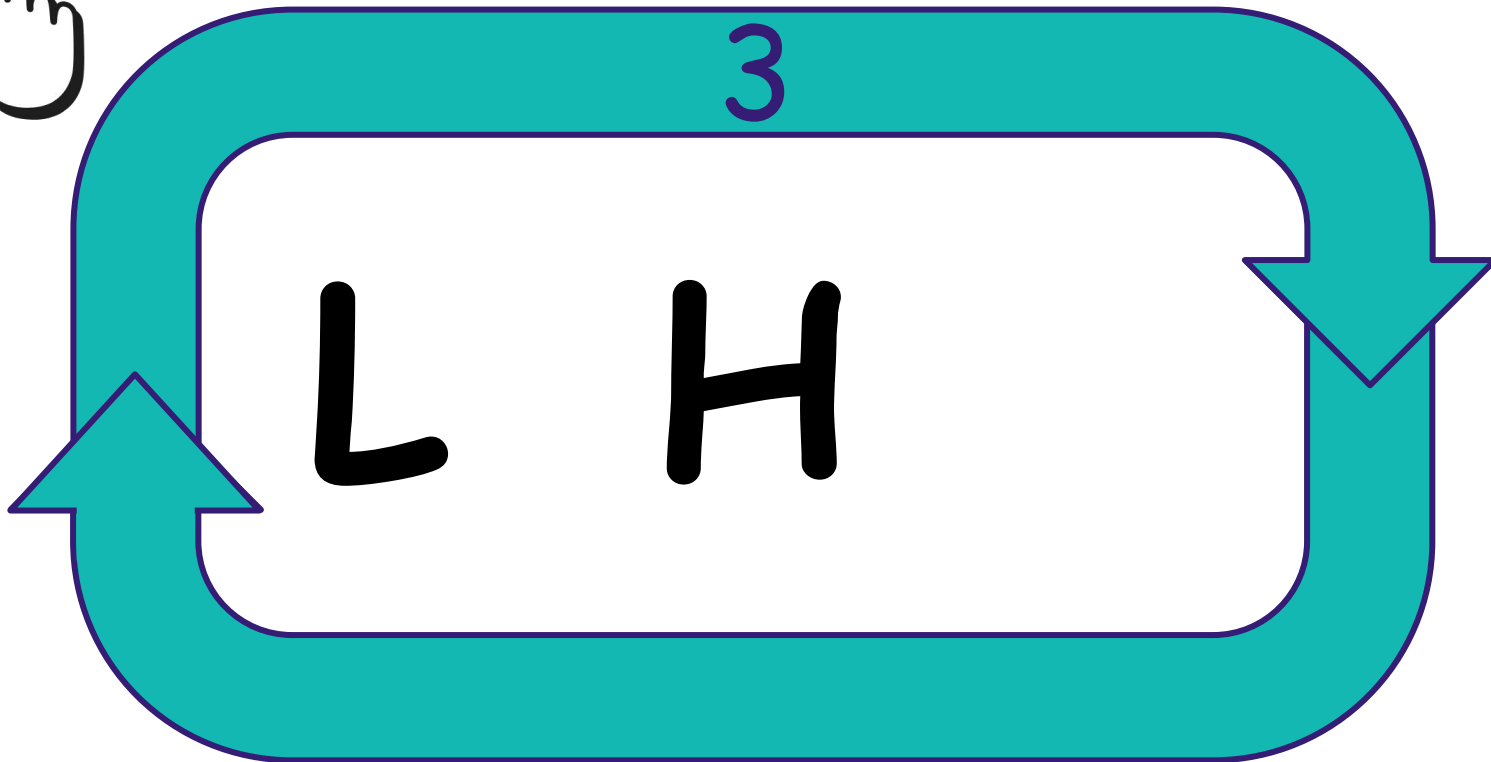
LOOP



START



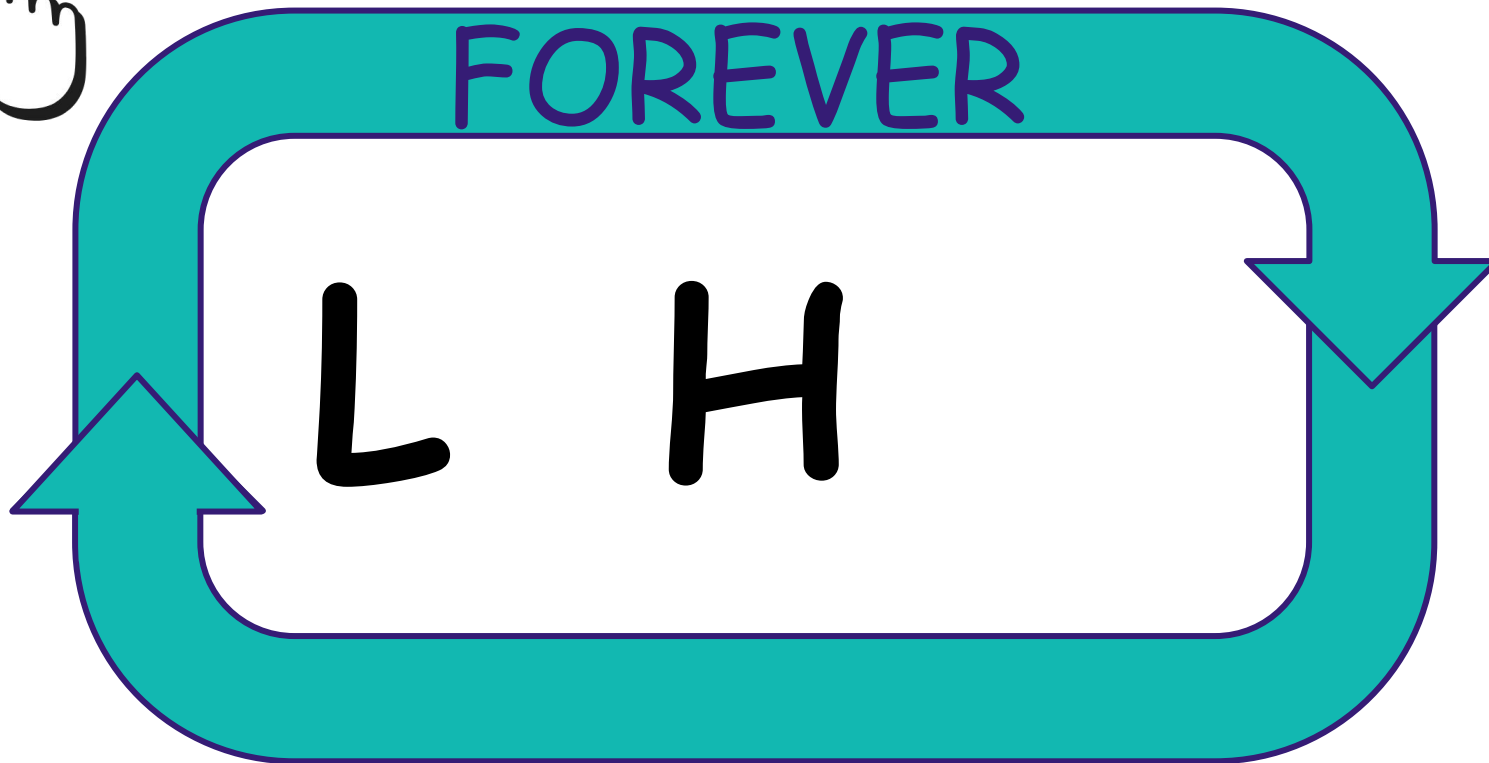
LOOP



START



LOOP



START

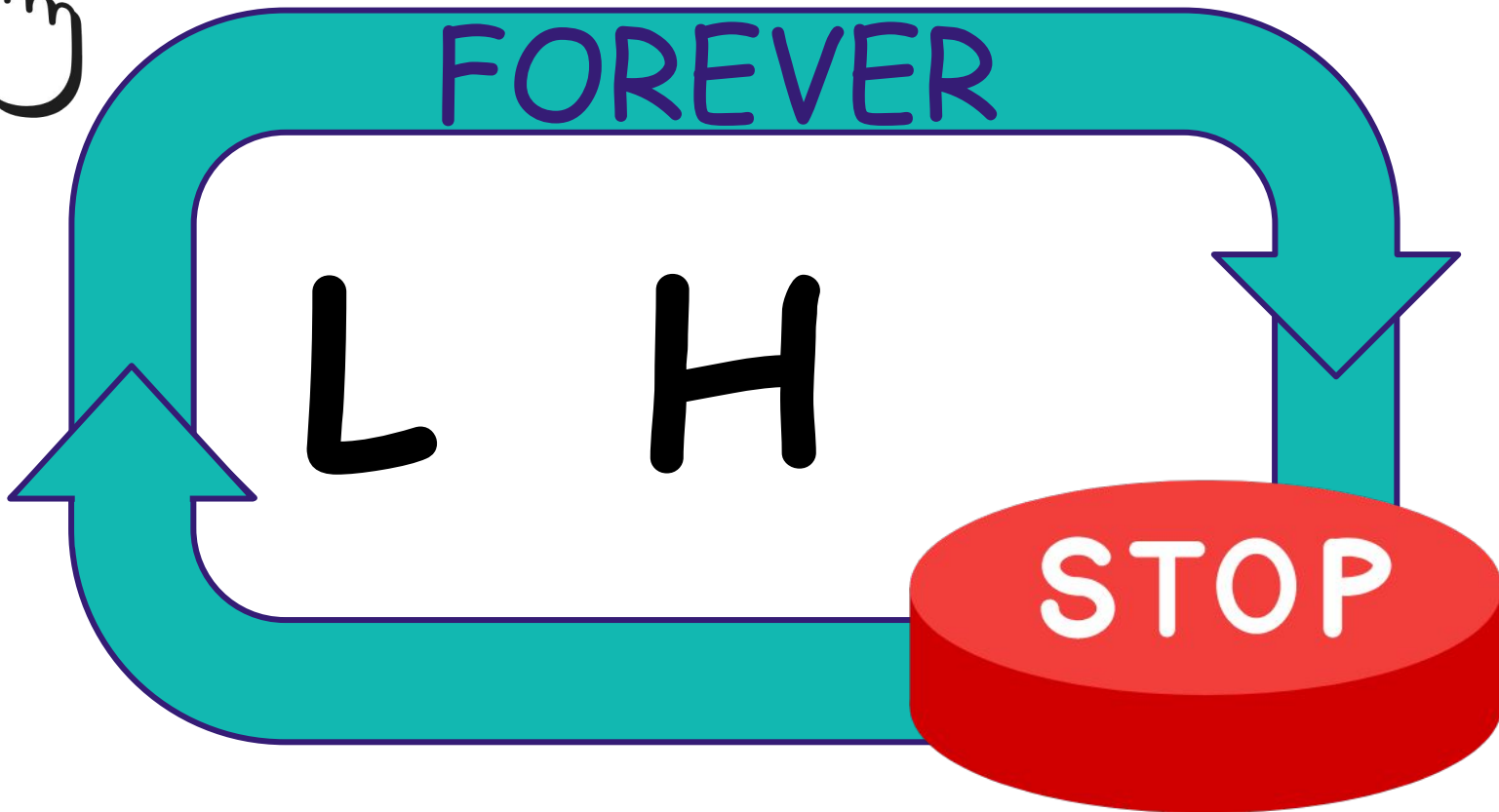


LOOP

FOREVER

L H

