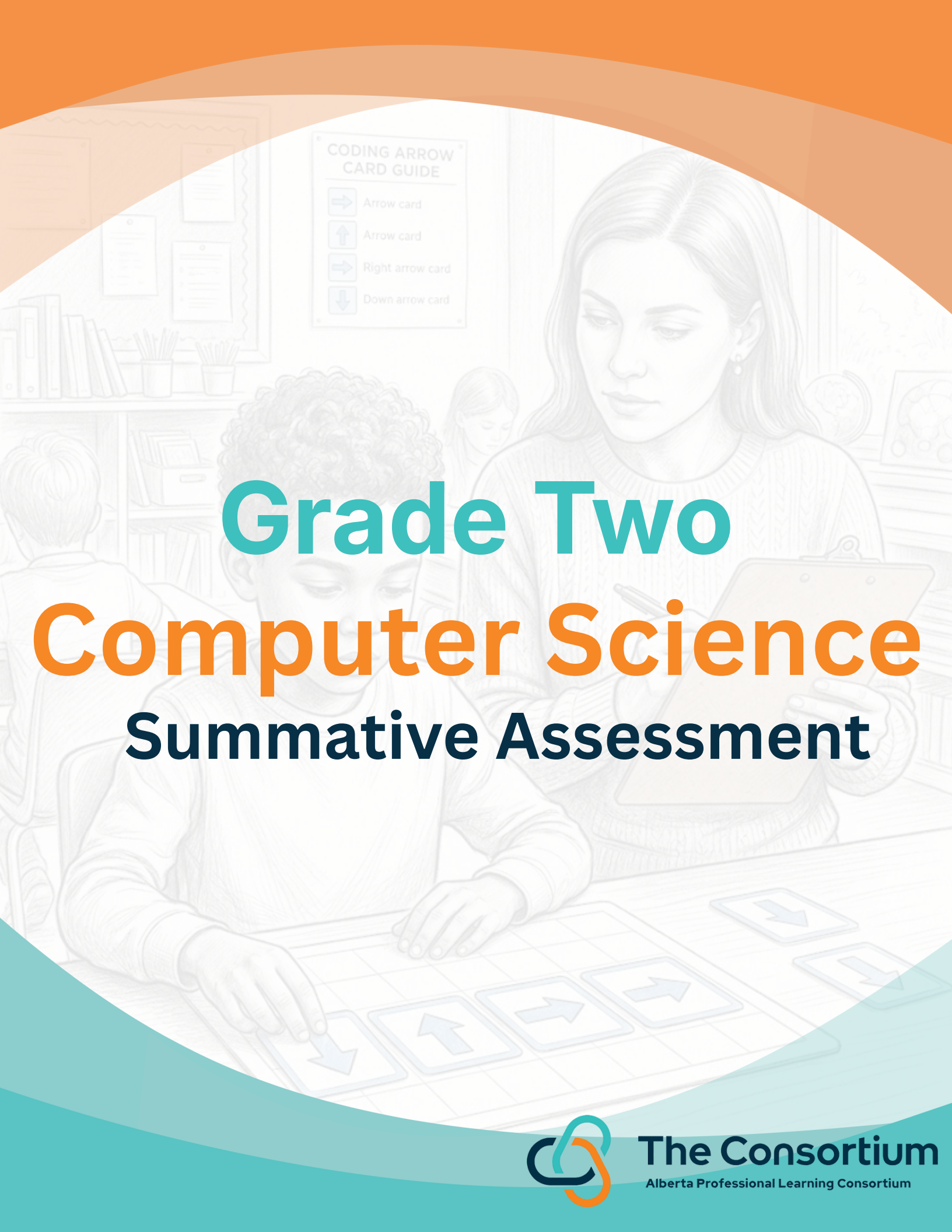


Grade Two Computer Science

Summative Assessment

The Consortium
Alberta Professional Learning Consortium

Grade 2 Science: Computer Science Observation Checklist

Student Name:

Date:

Skills	Met?	Teacher Notes / Evidence
Creativity & Design		
Identifies ways creativity is used to design instructions.	☐	
Uses creativity to generate original ideas or products.	☐	
Instruction Construction		
Works individually or in groups to create instructions using precise words, pictures, or diagrams.	☐	
Creates clear instructions consisting of three to four steps to achieve a desired outcome.	☐	
Analysis & Refinement		
Predicts the outcome of instructions that have three to four steps.	☐	
Tests instructions to verify that the desired outcome is achieved.	☐	

Skills	Met?	Teacher Notes / Evidence
Refines instructions to more efficiently achieve outcomes with less time, energy, or steps.	☐	
Problem Solving (Debugging)		
Identifies and removes errors (debugs) in a set of instructions.	☐	
Demonstrates how debugging increases the reliability of instructions.	☐	
Collaboration & Communication		
Exchanges ideas with others to design clear three- to four-step instructions.	☐	
Collaborates with peers to improve ideas and enhance problem solving.	☐	
Repetition & Logic		
Describes a situation in which repetition simplifies a set of instructions.	☐	
Designs instructions that include repeated steps to reach a goal.	☐	

Teacher Guide: Grade 2 Computer Science Summative Assessment

This guide is designed to assist teachers in administering the "Designing Instructions" assessment. It covers core computational thinking concepts including creativity, precise sequencing, collaboration, debugging, and repetition.

