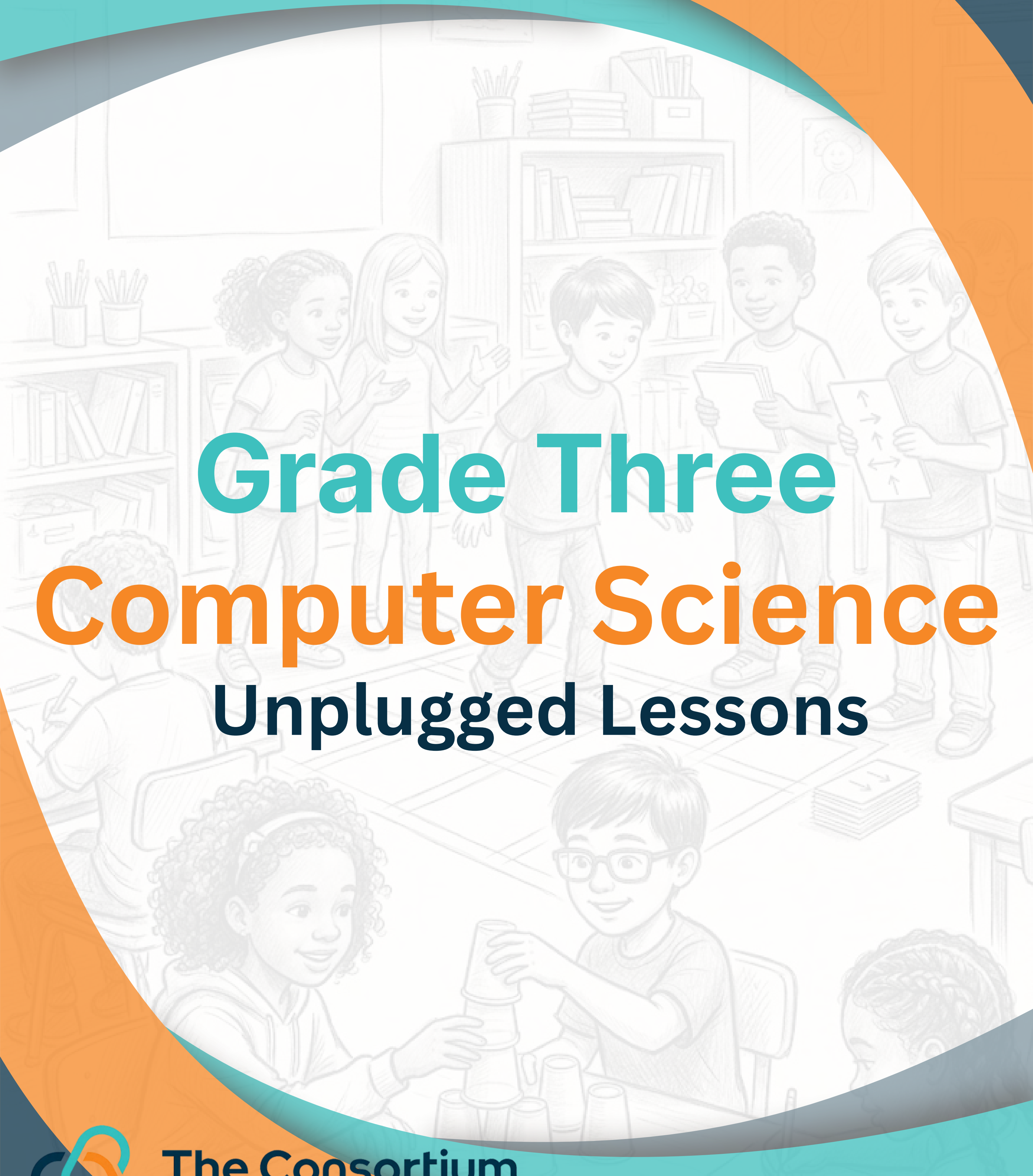


Grade Three Computer Science

Unplugged Lessons

The Consortium
Alberta Professional Learning Consortium

Grade 3 CS Unit Plan

Unit Overview: The Creative Problem Solvers

Organizing Idea: Computer Science: Problem solving and scientific inquiry are developed through the knowledgeable application of creativity, design, and computational thinking.

Guiding Question: How does creativity contribute to computational thinking?

Learning Outcome: Students investigate creativity and its relationship to computational thinking.

Lesson 1: Deconstructing the Grid (Introduction to Computational Thinking)

Focus: Understanding that computational thinking includes breaking a task into smaller chunks, finding patterns, identifying important details, and designing instructions.

Summary: Students will explore how computational thinking is a problem-solving process that uses creativity. Using a large layout on the board or floor, the class will take a big task (navigating a complex path to clear a workspace) and break it into smaller chunks. Students will create a set of instructions that could be followed by a human or a machine to complete a task. They will wrap up by working backwards if a mistake is made to fix their instructions.

Assessment: Students create a set of instructions to complete a task and identify the computational thinking elements used.

Lesson 2: Divergent Grids (Same Outcome, Different Paths)

Focus: Exploring how the same outcome can be achieved in different ways through divergent thinking.

Summary: Students will use 6x6 tabletop grids and arrow cards. Given a single starting point and a single final destination, students will learn that divergent thinking is the process of generating multiple unique ideas or solutions. Working in pairs, they will collaborate to write two different sets of instructions that achieve the same outcome (e.g., Route A focuses on the fewest turns, while Route B weaves around to collect tokens).

Assessment: Pairs successfully collaborate to write two different sets of instructions that achieve the exact same outcome.

Grade 3 CS Unit Plan

Lesson 3: Pixel Art Programming (Combining Ideas to Create Something New)

Focus: Understanding that creativity involves combining, changing, or reapplying existing ideas to produce something new.

Summary: Students will apply computational thinking to a screen-free "pixel coding" matrix (a grid of blank squares). The teacher will introduce how numbers and color codes are combined to make digital graphics. Students will create something new by combining, changing, or reapplying existing ideas, writing a binary or color-by-number sequence of instructions that a classmate must decode to reveal a hidden pixel image.

Assessment: Observation of students successfully coding and decoding pixel patterns to create new visual imagery.

Lesson 4: The Space Innovation (Engineering & Creativity)

Focus: Relating creativity to engineering, computing, and the development of new technologies, and examining a Canadian invention.

Summary: Students will learn that creativity is an important part of computer science, technology, and engineering. The class will examine a famous Canadian invention: the Canadarm. Using a specialized grid simulation, students will practice writing instructions to control a model robotic arm, identifying examples of creativity in computer science, technology, or engineering.

Assessment: Students successfully relate creative problem-solving to the real-world operation of the Canadarm and identify examples of tech creativity.