STOP



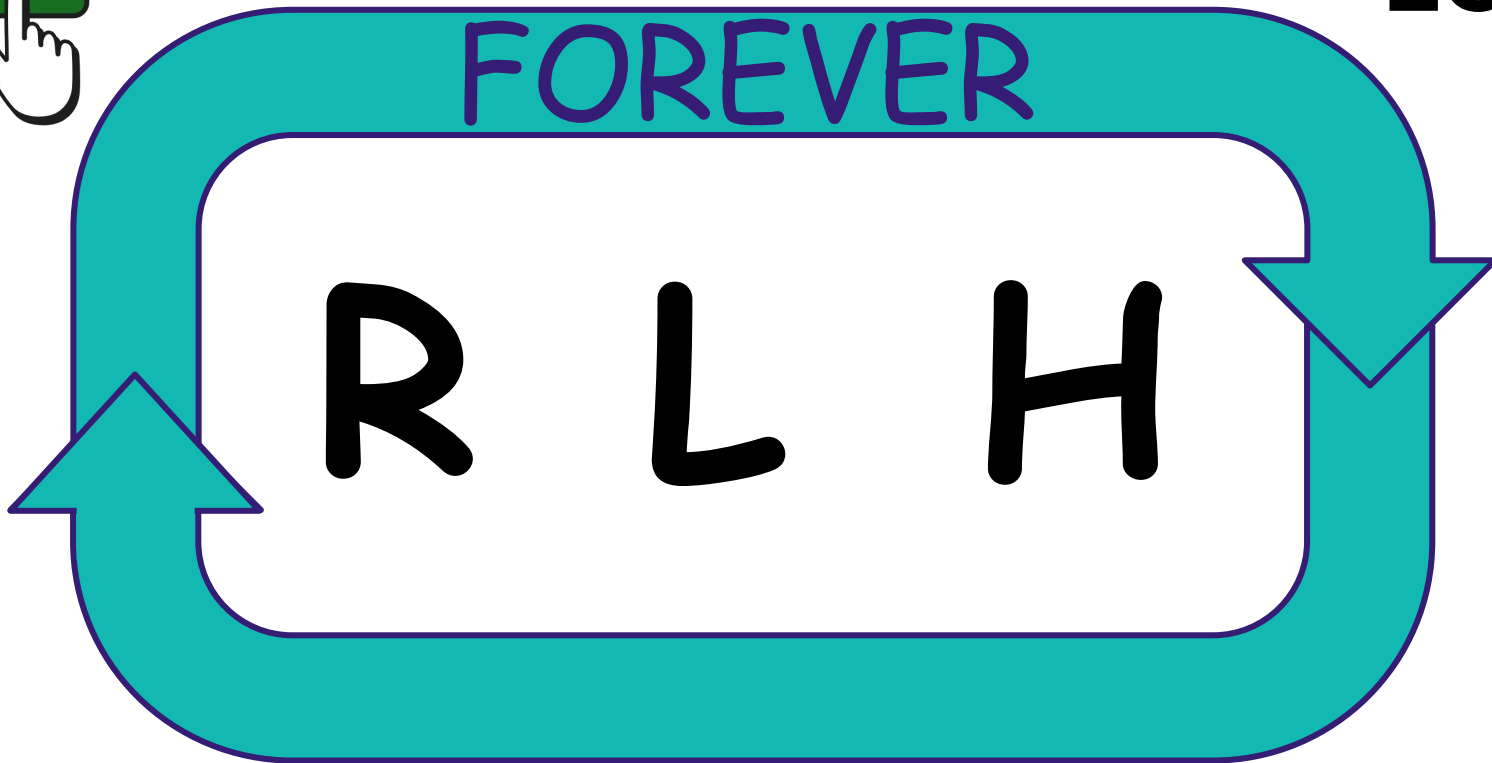
START



LOOP

FOREVER

R L H

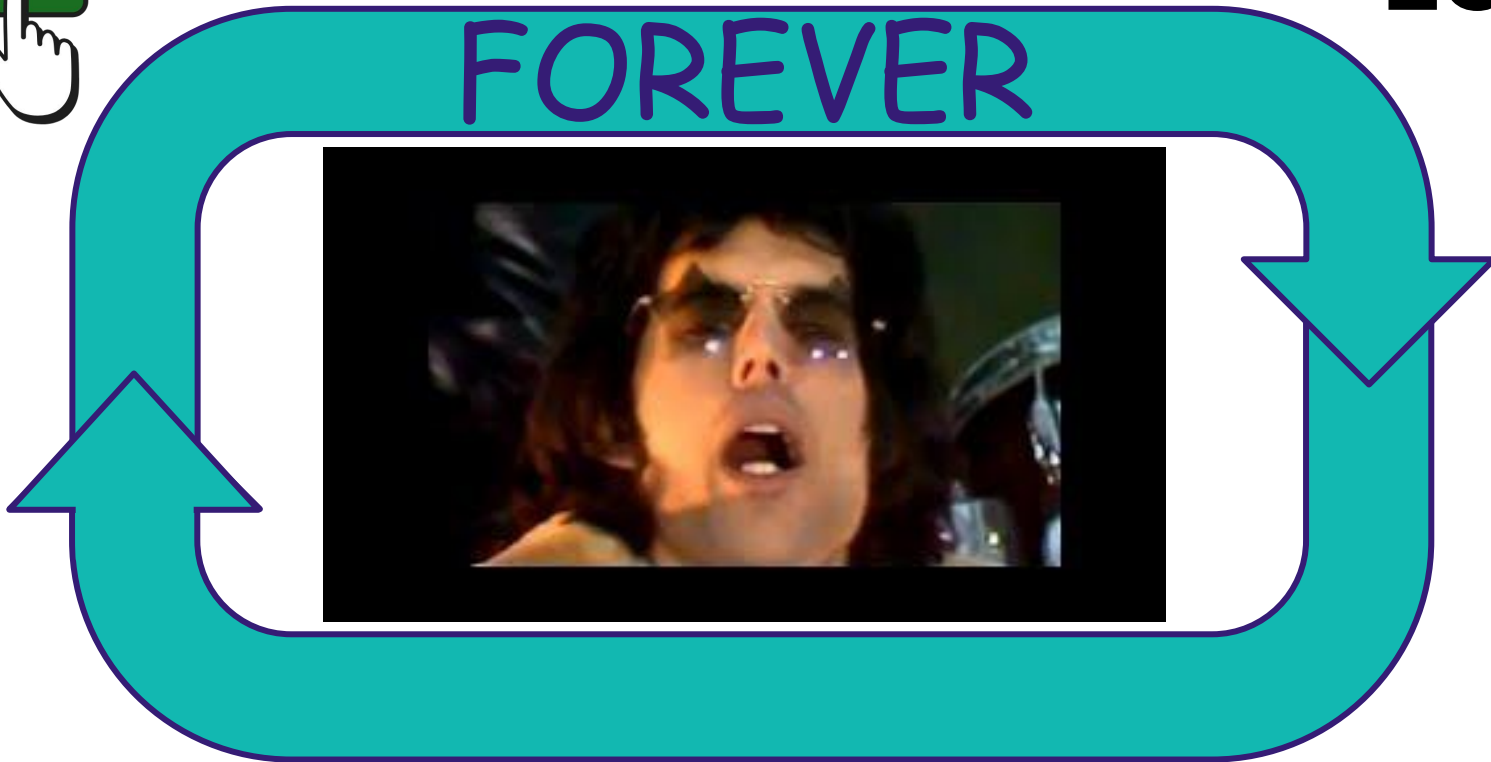


START



LOOP

FOREVER



START

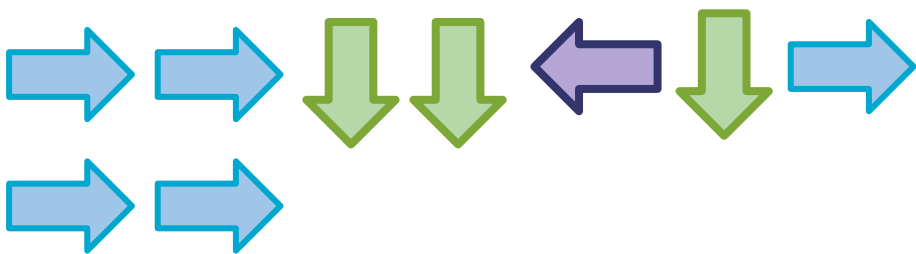