Part 1: Creativity and Design

Goal: To evaluate the student's ability to conceptualize an original goal and understand creativity in the context of design.

Teacher Instruction

Before starting, review the definition of creativity: making something original. Encourage students to think of a simple, achievable goal for their "robot," such as moving in a specific shape (square, triangle) or performing a simple action (clapping, spinning).

Part 2: Building Your Instructions

Goal: Demonstrate the ability to create precise, sequential instructions with a defined beginning and end.

Teacher Instruction

Ensure students include both "Start" and "Stop" markers to define the boundaries of their algorithm. Steps should use precise language (e.g., "Walk three steps forward" instead of "Go over there").

Step	Expected Components	Correctness Check
Start	Beginning marker	
1 - 4	Precise words & drawings	
Stop	Ending marker	

Part 3: Predict and Test

Goal: Foster collaboration and evaluate the ability to predict outcomes and test for reliability.

Teacher Instruction

Pair students together. One student acts as the "programmer" and the other as the "robot".

Predict: The partner should look at the steps and guess the outcome before performing them.
Test: The partner must follow the instructions *exactly* as written, without using outside knowledge. If the goal is reached, the instructions are "reliable".

Part 4: Debugging and Refining

Goal: Identify errors (bugs) and improve the efficiency of instructions.

Teacher Instruction

Explain that debugging is a normal part of computer science—it means finding and fixing errors.

Debugging: If the test failed, students should identify the specific step that was incorrect.
Efficiency: Ask students if any steps could be combined or simplified to "save energy" or make the robot faster.

Part 5: Repetition

Goal: Understand how to simplify instructions using loops or repetition.

Teacher Instruction

Provide the scenario of a robot clapping 10 times.

Correct Solution: Instead of writing "Clap" 10 times, a student might write "Repeat 10 times: Clap" or "Clap x10".

Scoring Rubric

Skill Observed	Mastery (3)	Progressing (2)	Emerging (1)
Creativity /Goal	Clearly defines an original, achievable goal.	Defines a goal that is somewhat unclear.	Unable to define a goal.
Sequencing	3-4 steps are logical, precise, and include Start/Stop.	Steps are logical but lack precision or Start/Stop.	Steps are out of order or missing.
Collaboration	Successfully predicts and tests with a partner.	Participates in testing but struggles with prediction.	Unable to work with a partner.
Debugging	Correctly identifies errors and suggests efficiency improvements.	Identifies an error but cannot suggest a fix.	Does not recognize errors.
Repetition	Provides a simplified, accurate instruction for repeating tasks.	Understands the concept but the instruction is slightly unclear.	Unable to simplify the instruction.

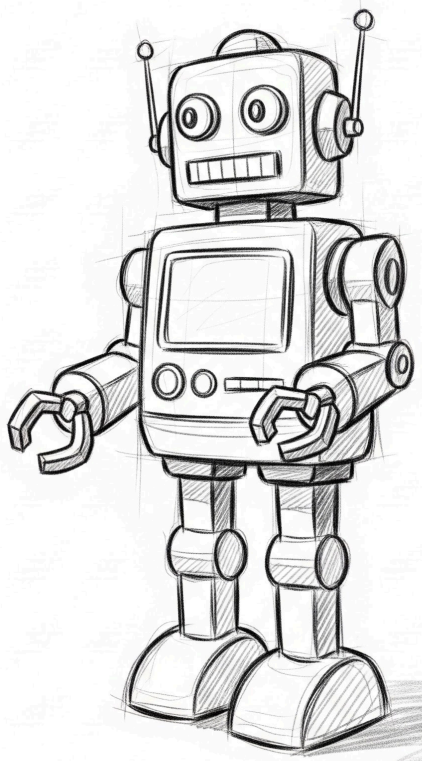
Name: _____

Computer Science: Designing Instructions

Part 1: Creativity and Design

Creativity means making something original, like a new game, recipe, or set of instructions. Think of a simple goal for a robot, like making a square or doing a dance.

What is your robot's goal?



Name: _____

Part 2: Building Your Instructions

Create a set of three to four steps to help your robot reach its goal. Use precise words and draw a picture for each step. Remember to include a Start and Stop.

Step	What to do (Words)	What it looks like (Drawing)
Start		
1		
2		
3		
4		
Stop		

Name: _____

Part 3: Predict and Test

Show your instructions to a partner (Collaboration!).

Partner Name:

1. Predict: What does your partner think will happen when they follow your steps?

---

2. Test: Let your partner follow your instructions exactly. Did they reach the goal? (Circle one)
   - ☐ Yes! It was reliable.
   - ☐ No. It needs debugging.

Part 4: Debugging and Refining

Debugging means finding and fixing errors in your instructions.

1. Did you find any errors? If so, what was the mistake?

---

2. **Efficiency:** Can you make your instructions shorter or faster to save energy?

Part 5: Repetition

Sometimes we do the same thing over and over. This is called repetition.

Scenario: You want your robot to clap 10 times.

Instead of writing "Clap" 10 times, how could you write a simpler instruction?