Lesson 5: The Inventors' Studio (The Computational Process)

Focus: Using imagination, observation, and making connections to solve a problem with computational thinking.

Summary: In this final lesson, students will look around the school or community to find a practical problem. They will use computational thinking to outline a solution—communicating how humans use this thinking to converse with computers more efficiently (like apps or robotics). Working in groups, they will design a master instruction blueprint that combines grid movements and pixel interfaces to act out their original technology concept.

Assessment: Students present their creative tech concept, detailing the specific instructions and computational steps used to solve the problem.

Grade 3 CS Lesson 1

Lesson Title: Deconstructing the Grid
Subject Area: Science (Computer Science)
Grade Level: Grade 3
Duration: 50–60 minutes

Lesson Overview

This lesson formally introduces the concept of computational thinking as a creative problem-solving process. Grade 3 students will learn that big problems can be solved efficiently by breaking them down into smaller chunks, identifying patterns, and abstracting away unimportant details. Using a large floor layout, students will design a master set of instructions to clear a workspace, and practice working backwards to fix mistakes if the instructions fail.

Learning Objectives: By the end of the lesson, students will be able to:

- [ ] Define computational thinking as a problem-solving process that uses creativity.
- [ ] Break a large grid navigation task into smaller, manageable chunks.
- [ ] Create a set of instructions that could be followed by a human or a machine to complete a task.
- [ ] Identify errors and work backwards if a mistake is made to fix the instruction set.

Essential Questions:

★ How does creativity contribute to computational thinking?
★ What steps do our brains take to break down a huge problem?

Materials & Resources

Category	Item Description
Classroom Tools	• Whiteboard and markers. • Large floor grid (6x6) marked with tape, featuring scattered board game tokens or toy items to "clean up."
Technology	None required (Unplugged computational thinking activity).
Files/Handouts	"Deconstruction Worksheet" divided into sections for Chunks, Patterns, and Final Instructions.

Grade 3 CS Lesson 1

Instructional Procedure

5–10 min | The Hook

Activity Description: The Messy Desk Dilemma.

Teacher Instructions: Stand in front of a completely disorganized table piled high with books, writing tools, tokens, and blocks. Act overwhelmed and dramatic.

Teacher Script: > *"Oh my goodness, look at this disaster area! My boss asked me to clean this up, organize the blocks, stack the books, and put away the game tokens perfectly. It is way too big of a job. My brain is stuck. I can't do it all at once! Help me, programmers! How can I make this huge problem feel smaller so I don't give up? What should I do first? (Allow students to suggest picking up just one type of item, like blocks)."*

Connecting to Concept: > "You just helped me use **computational thinking**! Computational thinking is a problem-solving process that uses creativity. Instead of panicking at a massive project, your creative brains told me to do something smart: break a task into smaller chunks! Today, we are going to learn how to break down big problems like real software engineers."

15–20 min | Direct Instruction

Activity Description: Deconstructing the Big Four of Computational Thinking.

Teacher Instructions: Gather students around the large 6x6 floor grid. On the board, write out the four parts of computational thinking as you explain them.

Teacher Script:

"When computer scientists look at a huge problem, they use four secret strategies to think like a computer:

1. **Breaking a task into smaller chunks:** We take a big grid path and focus on just reaching the first item. 2. **Finding patterns and similarities:** We look to see if our instructions repeat, like 'Move Forward, Move Forward'. 3. **Identifying the important details:** We ignore the color of the grid lines because that doesn't matter; we only focus on where the obstacles are. 4. **Designing instructions:** We write out a final code that can be followed by a human or a machine to complete the task.

Let's look at this floor grid. The robot's big job is to move across the grid, pick up the token, and carry it to the drop box. Let's break it down into chunks. Chunk 1: Move from the start square to the token. Chunk 2: Pick up the token. Chunk 3: Navigate to the drop box. See? Now it's just three small, easy jobs instead of one scary one!"

Grade 3 CS Lesson 1

20–30 min | Guided Exploration

Activity Description: Master Blueprint Mapping & Working Backwards.

Teacher Instructions:

1. Pair students up and distribute the "Deconstruction Worksheet" and a miniature 6x6 paper grid matching the floor layout.
2. Have students look at the target layout. Together, pairs must list the "chunks" of the task on their worksheet.
3. Next, they must work together to design and write out a set of instructions that could be followed by a human or a machine to complete the task (e.g., "1. Move Forward 3 spaces. 2. Grab token. 3. Turn left. 4. Move Forward 2 spaces").
4. **The Twist:** Have one student physically walk the floor grid to test their partner's code. Intentionally tell one partner to execute a step incorrectly to simulate a system error.
5. Instruct the programming partner to practice working backwards if a mistake is made, tracing their written lines from the crash point back to the source of the error to fix it.

Teacher Script:

"Design teams, look at your museum grid. Your goal is to clear the floor. First, break it into smaller chunks on your paper. Then, create a set of instructions that could be followed by a human or a machine to complete a task. Write it out step-by-step. If your partner runs the code and makes a mistake or hits a block, don't restart from the beginning! Use your computational thinking: practice working backwards if a mistake is made to find out exactly where the sequence glitched!"

10 min | Closure

Activity Description: Deconstruction Reflection.

Teacher Instructions: Gather the class back in a circle around the main floor grid.

Teacher Script:

"You all did an outstanding job decomposing problems today! Who can tell me what computational thinking is? (Expected answer: A problem-solving process using creativity). When we had a giant path to program, how did we make it easier? (Expected answer: We broke it into smaller chunks). And what did we do when our code crashed? We didn't give up; we practiced working backwards if a mistake was made to find the bug! Tomorrow, we are going to look at how different programmers can solve the exact same grid problem in completely different, creative ways using divergent thinking!"

Grade 3 CS Lesson 1

Differentiation

Support/Scaffolding: Provide a structured template for the worksheet where the blocks for "Chunk 1," "Chunk 2," and "Chunk 3" are already visually color-coded to correspond with colored zones on their practice grids, helping students isolate information.

Extensions: Challenge advanced students to find a hidden pattern in their grid instructions (e.g., noticing that "Forward, Turn Right" happens three times) and write a shorthand note about how that pattern could make their machine code run faster.

Assessment

Formative Assessment: Collect and review the "Deconstruction Worksheet" to verify that students were able to create a set of instructions that could be followed by a human or a machine to complete a task. Observe the pairs during the testing phase to see if they can identify computational thinking components and successfully work backwards if a mistake is made.

Name: ___________

Grid Architect: Deconstruction

🔍 Activity 1: The Big Break Down

Computer scientists solve big problems by breaking them into smaller **chunks**. Match the "Big Problem" on the left to the first small "Chunk" that helps solve it.