EVENT

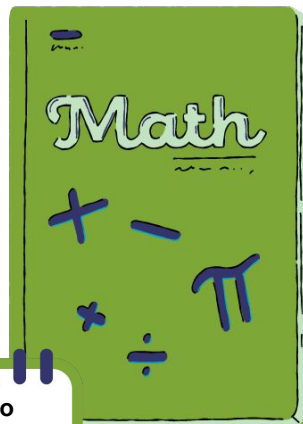
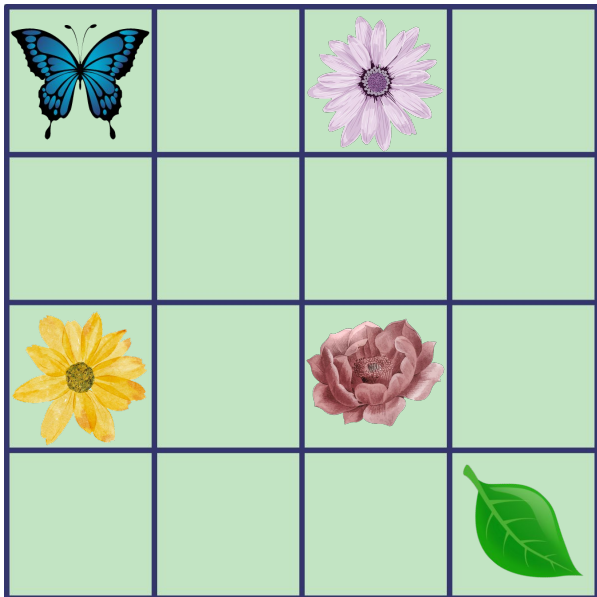
FOREVER

STOP





Will this set of instructions get the butterfly to all 3 flowers and end at the leaf?




