SCOPE & SEQUENCE – K-8

Computational Thinking

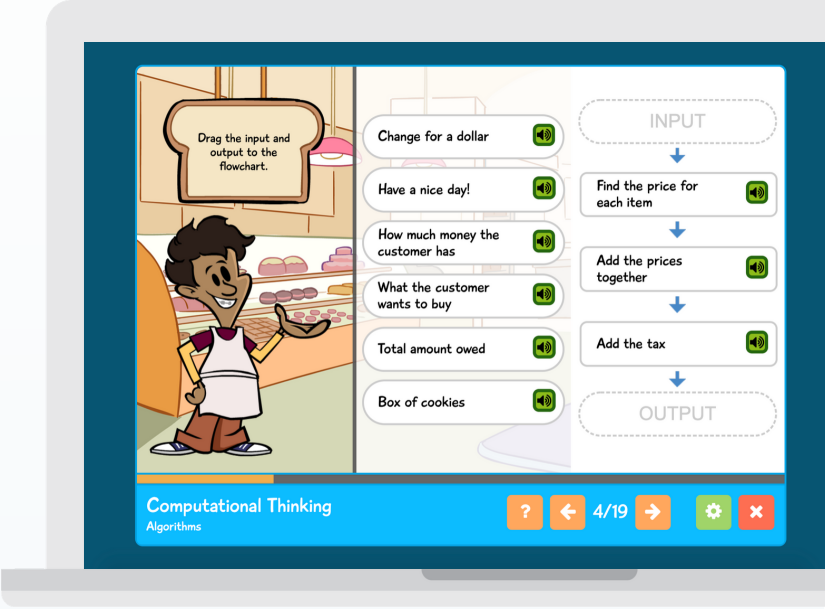
Learning.com's K-8 computational thinking curriculum develops students' problem-solving skills through scaffolded instructional content. The curriculum familiarizes students with computational thinking concepts by guiding them through interactive learning sequences, collaborative activities, and hands-on projects.

The computational thinking sequence blends computer science with our digital literacy curriculum so students become adept 21st-century problems solvers with career-ready digital skills and conceptual knowledge that are transferable to any context.

In addition, the track also cultivates metacognition, so students reflect on and understand their thinking and learning process. This focus helps students build a growth mindset and practice grit and perseverance in the face of complex data, open-ended questions, and ambiguous problems. Students, therefore, become agents of their own learning.

Learning.com's instructional content includes:

- **Lessons:** Self-paced, digital videos with an interactive interface
- **Application Exercises:** Hands-on and collaborative learning opportunities with unplugged options
- **Skills Checks:** Pre and post grade-level tests designed to capture student growth
- **Quizzes:** Formative check-ins to inform instructional decisions
- **Discussions:** Guided conversations to help students practice communication and articulate their understanding



ABOUT LEARNING.COM

Learning.com's digital literacy curriculum enables schools to develop students' technology skills throughout core instruction.

sales@learning.com • 800.580.4640 • learning.com

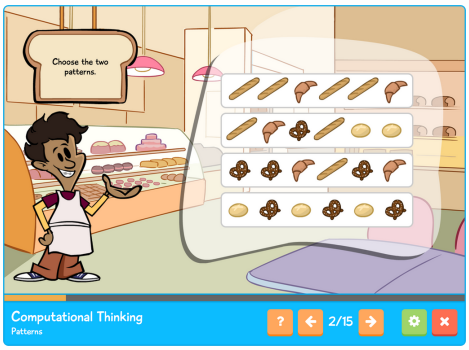
Learning.com

Skills Pre-Check

Estimated Time: 15 Minutes

UNIT 1

Computational Thinking: Patterns



Big Idea: Patterns enable us to make predictions about occurrences and synthesize sequences.

Skills

- Analyze sequences to identify whether a pattern is present.
- Apply a pattern's rule to predict the next step or find what is missing.
- Practice pattern recognition by designing pattern sequences.

UNIT 2

Computational Thinking: Directions

Big Idea: By building a set of directions (aka algorithms), we decompose the process for performing a task, creating a result, or solving a problem.

Skills

- Model daily processes by creating and following algorithms.
- Decompose the steps needed to solve a problem into a precise sequence of instructions.
- Debug errors in an algorithm by ensuring the steps are listed and in the correct order.
- Describe steps taken and choices made during the iterative process of creating algorithms.
- Analyze information, or data, in order to organize that information into useful and accurate directions.

Bonus Resource

Free unplugged activity to help students practice creating and applying direction sequences.