The Big Problem	The First Small Chunk
Cleaning a massive, messy desk	🧱 Find all the blue 4×4 bricks
Navigating a 6×6 floor grid	📦 Pick up just the board game tokens first
Building a giant Lego castle	🗺️ Move from the start to the first item

⚙️ Activity 2: Designing the Blueprint

Look at your floor grid. Your goal is to move across the grid, pick up the token, and reach the drop box. Write a set of instructions that a human or a machine could follow.

Chunk 1: Reach the Token

Step 1	
Step 2	
Step 3	
Step 4	

Name: ___________

Grid Architect: Deconstruction

Chunk 2: Reach the Drop Box

Step 1	
Step 2	
Step 3	
Step 4	

🛠️ Activity 3: The Debugging Challenge

Instructions don't always work the first time! If your partner hits a block, you must work backwards to find the mistake.

Look at the "Glitched" code below. Tracing backwards from the last step, circle the instruction that caused the error.

Step Number	Instruction	Status
1	Move Forward 2	✅ OK
2	Turn Left	❌ ERROR (Should be Right)
3	Move Forward 1	💥 CRASH

Name: ____________

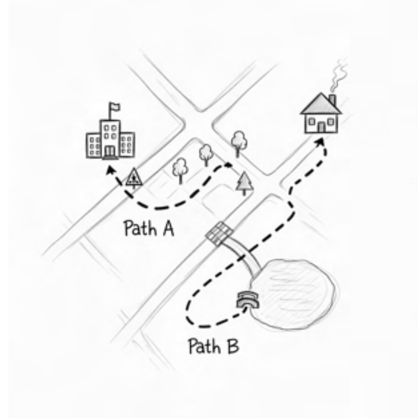
Grid Architect: Deconstruction

Activity 4: Creativity in Coding

Computational thinking is a problem-solving process that uses **creativity**. Draw a new obstacle for your grid and explain how you would "chunk" the path to get around it.

My New Grid Obstacle Drawing

Grade 3 CS Lesson 2



Lesson Title: Divergent Grids: Same Outcome, Different Paths
Subject Area: Science (Computer Science)
Grade Level: Grade 3
Duration: 50–60 minutes

Lesson Overview

This lesson introduces students to divergent thinking—the creative process of generating multiple unique solutions to a single problem. Using 6x6 tabletop grids and directional arrow cards, students will discover that the exact same computational outcome can be achieved in entirely different ways. Working collaboratively, pairs will design two distinct instruction sets to safely navigate a token to the same destination.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Define divergent thinking as the process of generating multiple unique solutions.
- ☐ Explain how creativity involves using imagination to find different ways to achieve a goal.
- ☐ Collaborate to write two different sets of instructions that achieve the same outcome.

Essential Questions:

- ★ How does creativity contribute to computational thinking?
- ★ If two programs look completely different but reach the same goal, which one is right?

Materials & Resources

Category	Item Description
Classroom Tools	• Whiteboard and markers. • Small board game tokens or toy cars (one per pair). • Standard 6-sided dice (to place random "bonus tokens" on the grid).
Technology	None required (Unplugged activity).
Files/Handouts	• Blank 6x6 grid papers (one per pair). • Sets of Arrow Coding Cards. • "Divergent Path" recording sheet (with columns for Route A and Route B).

Grade 3 CS Lesson 2

Instructional Procedure

5–10 min | The Hook

Activity Description: The Morning Route Challenge.

Teacher Instructions: Draw a simple map on the whiteboard showing the school in the center, surrounded by three different houses.

Teacher Script: > "Good morning, programmers! Think about how you got to school this morning. Did every single person in this room walk out their front door, take the exact same streets, and turn at the exact same corners? No! Some of you came by bus, some walked a short path, and some drove down the main road. But what was the final result? We all arrived at school! The same outcome can be achieved in different ways. Today, we are going to learn how computer scientists use **divergent thinking**—which means generating multiple unique ideas or solutions—to solve grid problems creatively!"

15–20 min | Direct Instruction

Activity Description: Defining Divergent Paths on a Grid.

Teacher Instructions: Model navigating a token across a grid under a document camera or on a drawn board grid. Show two distinct paths to reach the same star target.

Teacher Script:

*"Creativity involves imagination, observation, and making connections. In computer programming, there is rarely just one single 'correct' way to write code.

Look at our 6x6 grid. Our token starts at the bottom left, and our target is at the top right.

• **Path 1 (The Speedline):** I can use a direct sequence: move forward all the way, then turn, and move straight to the target.

• **Path 2 (The Scenic Route):** What if I use my imagination to weave around the middle squares instead?

Let's look at the instructions. Route A reads: Forward, Forward, Turn Right, Forward. Route B reads: Turn Right, Forward, Turn Left, Forward, Forward. Do they look the same? No. But do they achieve the exact same outcome? Yes! Both programs are successful because creativity allows us to develop different ways to solve a problem."*

Grade 3 CS Lesson 2

20–30 min | Guided Exploration

Activity Description: The Split-Route Collaboration.

Teacher Instructions:

1. Place students into pairs. Distribute a 6x6 paper grid, a toy car/token, arrow cards, and the recording sheet.
2. Have all pairs place their token at the bottom-left square ("Start") and a target card at the top-right square ("Goal").
3. **The Divergent Task:** The pair must collaborate to write two different sets of instructions that achieve the same outcome.
4. **Route A Challenge:** Together, they must design a path that gets the token to the goal using the *fewest possible arrow cards* (the most direct route). They write this sequence down in the Route A column.
5. **Route B Challenge:** Together, they must roll a die to place three "bonus points" (like counting squares to drop coins) onto the grid. They must use their imagination to create a completely new, winding sequence of instructions that collects the items but still lands on the exact same final goal square. They record this in the Route B column.

Teacher Script:

"Design teams, it's time to unleash your divergent thinking! Your goal is set, but you cannot just give me one answer. You must collaborate to write two different sets of instructions that achieve the same outcome. Route A needs to be a fast, straight shot. For Route B, reapply your existing ideas to create something completely original that snakes across the board to gather your items. Both paths must safely hit the exact same goal!"

10 min | Closure

Activity Description: Code Swap & Debrief.

Teacher Instructions: Have pairs trade their "Divergent Path" sheets with another nearby pair.

Teacher Script:

"Look at the sheet your neighbors gave you. Test their Route A and Route B code on your grid. Did they look like yours? Probably not! Who can remind us, what is divergent thinking? (Expected answer: Generating multiple unique solutions to a problem). In computer science, we use our creativity to find different, innovative ways to reach our goals. Tomorrow, we are going to pivot from grid paths into the creative world of digital graphics by learning how to code individual pixels!"