Create a set of instructions for counting by 2's.


Can you use a "repeat" to make it more refined?

Trade with a friend and check each other's instructions

Collaborate to design an algorithm to solve a problem.



SKILLS & PROCEDURES



2, 4, 6, 8,
10, 11, 13,
15, 17, 19

Max was skip counting by 2's but he made a mistake somewhere.

Can you find the mistake? What did Max do wrong? Can you fix his mistake?





KNOWLEDGE

The process of **abstraction** includes

- determining what details to keep and what to ignore
- removing unnecessary details
- identifying important information
- generalizing patterns

Information is data that is organized to be more useful.

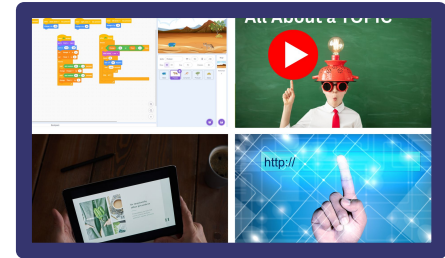
An **abstraction** is a simplified version of something complex.

Abstractions can make daily life easier; e.g.,

- simple controls on appliances
- light switches
- steering wheels
- apps

Computational artifacts can be designed to address societal needs and wants; e.g.,

- weather modelling
- communications
- automotive controls
- medical research
- apps

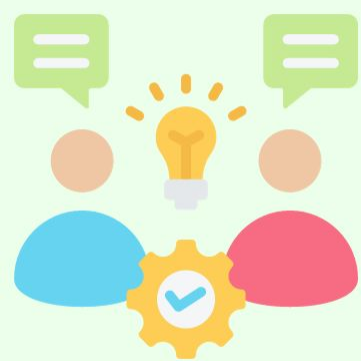


Structures used in coding include

- **sequences**
- **conditionals** (if-then-else statements)
- **loops**

Sequence structures are ordered sets of instructions within code.

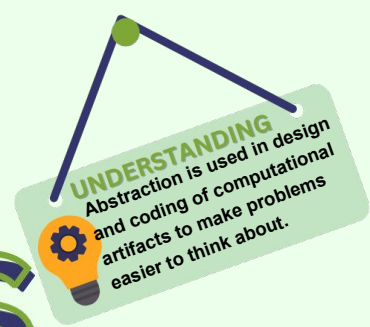
Conditional structures are statements that tell computers to complete different actions based on different situations.



UNDERSTANDING

Abstraction is used in design and coding of computational artifacts to make problems easier to think about.

SKILLS & PROCEDURES



Apply abstraction during the design process.

 SKILLS & PROCEDURES 



Identify examples of abstractions encountered in daily life.

 SKILLS & PROCEDURES 

Discuss the role of design and coding in society.

 SKILLS & PROCEDURES 

Use a visual block-based language to design code that includes relevant design structures.

 SKILLS & PROCEDURES 

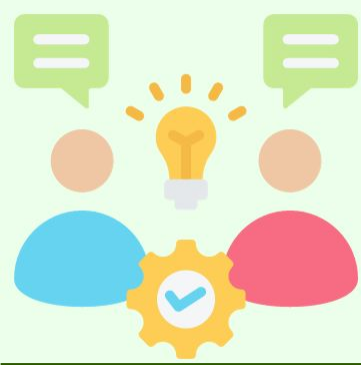


KNOWLEDGE

The use of computers, coding, and technology can have impacts that are

- personal
- social
- environmental
- economic

Impacts of computers, coding, or technology may be intentional or unintentional.



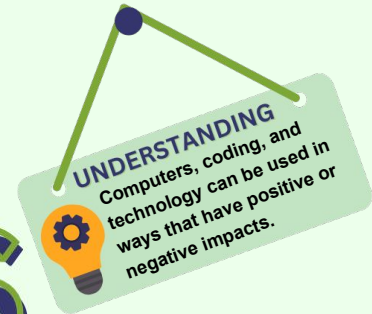
UNDERSTANDING

Computers, coding, and technology can be used in ways that have positive or negative impacts.









SKILLS & PROCEDURES



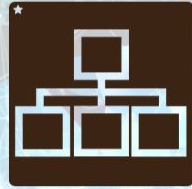
Discuss how computers, coding, or technology have had impacts.

 **SKILLS & PROCEDURES** 

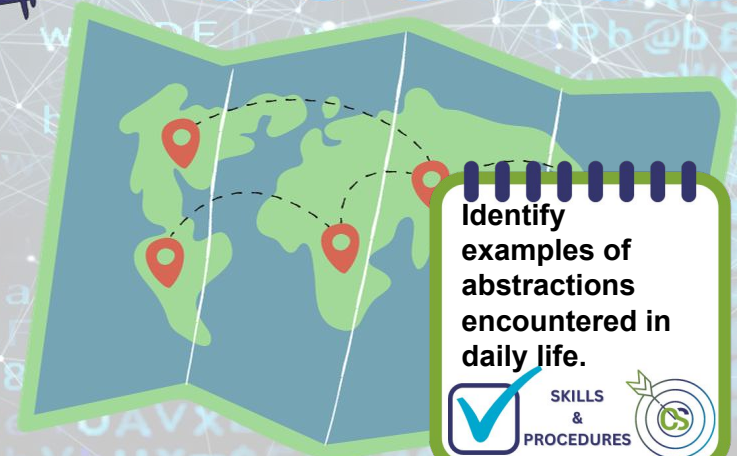
Predict possible impacts of computers, coding, or technology.

 **SKILLS & PROCEDURES** 

ABSTRACTION



```
101001100010111
000111010111001
011010101001010
110110110101010
101010101001010
```



Identify examples of abstractions encountered in daily life.