DOWNLOAD NOW

Skills Post-Check

Estimated Time: 15 Minutes

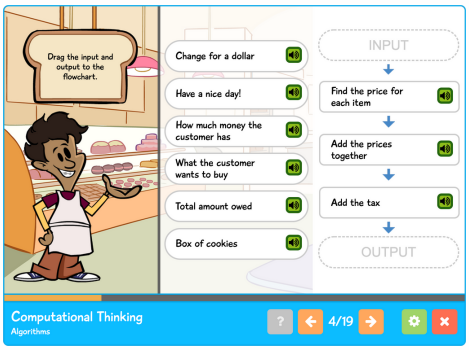
Skills Pre-Check

Estimated Time: 15 Minutes

UNIT 1

Computational Thinking: Algorithms

Big Idea: Algorithms represent tasks and procedures that are repeated, which enables these processes to be automated digitally or replicated by humans.



Skills

- Analyze a situation by looking at an output in order to develop a problem statement.
- Develop algorithms with multiple branches, representing decisions that impact the overall output.
- Test and debug an algorithm to ensure it runs as intended.
- Apply algorithms to automate tasks.
- Analyze algorithms for the same task to determine which is the most appropriate.

UNIT 2

Computational Thinking: Modelling

Big Idea: Models help to visualize abstract concepts and represent decomposed facets of larger problems.

Skills

- Decompose open-ended questions into smaller problems, like inputs and outputs.
- Practice creating a binary model.
- Build data models to represent similar data sets.
- Analyze variable and design simulations.
- Assess prototypes and alter variables.

Bonus Resource

Free unplugged activity to help students practice building data models and assessing prototypes.

DOWNLOAD NOW

Skills Post-Check

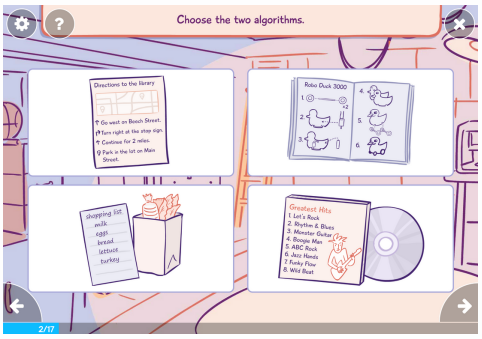
Estimated Time: 15 Minutes

Skills Pre-Check

Estimated Time: 15 Minutes

UNIT 1

Computational Thinking: Algorithmic Problem Solving



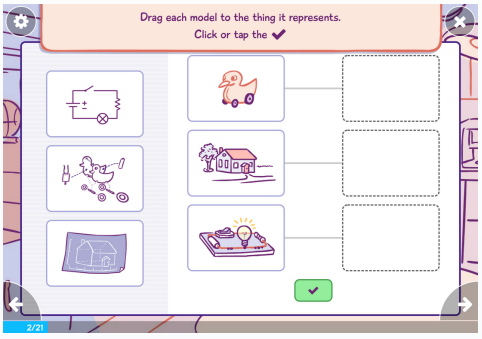
Big Idea: Having a systematic problem-solving process helps us to work with open-ended and ambiguous questions.

Skills

- Decompose problems and subproblems to facilitate the problem-solving process.
- Define inputs and outputs in open-ended questions.
- Use flowcharts to address complex problems as algorithms.
- Create clearly named variables that represent different data types, and perform operations on their values.
- Design algorithms for a variety of tasks.

UNIT 2

Computational Thinking: Models and Simulations



Big Idea: Processes are made more efficient, effective, and optimized through careful analysis.

Skills

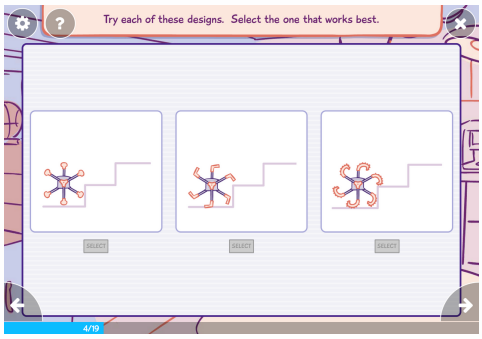
- Analyze flowcharts, including different process structures.
- Formulate data to optimize pattern recognition.
- Create hierarchies to represent the connections between the sub-problems.
- Document algorithms in order to make them easier to follow, test, and debug.

Continued on the next page.

UNIT 3

Computational Thinking: Implement and Test

Big Idea: Often there are many right answers; our job is to seek the best.



Skills

- Refine computational models based on the data they have generated.
- Systematically test and refine programs using a range of test cases.
- Develop functions to distill the algorithmic process.
- Analyze and compare different algorithms to find the least complicated process.
- Simplify algorithms in order to improve efficiency and/or make the process easier for a user to follow.

Bonus Resource

Free unplugged activity to help students practice algorithmic thinking and test their designs.

DOWNLOAD NOW

Skills Post-Check

Estimated Time: 15 Minutes