Grade 3 CS Lesson 2

Differentiation

Support/Scaffolding: For pairs struggling to visualize a second route, give them a physical barrier block to place right in the middle of their Route A path. This concrete limitation forces them to generate a unique alternative route around it.

Extensions: Challenge advanced pairs to analyze their two completed codes and determine the mathematical efficiency of each, identifying which sequence uses fewer command chunks to communicate with the machine.

Assessment

Formative Assessment: Monitor pairs as they fill out their recording sheets. The primary assessment criteria is confirming that students can successfully collaborate to write two different sets of instructions that achieve the same outcome. Check that Route A and Route B feature distinct, unique steps but result in the exact same final token placement.

Divergent Paths: Same Outcome, Different Paths

Name: Person
Date: Date

🔍 Activity 1: Think Like a Programmer

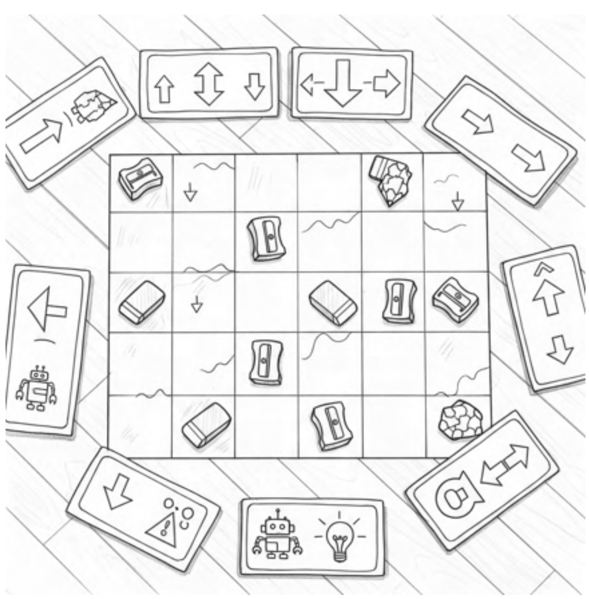
Computer scientists use different ways to solve problems. Match the term on the left to its correct definition on the right.

Key Term	Definition
Divergent Thinking	Using imagination and observation to find new ways to solve a problem.
Creativity	A single result or goal reached by a set of instructions.
Outcome	The process of generating multiple unique ideas or solutions.

Activity 2: Divergent Path Recording Sheet

Work with your partner to design two different paths for your token. Both paths must start at the bottom-left and end at the top-right.

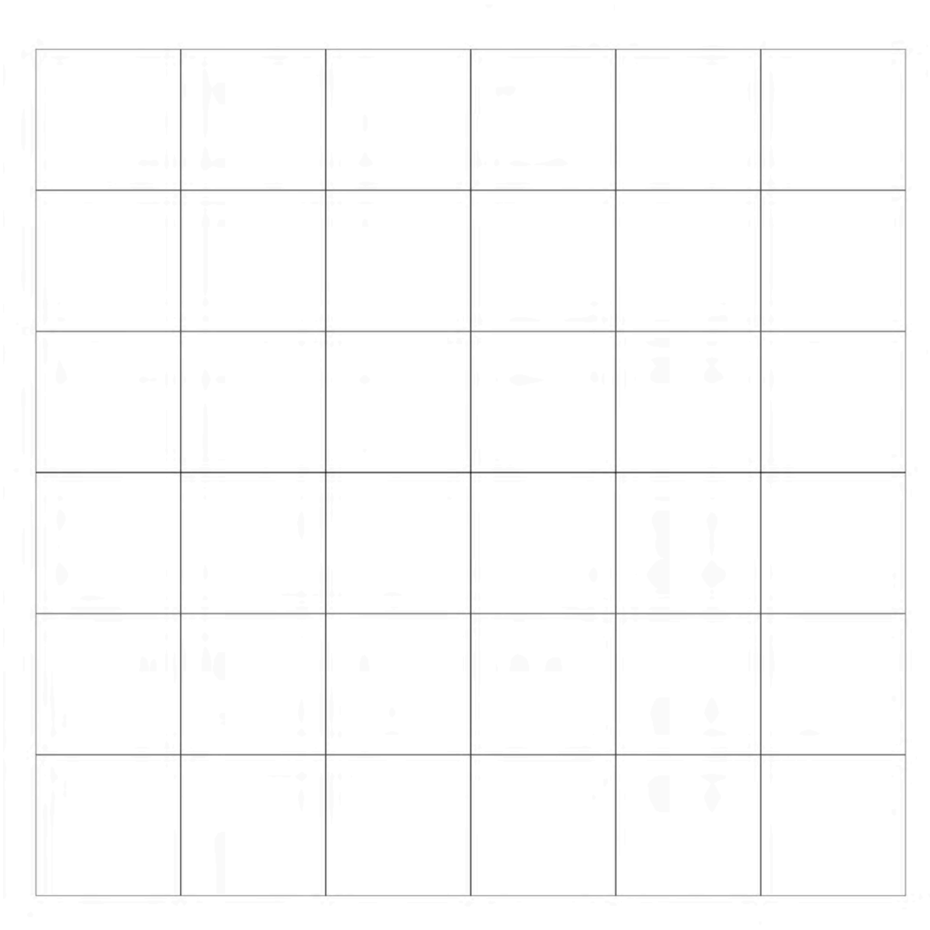
Route A (The Speedline)	Route B (The Scenic Route)
Goal: Use the fewest cards possible.	**Goal:** Collect 3 "bonus points".
1.	1.
2.	2.
3.	3.
4.	4.
5.	5.
6.	6.



🎨 Activity 3: Reflection

In the box below, draw a quick sketch of your Route B. How did your team use creativity to reach the goal in a different way than your neighbors?

My Divergent Route Sketch



Reflection Question: If two programs look completely different but reach the same goal, which one is "right"? Why?

Answer:

Grade 3 CS Lesson 3

Lesson Title: Pixel Art Programming
Subject Area: Science (Computer Science)
Grade Level: Grade 3
Duration: 50–60 minutes

Lesson Overview

In this lesson, students discover that creativity involves combining, changing, or reapplying existing ideas to produce something new. Students will reapply their grid navigation skills to a new context: pixel arrays. They will learn how computers use simple binary instructions (grids of filled vs. empty blocks) to build digital art and display screens. Working in pairs, students will write a set of pixel instructions to encode a custom design, which a classmate must decode to reveal a hidden image.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Explain how creativity involves combining or reapplying existing ideas to produce something new.
- ☐ Create a set of pixel instructions that can be followed by a human to complete a task.
- ☐ Identify how computational thinking is used to create virtual graphics and apps.

Essential Questions:

- ★ How can we combine the ideas of a grid and a picture to communicate with a computer?
- ★ How does a computer turn simple numbers into beautiful screen images?