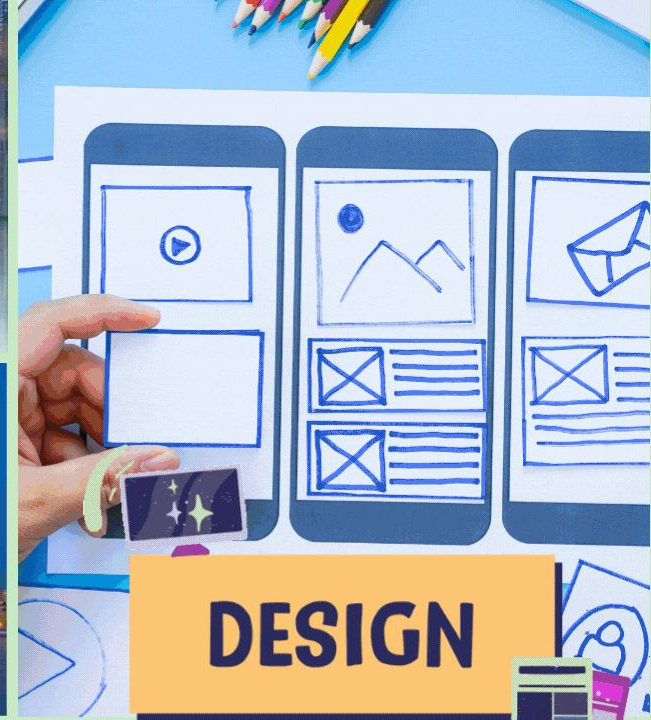




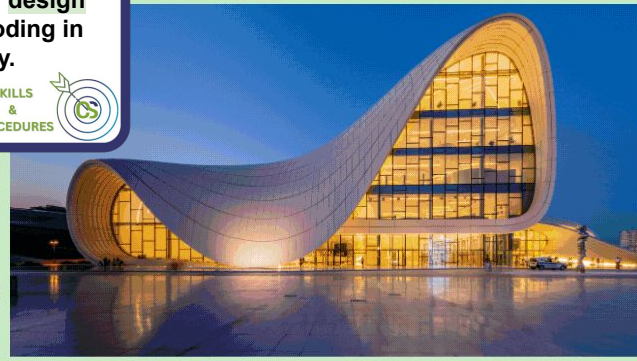
Discuss the role of design and coding in society.



SKILLS & PROCEDURES



DESIGN





Discuss the role of design and coding in society.

SKILLS & PROCEDURES



Discuss how computers, coding, or technology have had impacts.

SKILLS & PROCEDURES

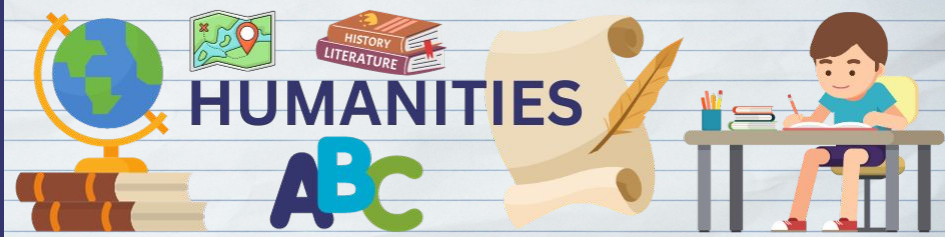




FINE ARTS

Create a picture or 3D model of your vision of a future world.

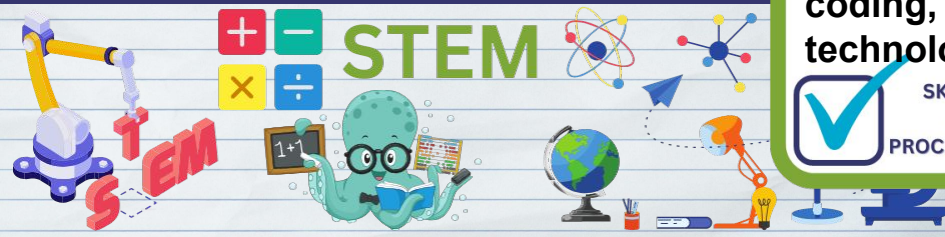
How might the Arts be impacted by technology in the future?



HUMANITIES

Write about what a future world might look like.

How might communication be impacted by technology?



STEM

Predict possible impacts of computers, coding, or technology.



SKILLS & PROCEDURES

Research technology concepts and evaluate the likelihood they will exist one day.

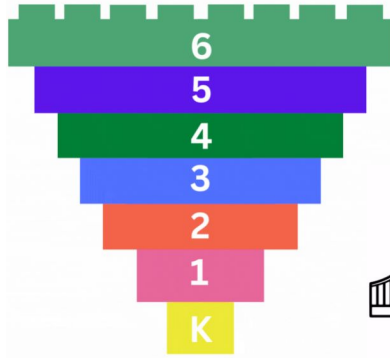
Use a program like Scratch to create a story or game about the future.



PHYSICAL EDUCATION & WELLNESS

Research technology concepts related to health and medicine. What concepts do you think would be the most impactful on human health?

Use a visual block-based language to design code that includes relevant design structures.



Bridging Gaps



when  clicked

go to x y

grade 5

Relate a block of code to an outcome or a behaviour.



Explain what will happen when single or multiple blocks of code are executed.

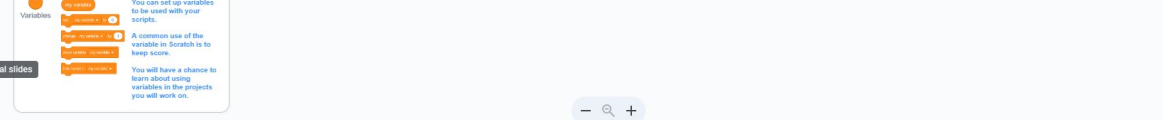
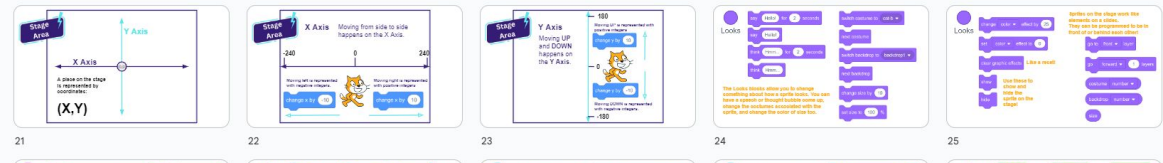
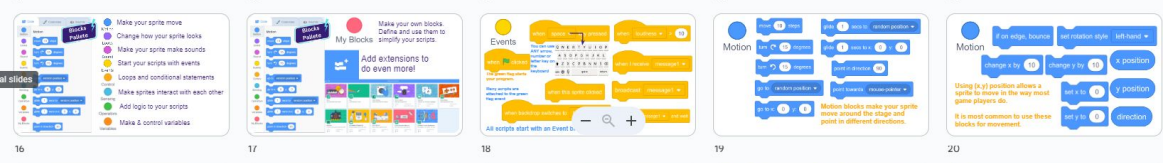
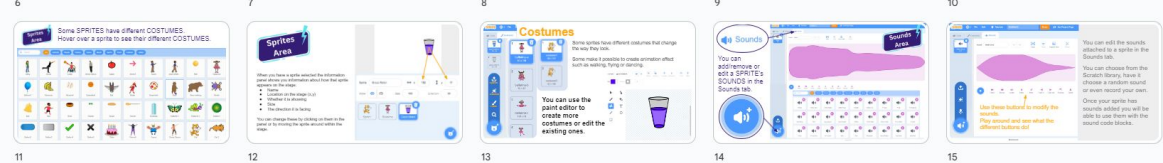
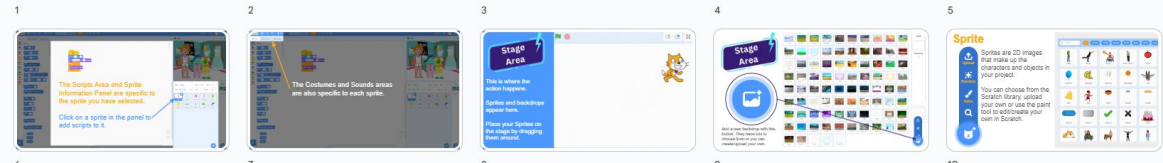
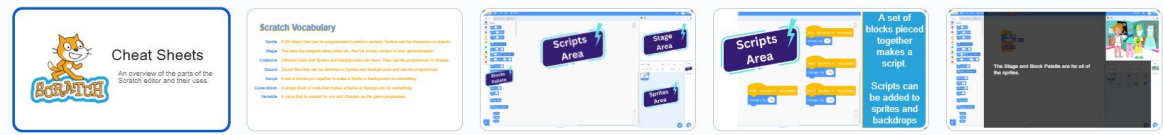


