

Coding

In this demonstration, students will explore the fundamentals of coding. The presentation will cover concepts like loops and conditional statements. The activities will help students think like a programmer and will provide a solid introduction into the world of programming.

Grade-Specific Summary Table

Grade	Curriculum Focus	Demo Activity	Key Concept(s)
K	Computer Science: Order of Instructions	- Maze Solver (Faith activity) - Programmers Drawings	- Order in which instructions are followed affects outcome - Sequential thinking
1	Computer Science: Instructions & Outcomes	- Maze Solver - Loops activity - Conditional statements - Programmers Drawings	- Form of instructions may not affect outcome - Instructions ordered to produce desired outcome - Precise, sequential instructions
2	Computer Science: Problem Solving & Efficiency	All activities: - Maze optimization - Loops - Conditional statements - Decomposition (lunch menu) - Programmers Drawings	- Creativity and collaboration in design - Precise, reliable, efficient instructions - Simplifying by repeating steps (loops) - Breaking problems into parts

4+	Computer Science: Algorithms & Programming Concepts	All fundamental concepts: - Sequencing - Loops - Conditional statements - Decomposition Coding languages	- Algorithms and sequential logic Computational thinking Problem decomposition Programming syntax and languages
-----------	--	---	--

This table provides a quick overview of which demo slides/activities match Alberta curriculum for each grade.

Detailed Curriculum Links and Objectives by Grade

Kindergarten: Computer Science - Order of Instructions

Curriculum Link:

- The order in which instructions are followed can affect the outcome.

Objectives:

- Understand that following steps in order produces different results.
- Give and follow sequential instructions to solve a problem (navigate maze, draw images).
- Recognize that changing the order of steps changes the outcome.

Grade 1: Computer Science - Instructions & Outcomes

Curriculum Link:

- The form in which instructions are given may not affect the outcome (different methods, same result).
- Instructions are ordered in a way that will produce a desired outcome.

Objectives:

- Recognize that different instructions can achieve the same goal.
- Order instructions sequentially to reach a specific outcome.
- Introduce concept of loops (repeating actions) and conditions (if-then logic).
- Practice clear communication of instructions to others.

Grade 2: Computer Science - Problem Solving & Efficiency

Curriculum Link:

- Instructions are designed using creativity and problem solving, which can be enhanced through collaboration.
- Instructions can be created to be precise, reliable, and efficient to achieve the desired outcome.
- Instructions may be simplified by repeating steps (loops).

Objectives:

- Design efficient instructions by reducing unnecessary steps.
- Use loops to simplify repetitive tasks.
- Apply conditional statements to create organized solutions.
- Break complex problems into smaller, manageable parts (decomposition).
- Collaborate with peers to communicate instructions clearly and precisely.

Grade 4+: Computer Science - Algorithms & Programming Concepts

Curriculum Link:

- Introduction to fundamental programming concepts and computational thinking applicable to all coding languages.

Objectives:

- Master fundamental programming concepts: sequencing, loops, conditionals, decomposition.
- Understand algorithms as step-by-step problem-solving methods.
- Recognize that different coding languages achieve the same tasks with different syntax.
- Apply computational thinking to real-world problems.
- Gain familiarity with programming platforms like Scratch.

Cross-Curricular Connections (All Grades)

The coding demo reinforces skills across multiple subjects:

- Mathematics: Patterns, sequences, logical reasoning
- Language Arts: Clear communication, precise instruction writing
- Problem-Solving: Breaking down complex tasks, testing solutions
- Collaboration: Working together to achieve goals, peer instruction

Activity-Based Summary Table

Demo Activity	Alberta Curriculum Outcome	Key Concepts
What is Coding? Introduction	Understanding technology and instructions	Step-by-step instructions, coding applications
Maze Solver (Faith Activity)	Sequential instructions, efficiency	Ordered steps, optimization, reducing steps
Who Codes? Programmers	Career awareness, technology literacy	Programmer role, precise instructions
Loops Activity	Repetition, pattern recognition	Repeating instructions, identifying patterns
Conditional Statements (Fruit)	Logic, cause-and-effect thinking	If-then logic, rules and constraints
Decomposition (Lunch Menu)	Breaking down problems	Dividing complex problems, solving parts
Programmers Drawings	Communication, collaboration	Clear instructions, interpretation, teamwork
Coding Languages & Greetings	Syntax, language diversity	Different ways to express same idea, Scratch intro