Materials & Resources

Category	Item Description
Classroom Tools	• Whiteboard and colorful dry-erase markers. • Large visual example of a pixelated heart or star.
Technology	None required (Unplugged pixel matrix activity).
Files/Handouts	• "Pixel Code Matrix" worksheet (featuring a blank 6x6 square grid). • "Programmer's Code Sheet" (blank lines for writing down coordinate steps).

Grade 3 CS Lesson 3

Instructional Procedure

5–10 min | The Hook

Activity Description: The Mega Zoom Zoom.

Teacher Instructions: Project or hold up a very close-up, highly pixelated image of a classic video game character or icon (like a heart or a coin). Ask students if they can guess what it is. Slowly pull the image back until they recognize it.

Teacher Script: > "Look closely at this poster, programmers! It just looks like a bunch of colored squares block by block. But when I pull it back... wow! It's a heart! Computers don't see images the way we do. They break pictures down into tiny squares called **pixels**. Computational thinking can be used by humans to communicate with computers more efficiently to build our favorite apps and video games. Today, we are going to combine our grid math skills with our imagination to create something brand new: our very own pixel code!"

15–20 min | Direct Instruction

Activity Description: Coding by Grid Coordinates.

Teacher Instructions: Draw a simple 4x4 matrix on the whiteboard. Label the columns A, B, C, D and the rows 1, 2, 3, 4. Explain that creativity involves combining existing ideas to make something new. Show how to write a code that tells a human exactly which squares to color in.

Teacher Script:

*"Creativity involves combining, changing, or reapplying existing ideas to produce something new. We already know how to use arrow cards to walk across a grid[cite: 2, 4]. Today, we are going to reapply that grid idea to make art!

Look at this board. If I want a machine to paint a stripe, I have to design instructions. Instead of arrows, I will give coordinates. If my code says, 'Color in A1, B2, C3, D4,' watch what happens when I follow it. (Color in the diagonal line). Look at that! We just created a diagonal line graphic using nothing but simple grid steps. We are creating a set of instructions that could be followed by a human or a machine to complete a task!"*

Grade 3 CS Lesson 3

20–30 min | Guided Exploration

Activity Description: The Pixel Exchange Studio.

Teacher Instructions:

1. Pair students up. Distribute the "Pixel Code Matrix" worksheet and the code sheet.
2. **Phase 1 (Encode):** Student A (the Artist) secretly colors in a simple design on their 6x6 grid (e.g., a smiley face, a letter, or a sword). On their code sheet, they write down the instructions step-by-step by listing the coordinates that need to be filled (e.g., "Step 1: Color A2. Step 2: Color A3.").
3. **Phase 2 (Decode):** Student A folds their drawing backward so it is hidden and hands only the written set of instructions to Student B.
4. Student B acts as the computer terminal. They take a fresh, blank grid and follow the instructions precisely, coloring in the blocks as commanded.
5. Once complete, they unfold the original drawing to verify if their final outcome matches the artist's original vision perfectly.
6. Swap roles so Student B becomes the encoder.

Teacher Script:

"Artists and programmers, it's time to create! Secretly design a brand new shape on your grid. Then, convert your drawing into clear code instructions. Remember, you are designing instructions for a machine—your partner cannot guess! They must look only at your coordinates. Let's see if your code is clear enough to reproduce your digital masterpiece!"

10 min | Closure

Activity Description: Graphic Showcase & Connections.

Teacher Instructions: Have students display their matching coded pictures on their desks.

Teacher Script:

"Look around the room! You combined your math grid logic with your artistic imagination to produce something completely new. This is exactly how game designers make virtual reality, graphics, and mobile phone apps function behind the scenes! Who found a bug in their coordinate code where their partner accidentally colored the wrong block? (Allow sharing). How did you fix it? You worked backwards to check the numbers! Tomorrow, we are going to look at a massive, world-famous invention built by creative Canadian engineers that uses these exact same kinds of coordinates to move things out in space: the Canadarm!"

Grade 3 CS Lesson 3

Differentiation

Support/Scaffolding: For students struggling to translate coordinates, use a simpler binary coding sheet instead of letters and numbers (e.g., Row 1: 0, 1, 1, 0, where 0 means "leave white" and 1 means "color in black"), lowering the cognitive load of tracking two axes at once.

Extensions: Challenge advanced pairs to create a two-color pixel code. They must add a specific color verb indicator to their instruction sequence (e.g., "Step 1: Color B4 Red. Step 2: Color C4 Blue."), making the program matrix more complex.

Assessment

Formative Assessment: Monitor the pairs closely during Phase 2 of the Guided Exploration. Check the "Programmer"s Code Sheets" to ensure students are successfully creating a set of instructions that can be followed by a human to complete a task. Check if the decoded matrix matches the encoded matrix to verify the precision of their computational steps.

Name: ______________

Pixel Art Programming

🎨 Activity 1 — The Artist (Encode)

Secretly design a simple shape or letter by coloring in the squares on the grid below. Do not show your partner yet!

Name: ______________

Pixel Art Programming

💻 Activity 2 — The Programmer (Decode)

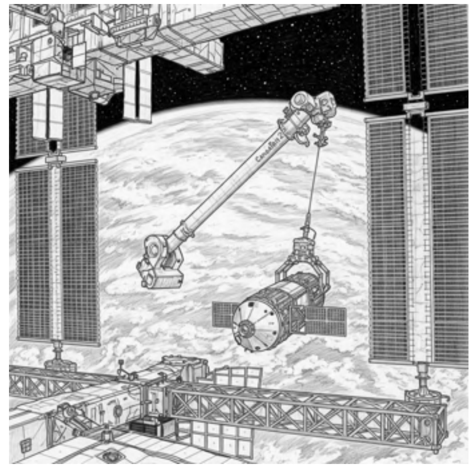
Now, convert your drawing into code instructions. Start in the upper right corner and list the number of squares to color followed by a block with the color filled in.

eg: 2⬛3⬜
1⬛2⬜ 2⬛

My Pixel Code Instructions:

📝 Activity 4: Reflection

How does a computer turn simple numbers into beautiful screen images?



Grade 3 CS Lesson 4

Lesson Title: The Space Innovation
Subject Area: Science (Computer Science)
Grade Level: Grade 3
Duration: 50–60 minutes

Lesson Overview

In this lesson, students explore how creativity is an important part of computer science, technology, and engineering by examining a major Canadian invention: the Canadarm. Students will relate creativity to engineering, computing, and the development of new technologies by acting as aerospace programmers. Using an unplugged grid simulation, they will design a set of instructions to control a robotic arm model, moving a space payload safely from a starting point to a docking terminal.