Translate a given algorithm to code using a visual block-based language.



Design an algorithm that includes a loop and translate it into code.





<https://bit.ly/ScratchCheatCards>



SCRATCH

scratch.mit.edu

IDEAS



What will you create?

Choose a tutorial

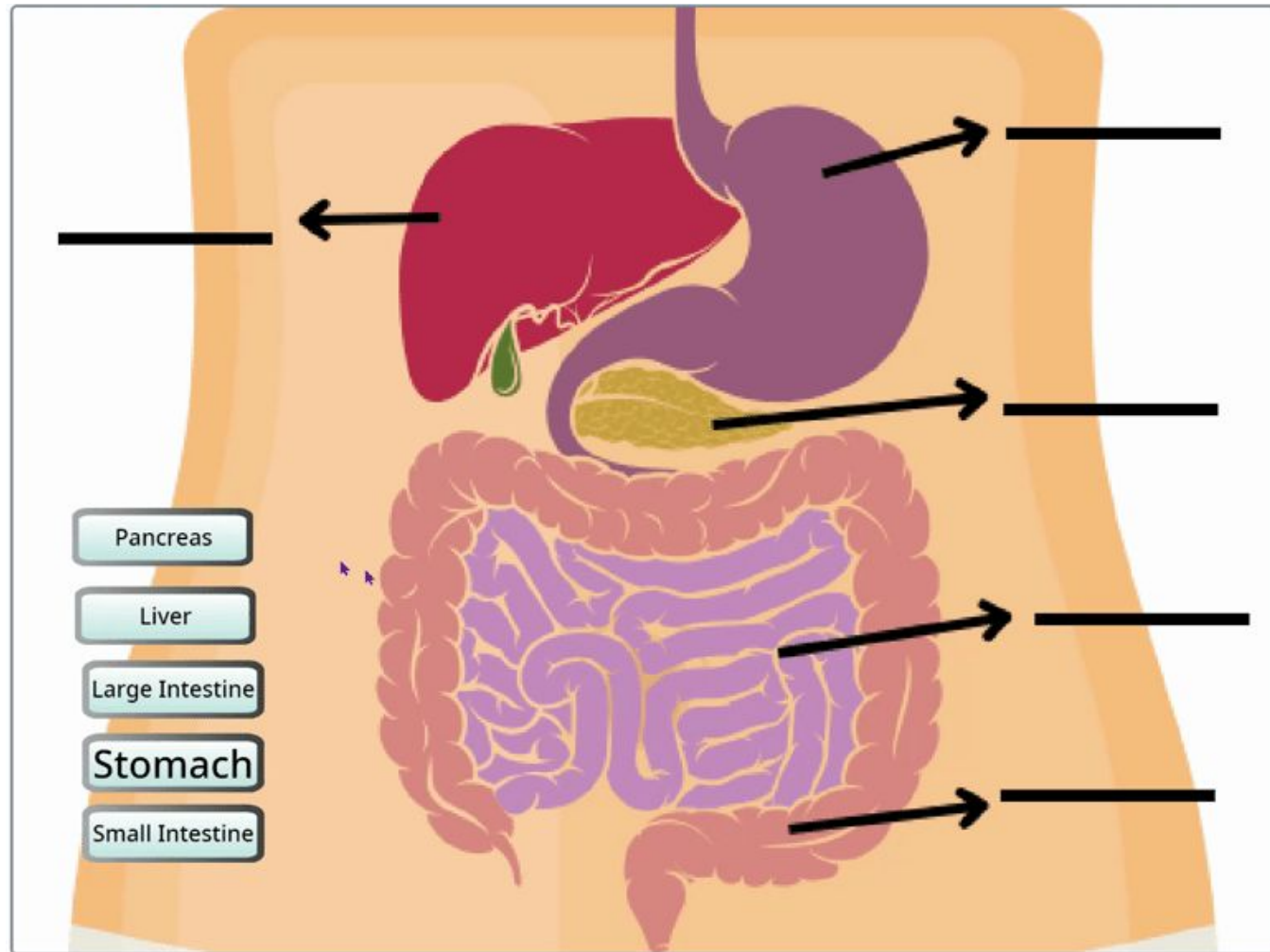


Getting Started

New to Scratch? Try the Getting Started tutorial.

Try It!

[etwithmrsd.com](https://www.etwithmrsd.com)