Learning Objectives: By the end of the lesson, students will be able to:

- [ ] Relate creativity to engineering, computing, and the development of new technologies.
- [ ] Examine a Canadian invention and identify examples of creativity within it.
- [ ] Create a set of instructions that can be followed by a human to complete a mechanical task.

Essential Questions:

★ How does creativity contribute to computational thinking and real-world engineering?
★ How do scientists use precise code to control machines that are far away from them?

Materials & Resources

Category	Item Description
Classroom Tools	• Whiteboard and markers. • Split-pin brad fasteners and cardboard strips (to create simple 2-segment pivoting robotic arms). • Small plastic cups (representing the space payload).
Technology	None required (Unplugged robotics simulation).
Files /Handouts	• Image or short profile of the Canadarm on the International Space Station. • "Canadarm Sector Grid" sheet (a 6x6 grid with marked launch and docking coordinates).

Grade 3 CS Lesson 4

Instructional Procedure

5–10 min | The Hook

Activity Description: Reaching for the Stars.

Teacher Instructions: Stand a few feet away from a student. Hold your arm completely stiffly, moving only your shoulder and your elbow in mechanical, rigid lines to pick up a plastic cup.

Teacher Script: > *"Programmers, imagine you are an astronaut floating in outer space. You need to catch a floating supply ship, but it is too far away to reach with your hands. What do you do? In 1981, creative Canadian engineers solved this exact problem by inventing the* ***Canadarm****—a giant robotic arm that attached to space shuttles! Canadians are responsible for many creative inventions, such as the Canadarm. It used imagination and engineering to do things humans never could. Today, we are going to step into the shoes of those Canadian space engineers and write the code to operate our own robotic arm!"*

15–20 min | Direct Instruction

Activity Description: Connecting Creativity to Robotics.

Teacher Instructions: Display an image of the Canadarm. On the whiteboard, map out a 6x6 grid to show how mechanical joints require coordinate and directional instructions to work.

Teacher Script:
*"Creativity is an important part of computer science, technology, and engineering. Engineers don't just build parts; they have to use computational thinking to program how those parts move efficiently!

The Canadarm doesn't have a human brain. It needs computer programmers to design instructions that tell it exactly how to rotate its shoulder, bend its elbow, and close its gripper. We can identify examples of creativity in how these instructions are written to avoid hitting the space station!

Let's look at our robot command code variables for today:*

EXTEND [Number]: Moves the arm forward by that many grid blocks.*

ROTATE [Right/Left]: Pivots the arm joint.*

GRAB / DROP: Closes or opens the clamp.*

Let's practice creating a set of instructions that could be followed by a machine to capture a satellite at coordinate C5."

Grade 3 CS Lesson 4

20–30 min | Guided Exploration

Activity Description: The Canadarm Sector Simulation.

Teacher Instructions:

1. Group students into pairs. Distribute the "Canadarm Sector Grid", a small cardboard arm model put together with a brad fastener, and a plastic cup payload.
2. **The Space Setup:** Have students place their cardboard arm base at the bottom-center of the grid ("The Space Station"). Place the plastic cup at a specific grid coordinate on the upper half of the sheet ("The Satellite").
3. **The Engineering Task:** Student A is the Mission Controller (the Programmer). They must design a set of instructions using the robot command variables to move the arm from the base, grab the satellite, and bring it back safely. They write this code on their sheet.
4. Student B is the Canadarm (the Machine). They read the code text exactly as written and physically move the cardboard arm model to execute the tasks block by block.
5. If the arm overshoots or drops the payload, the partners must identify the error, work backwards, and rewrite the code to secure the payload safely.
6. Switch roles so each student gets to act as the programmer.

Teacher Script:

"Space engineers, your mission is live! Work with your partner to look at your target satellite coordinate. Create a set of instructions that can be followed by your machine partner to complete the task. Be precise with your commands! If your robotic arm misses the cup, trace your steps to figure out where the numbers went wrong, and debug it to save the mission!"

10 min | Closure

Activity Description: Mission Debrief & Engineering Connections.

Teacher Instructions: Gather the class back to the circle.

Teacher Script:

"Mission accomplished! You successfully related creativity to engineering, computing, and the development of new technologies today. By programming your model arms, you did the exact same type of work that Canadian space scientists did when they built the real Canadarm! Who can tell me an example of creativity we used in our computer science task today? (Expected answer: We had to invent a creative path to reach the coordinates without breaking the arm). Tomorrow, for our final lesson, you will become inventors in our Collaborative Studio, designing an app concept to solve a real-world problem!"

Grade 3 CS Lesson 4

Differentiation

Support/Scaffolding: Provide a visual layout of the robot commands on a reference sheet (e.g., an arrow pointing forward for "EXTEND", a curved arrow for "ROTATE") to assist students who have difficulty translating text commands into physical actions.

Extensions: Challenge advanced pairs to introduce space debris (an obstacle block) onto their grid. They must use divergent thinking to write an alternate route that navigates the Canadarm safely around the debris while maintaining efficiency.

Assessment

Formative Assessment: Observe pairs during the Guided Exploration. Check that students are writing a functional set of instructions that can be followed by a human or machine to complete a task. Review their recording sheets to confirm they are incorporating the correct command verbs (EXTEND, ROTATE, GRAB) alongside coordinate points.

The Canadarm: Canada’s Super Robot in Space!

🚀 What is the Canadarm?

Imagine you are an astronaut floating high above the Earth. You need to catch a supply ship to get your food and tools, but it is way too far away to reach with your hands. What do you do?

In 1981, creative Canadian engineers solved this problem by inventing the Canadarm. It was a giant, powerful robotic arm that attached to space shuttles. This invention showed the whole world how Canadians use imagination and engineering to do things humans never could on their own.