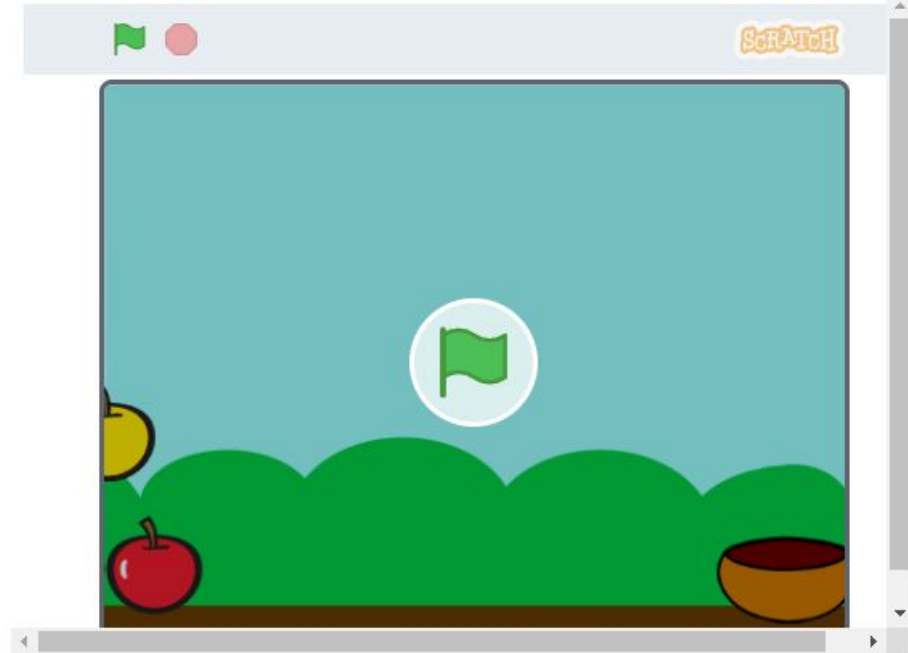


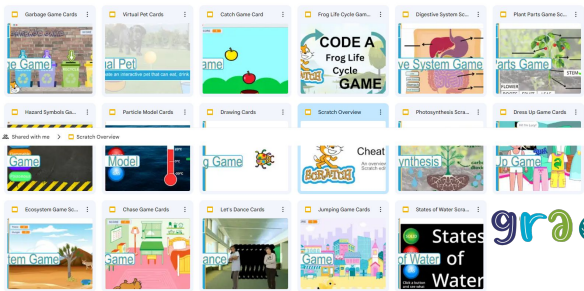
CATCH GAME

Instructions



Example





grade 5

Relate a block of code to an outcome or a behaviour.

SKILLS & PROCEDURES

Explain what will happen when single or multiple blocks of code are executed.

SKILLS & PROCEDURES

Translate a given algorithm to code using a visual block-based language.

SKILLS & PROCEDURES

Design an algorithm that includes a loop and translate it into code.

SKILLS & PROCEDURES

1. Create a catching game with points

1

2. 1. Delete the cat sprite
2. Choose a new sprite to be your catcher
3. Choose a backdrop

2

3. Drag your sprite to the bottom of the stage where you want it placed at the start of each game.

3

4. Add this script to your catcher sprite to tell it to go to the starting point and to move left and right

4

5. 1. Add a new sprite that will be the one to catch.
2. Add this code.
3. Drag your sprite to the top of the stage, click the green flag and watch it fall.

5

6. Add the "if" statement to your script and create one to make the apple start at the top for each new game.

6

7. Variables

1. In the orange variables area click "Make a Variable"
2. Name your new variable "Score" and click OK.

7

8. Add this script to your catch sprite

8

9. Change the colour of the new sprite by using the fill button in the Costumes tab. You will see all of the code was copied along with the new sprite!

9

10. Change the colour of the new sprite by using the fill button in the Costumes tab. You will see all of the code was copied along with the new sprite!

10

11. Edit this script to give more points for a golden apple.

11

12. Hover over the "Add Sprite" button and choose to Paint a new sprite.

12

13. Use the Text tool to create a message that will display when a player wins.

13

14. Add this script to your new sprite.

14

15. Add this script to your new sprite.

15

CODE.ORG

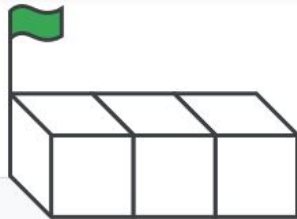
Curriculum Catalog



Find the perfect fit for your learning environment:
Code.org courses, tutorials, and more designed for all
ages and experience levels.

<p>ARTIFICIAL INTELLIGENCE 11 AI and Machine Learning Grades: 6-12 Duration: Quarter</p> <p>Quick View Assign</p>	<p>ARTIFICIAL INTELLIGENCE 11 AI for Oceans Grades: 3-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>DATA 11 AP CSA Consumer Review Lab Grades: 9-12 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING CS Fundamentals: Course B Grades: 1-2 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING CS Fundamentals: Course C Grades: 2-3 Duration: Month</p> <p>Quick View Assign</p>	<p>GAMES AND ANIMATIONS 12 CS Fundamentals: Course D Grades: 3-4 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING Code Break Grades: 8-12 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING Code Break for Younger Students Grades: K-7 Duration: Month</p> <p>Quick View Assign</p>	<p>GAMES AND ANIMATIONS 11 Code Your Own Sports Game Grades: 2-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>GAMES AND ANIMATIONS 11 Computer Science A Grades: 9-12 Duration: School Year</p> <p>Quick View Assign</p>	<p>ARTIFICIAL INTELLIGENCE 12 Computer Science Discoveries Grades: 6-12 Duration: School Year</p> <p>Quick View Assign</p>	<p>CYBERSECURITY 14 Computer Science Principles Grades: 9-12 Duration: School Year</p> <p>Quick View Assign</p>
<p>DATA 11 AP CSA Data Lab Grades: 9-12 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING AP CSA Magpie Lab Grades: 9-12 Duration: Week</p> <p>Quick View Assign</p>	<p>PROGRAMMING Accelerated Intro to CS Course Grades: 4-8 Duration: Month</p> <p>Quick View Assign</p>	<p>INTERNET 14 CS Fundamentals: Course E Grades: 4-5 Duration: Quarter</p> <p>Quick View Assign</p>	<p>INTERNET 14 CS Fundamentals: Course F Grades: 5 Duration: Quarter</p> <p>Quick View Assign</p>	<p>PROGRAMMING CS Fundamentals: Express Course Grades: 3-8 Duration: Quarter</p> <p>Quick View Assign</p>	<p>ART AND DESIGN 11 Code with Anna and Elsa Grades: 2-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>ENGLISH LANGUAGE ARTS 14 Coding Book Covers Grades: 3-5 Duration: Week</p> <p>Quick View Assign</p>	<p>HISTORY 14 Coding Interactive Maps Grades: 3-6 Duration: Week</p> <p>Quick View Assign</p>	<p>PROGRAMMING Counting Activity Grades: 3-5 Duration: Lesson</p> <p>Quick View Assign</p>	<p>PROGRAMMING Course 1 Grades: K-1 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING Course 2 Grades: 3-5 Duration: Month</p> <p>Quick View Assign</p>
<p>ART AND DESIGN 11 Artist Grades: 2-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>CYBERSECURITY Blockchain Grades: 7-12 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING CS Fundamentals: Course A Grades: K-1 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING CS Fundamentals: Pre-reader Express Grades: K-3 Duration: Month</p> <p>Quick View Assign</p>	<p>GAMES AND ANIMATIONS 11 Choose Your Team and Make a Basketball Game Grades: 2-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>PROGRAMMING Classic Maze Grades: 2-12 Duration: Lesson</p> <p>Quick View Assign</p>	<p>MATH 12 Coding a Geometric Star Quilt Grades: 3-8 Duration: Week</p> <p>Quick View Assign</p>	<p>ENGLISH LANGUAGE ARTS 14 Coding a Time Capsule Grades: 3-8 Duration: Week</p> <p>Quick View Assign</p>	<p>ENGLISH LANGUAGE ARTS 13 Coding with Poetry Grades: 3-8 Duration: Week</p> <p>Quick View Assign</p>	<p>PROGRAMMING Course 3 Grades: 3-5 Duration: Month</p> <p>Quick View Assign</p>	<p>PROGRAMMING Course 4 Grades: 4-5 Duration: Month</p> <p>Quick View Assign</p>	<p>PHYSICAL COMPUTING 13 Creating Apps with Devices (Circuit Playground) Grades: 6-12 Duration: Quarter</p> <p>Quick View Assign</p>