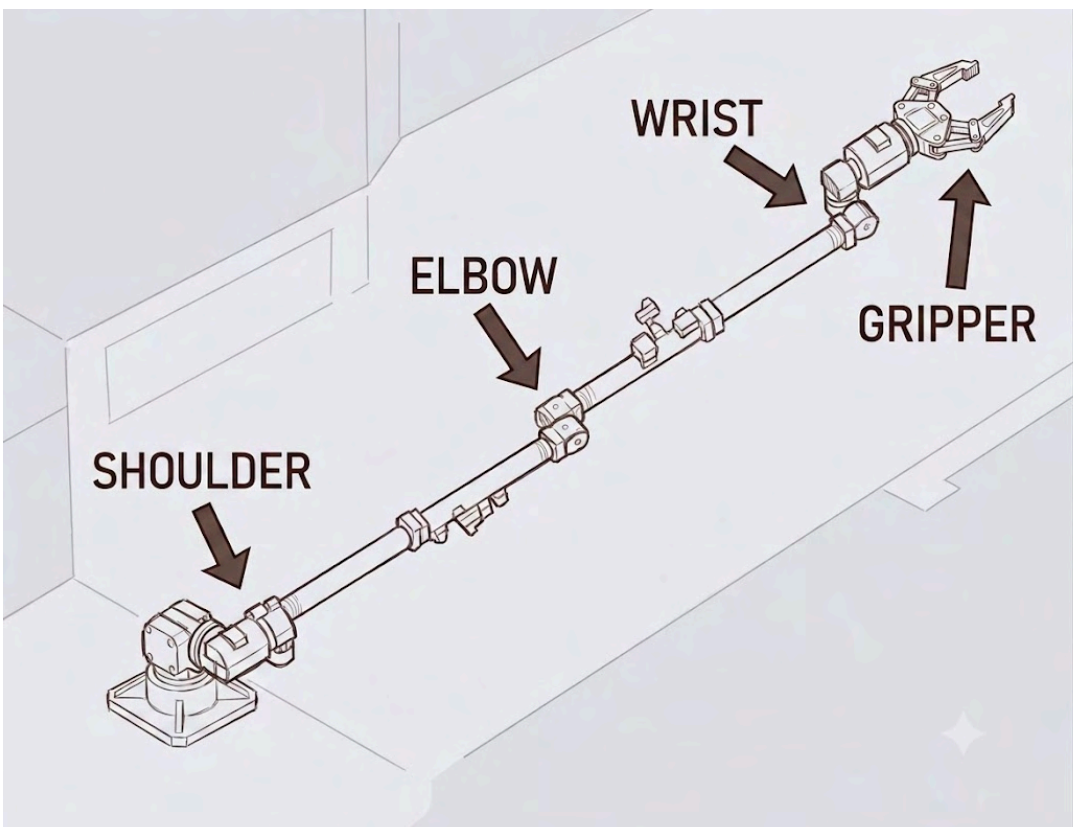
🛠️ How Does it Work?

The Canadarm doesn't have a human brain. It needs computer programmers to write special code that tells it exactly how to move. Because space is dangerous, these instructions have to be very precise so the arm doesn't hit the space station!

The Parts of the Arm

The Canadarm was designed to move just like a human arm, but much bigger and stronger.

Part Name	What it Does
Shoulder	Allows the arm to pivot and move in wide circles.
Elbow	Bends to reach around corners or obstacles.
Wrist	Twists and turns to get into the perfect position.
Gripper	The "hand" that grabs satellites or supply ships.



💡 Creativity and Computing

You might think engineering is just about math, but it actually takes a lot of **creativity**! Engineers use **computational thinking** to solve space problems.

- **Designing Paths:** Programmers had to invent creative ways to move the arm without breaking it.
- **Problem Solving:** If a satellite was spinning too fast, engineers had to think of new ways to catch it safely.
- **Building the Future:** The success of the first Canadarm led to the creation of **Canadarm2**, which now lives on the International Space Station.

⭐ Fast Facts for Space Explorers

- **Birthday:** The Canadarm first went into space on November 13, 1981.
- **Job Description:** It was used to move heavy "payloads"—which are like space cargo—and even to help astronauts move around during space walks.
- **Canadian Pride:** Because of this invention, Canada became a world leader in space robotics.
- **Debugging:** Just like you do in class, space engineers have to "work backwards" to fix mistakes in their code if something doesn't go right.

Mission Task: Look at the diagram of the Canadarm on your worksheet. Can you identify where the shoulder, elbow, and gripper are? Think like a programmer—how would you tell the arm to "EXTEND" to reach a target?

Name: ____________

Mission: Canadarm Innovation

🚀 Activity 1: Robot Command Match-Up

Draw a line from the **Command Variable** to the **Mechanical Action** it performs.

Command Variable	Mechanical Action
EXTEND [Number]	Pivots the arm joint to a new angle.
ROTATE [Right/Left]	Closes or opens the clamp to hold a payload.
GRAB / DROP	Moves the arm forward by a specific number of blocks.

🛠️ Activity 2: Coding the Mission

Instructions help space engineers control machines that are far away. Look at your Canadarm Sector Grid. Your target satellite is at coordinate **C5**. Circle the code sequence that shows the most creative and efficient path to the satellite.

Code Option A	Code Option B
1. EXTEND 3 2. ROTATE Right 3. EXTEND 2 4. GRAB	1. ROTATE Right 2. EXTEND 10 3. DROP 4. ROTATE Left

Name: ______________

Mission: Canadarm Innovation

🎨 Activity 3: Design Your Mission

Draw a picture of your Canadarm model moving the payload (plastic cup) from the launch point to the docking terminal.

Grade 3 CS Lesson 5

Lesson Title: Level Up! The Inventors' Studio
Subject Area: Science (Computer Science)
Grade Level: Grade 3
Duration: 50–60 minutes

Lesson Overview

In this final unit lesson, students investigate creativity and its relationship to computational thinking by stepping into the role of software inventors. They will learn how humans use computational thinking to communicate with computers more efficiently through things like apps, virtual reality, and robotics. Working in design groups, students will identify a school problem, break it into chunks, and design a master technology blueprint that combines a grid-based movement path and a pixelated user interface.

Learning Objectives: By the end of the lesson, students will be able to:

- ☐ Identify how computational thinking is used to solve problems or achieve desired outcomes.
- ☐ Create something new by combining, changing, or reapplying existing ideas.
- ☐ Create a set of instructions that could be followed by a human or a machine to complete a task.

Essential Questions:

- ★ How does creativity contribute to computational thinking?
- ★ How do apps and robots use grids and pixels to make human lives easier?

Materials & Resources

Category	Item Description
Classroom Tools	• Whiteboard and markers. • Board game tokens (to act as the app's user cursor).
Technology	None required (Unplugged app design studio).
Files/Handouts	• "Inventors' App Blueprint" packet (Page 1: 6x6 movement grid for a helper robot; Page 2: Blank 6x6 pixel matrix for the app icon design).

Grade 3 CS Lesson 5

Instructional Procedure

5–10 min | The Hook

Activity Description: The Convergence Brainstorm.

Teacher Instructions: Gather students on the carpet and write the word "INVENTION" in large letters on the whiteboard.

Teacher Script: > "Inventors, welcome to the grand finale of our unit! Think about your favorite video game, a cool tablet app, or even a robotic vacuum cleaner. Did those ideas just appear out of nowhere? No! Creative programmers looked at a real-world problem and used computational thinking to solve it. Computational thinking can be used by humans to communicate with computers more efficiently so we can build apps and robotics that help people every day. Today, you are entering the Inventors' Studio to design your very own technology concept!"

15–20 min | Direct Instruction

Activity Description: Combining Grids and Pixels.

Teacher Instructions: Explain that creativity involves combining, changing, or reapplying existing ideas to produce something new. Show how an app uses both navigation pathways (grids) and graphics (pixels) to work.

Teacher Script:

*"To make something completely new today, we are going to use the power of creativity by combining two ideas we've already mastered: grid navigation and pixel art coding!

Imagine we want to invent an app called 'The Library Helper'. Part 1 of our app needs to show a clean screen interface. We reapply our pixel coding to draw a book icon on our screen grid. Part 2 of our app needs to give instructions to a robot to pick up a book dropped on the floor. We reapply our grid path instructions to guide the machine cursor right to the target!

By putting these pieces together, we identify computational thinking used to solve problems and achieve desired outcomes. Let's design instructions to build our own app concepts!"*

Grade 3 CS Lesson 5

20–30 min | Guided Exploration

Activity Description: The App Innovation Challenge.

Teacher Instructions:

1. Group students into pairs or design teams of three. Distribute the "Inventors' App Blueprint" packets.
2. **Step 1: Identify the Problem.** Have teams choose a classroom helper concept (e.g., a recess toy organizer app, a pencil sharpening robot program, or a desk cleanup helper tool).
3. **Step 2: Reapply Pixel Coding.** Teams will use their imagination to design a custom pixel icon for their app on the 6x6 matrix, writing out the step-by-step coordinate instructions so a machine could load the graphic.
4. **Step 3: Reapply Grid Navigation.** On the 6x6 movement sheet, teams will place obstacles and a token target. They will create a set of instructions that could be followed by a machine to guide their helper token through the path to solve the problem.
5. If a line of code doesn't work during testing, teams must practice working backwards to fix the mistake.

Teacher Script:

"Design teams, it's time to build! Create something new by combining and reapplying your best grid and pixel ideas. Write down a precise set of instructions that could be followed by a human or a machine to make your invention work. Make sure your steps are clear, break your task into smaller chunks, and test your code carefully!"

10 min | Closure

Activity Description: The Inventors' Pitch.

Teacher Instructions: Have each group stand up and showcase their pixel art icon and grid instructions to the class.

Teacher Script:

"What an unbelievable studio showcase! You investigated creativity and its relationship to computational thinking by turning simple blank squares into functional app blueprints. Who can share how breaking a task into smaller chunks helped their team stay organized today? (Allow sharing). Remember, every piece of technology you use in the real world starts exactly like this: with a creative mind, a clear problem, and step-by-step logical instructions. Sensational work, Grade 3 engineers!"

Grade 3 CS Lesson 5

Differentiation

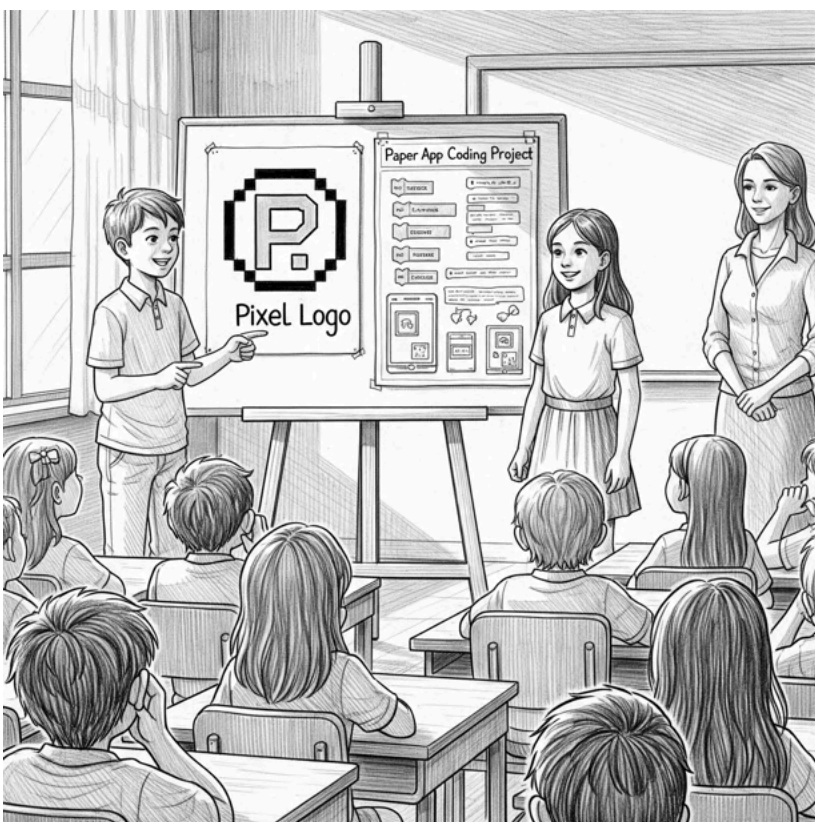
Support/Scaffolding: Provide a structured word bank with sentence starters on the blueprint packet (e.g., "Our app solves the problem of ______." "Step 1: Move ____ spaces.") to guide students through the deconstruction process.

Extensions: Challenge advanced groups to explain how their unplugged concept could relate to a real-world machine like the Canadian Canadarm, describing how a real astronaut might interact with an interface grid to operate a device.

Assessment

Formative Assessment: Observe the collaborative dynamics within teams during the design phase. Check that students are actively identifying computational thinking strategies to solve their chosen problem.

Summative Assessment: Review the final completed "Inventors" App Blueprint" packets. Verify that students successfully created something new by combining existing grid and pixel ideas, resulting in a cohesive, multi-step set of instructions designed to achieve a clear, helpful outcome.



Name: ______________

Level Up! The Inventors' Studio

💡 Activity 1: Identify the Problem

Inventors look at a real-world problem and use computational thinking to solve it. Circle the **classroom problem** your team wants to solve today:

Problem A	Problem B
✏️ Pencil Sharpening Helper	🧸 Recess Toy Organizer
Problem C	Problem D
🧹 Desk Cleanup Tool	📓 Library Book Return Robot

🎨 Activity 2: The App Icon (Pixel Coding)

Design a custom 6×6 pixel icon for your helper app. Write the step-by-step coordinate instructions so a machine could load your graphic.

Step	Colors eg: 1⬛2⬜ 1⬛2⬜ 1⬛
1	
2	
3	
4	
5	
6	

Name: ________________

Level Up! The Inventors' Studio

Space for your Pixel Drawing:

Name: ______________

Level Up! The Inventors' Studio

🤖 Activity 3: Program the Path (Grid Navigation)

Create a set of instructions to guide your helper robot through a grid to solve the problem. Avoid any obstacles you place on the map.

My Robot's Code:

Move	
Move	
Move	
Move	
Action