Google CS First



Get started with CS First

Coding instructions like hints, highlights, and text-to-speech live inside *Scratch for CS First*



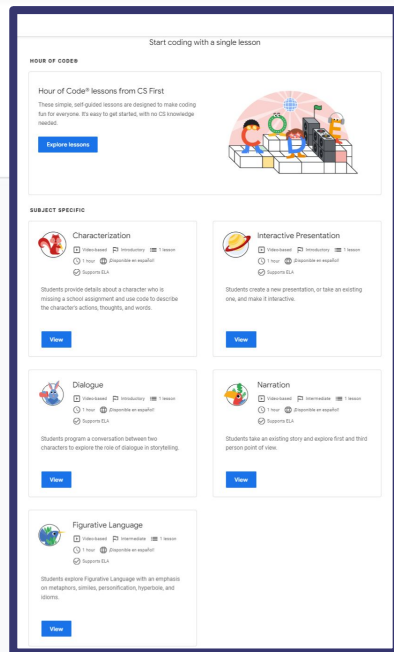
Welcome to CS First

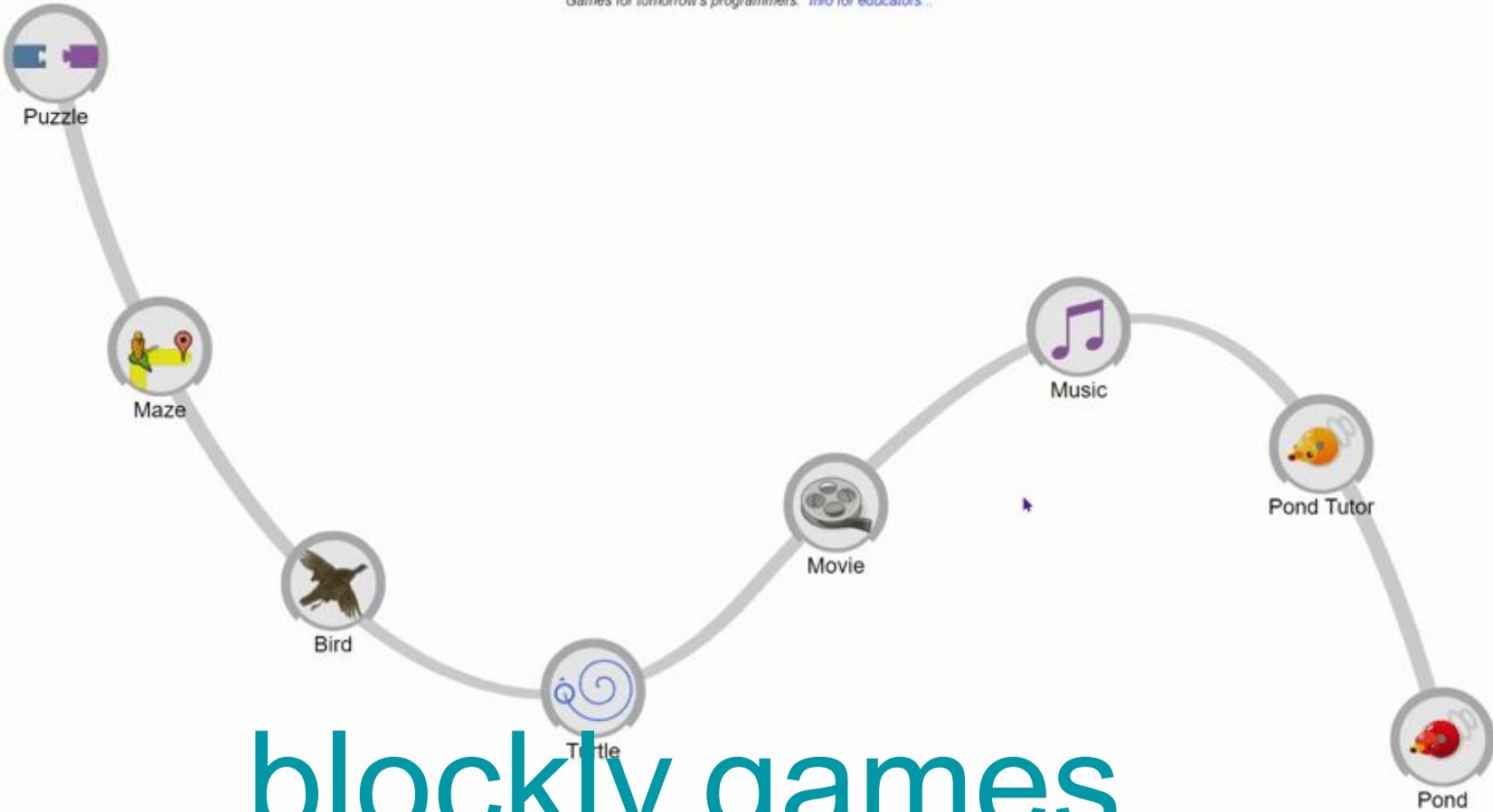
Text-based Introductory 4 lessons

4–6 hours

Students go on a first journey through Scratch for CS First with four fun projects that introduce the core elements of Scratch and foundational coding skills.

View





blockly.games

Toys!!!

Your ATA Library

 The Alberta Teachers' Association

Kits in Your ATA Library

Makerkits A-D

Makerkits E-G

Makerkits H-L

Makerkits M-O

Makerkits P-S

Makerkits T-Z

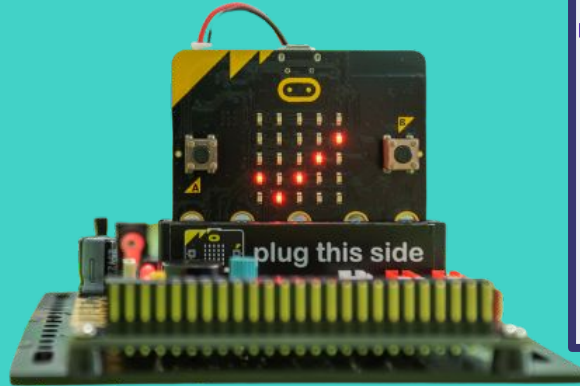


Toys!!!

I highly recommend!



micro:bits



Sphero Minis

