Pre-K Computer Science Guidance

Introduction

This document extends the 2023 Oklahoma Academic Standards for Computer Science (OAS-CS) by providing developmentally appropriate guidance for pre-kindergarten (PK). Each standard is paired with the following:

- A Computer Science Practice (mirroring the K–12 OAS-CS)
- A concise Unplugged Idea so teachers can implement the concept without devices
- A computational thinking connection to clarify the learning goal

What is Computer Science for Pre-K students?

Computer science is more than just using computers—it's about helping children learn how to think clearly, solve problems, and understand how things work.
In Pre-K, computer science doesn't mean coding or programming on a screen. Instead, it looks like:

- Giving step-by-step directions (like how to get ready for lunch)
- Sorting and organizing objects
- Recognizing patterns
- Playing games that involve rules and logic
- Exploring how tools and machines help us

Computer science education helps young learners build the thinking skills they'll need in school and in life. It's not just about using technology—it's about thinking like a computer scientist, even when there's no computer in sight.

Computer Science Practices

The Oklahoma Computer Science Practices describe the ways students explore, think, and solve problems like computer scientists. Even our youngest learners can begin developing these habits—like noticing patterns, following steps, and working with others to solve problems.
While the practices aren't tied to a specific grade level, they grow over time as students do. In Pre-K, this might look like sorting objects, giving simple instructions, or asking questions about how things work. These early experiences help children build the thinking skills they'll use later in computer science. Each practice connects with the OAS-CS to support students in learning big ideas in age-appropriate ways.

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Devices (D)	PK.CS.D.01 With guidance, explore turning a classroom device on and off and choosing a favorite activity using large picture icons.	Pretend a flashlight is a "computer;" flip the switch on/off to explore power states. Have children choose their favorite colored button to "start" different activities.	**Pattern Recognition**: Understanding that devices have consistent on/off states. **Abstraction**: Using simple symbols (buttons/switches) to control complex systems.
	Developing a Productive Computing Environment		
Hardware & Software (HS)	PK.CS.HS.01 With guidance, name basic parts of a classroom device (e.g., screen, button) using simple words such as "on button" and "picture."	Use a cardboard laptop with labeled sticky notes; match labels to pretend parts. Include "screen" (paper), "keys" (bottle caps), and "on button."	**Decomposition**: Breaking down complex devices into understandable parts. **Abstraction**: Using simple names for complex components.
	Developing and Using Abstractions		
Troubleshooting (T)	PK.CS.T.01 With guidance, seek help or try a simple fix (e.g., "tap again") when a device does not work as expected.	Role-play an "Oops Robot:" when the robot freezes (child stops moving), other children offer help or suggest "tap again" to restart.	**Debugging**: Identifying when something isn't working as expected. **Algorithm**: Following simple steps to solve problems.
	Testing and Refining Computational Artifacts		

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Network Communication & Organization (NCO)	PK.NI.NCO.01 Play a pretend messaging game to discover that toys or tablets can “talk” to one another when connected.	Talk through paper-cup telephones to show messages travelling on a “string network.” Practice sending simple messages like “Hello” back and forth.	**Systems Thinking:** Understanding that devices can connect and share information. **Abstraction:** Representing complex networks with simple physical models
	Communicating About Computing		
Cybersecurity (CY)	PK.NI.CY.01 With guidance, use a picture password or badge to log in and discuss keeping badges safe.	Play “badge check-in” at the classroom door; each child has a unique picture badge that stays in a safe box afterward. Discuss why we keep special things safe.	**Security Concepts:** Understanding that some things need protection. **Pattern Recognition:** Using unique identifiers (badges) for access.
	Developing a Productive Computing Environment		

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Storage (S)	PK.DA.S.01 With guidance, save a digital photo of their drawing to a folder with their name	Put drawings into personal envelopes labeled with their name to model “saving” work in organized “folders.”	**Organization**: Understanding that information can be stored and retrieved. **Abstraction**: Using labels and containers to organize data.
	Developing and Using Abstractions		
Collection, Visualization, & Transformation (CVT)	PK.DA.CVT.01 With guidance, sort objects by color and help create a pictograph of the class's favorite color.	Sort colored blocks into baskets, then line baskets up as a living bar graph. Count objects in each basket together.	**Data Collection**: Gathering information systematically. **Pattern Recognition**: Organizing data to see relationships. **Representation**: Showing information in visual ways.
	Creating Computational Artifacts		
Inference & Models (IM)	PK.DA.IM.01 With guidance, talk about which color group has more or less in a pictograph (see standard above).	(See Unplugged Idea above) Compare which basket is taller and discuss “more” vs “less.” Use language like “Most children like blue” or “Only a few chose yellow.”	**Data Analysis**: Making sense of collected information. **Logical Reasoning**: Drawing conclusions from evidence. **Comparison**: Understanding relative quantities.
	Developing and Using Abstractions		

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Algorithms (A)	PK.AP.A.01 With guidance, follow a simple two-step direction sequence (e.g., “clap, then jump”).	Play two-move “Simon Says” to practice sequencing: “Simon says clap your hands, then touch your nose.”	**Sequencing**: Understanding that order matters in instructions. **Algorithm**: Following step-by-step directions to achieve a goal.
	Developing and Using Abstractions		
Variables (V)	PK.AP.V.01 Recognize that computers use symbols (e.g., arrows) to show directions.	Lay arrow cards on the floor; step along the path to reach a toy. Change where arrows point to reach different destinations.	**Symbolic Representation**: Understanding that symbols can represent actions or directions. **Abstraction**: Using simple symbols to convey complex ideas.
	Developing and Using Abstractions		
Control (C)	PK.AP.C.01 With guidance, give a sequence of two to three movements to a classmate acting as a robot.	Child “programmer” gives step-by-step moves to a “robot” friend: “Take two steps forward, turn right, sit down.” Alternatively, because students may not know directions of left and right, label left and right with colors (blue and red) and direct them by saying, “turn right toward red.”	**Programming**: Giving precise instructions to achieve a goal. **Sequencing**: Understanding that multiple steps create complex actions. **Communication**: Using clear, specific language.

	Creating Computational Artifacts		
Modularity (M)	PK.AP.M.01 With guidance, break a classroom clean-up activity into small steps.	List picture cards for cleaning: “pick up blocks,” “put in bin,” “close lid.” Practice doing one step at a time.	**Decomposition**: Breaking big tasks into smaller, manageable parts. **Organization**: Understanding that complex activities have multiple components.
	Recognizing and Defining Computational Problems		
Program Development (PD)	PK.AP.PD.01 With guidance, draw pictures in order (first, next, last) to tell a short story.	Sequence photo cards of daily routine (wake, eat, school) on a storyboard. Talk about what comes “first,” “next,” and “last.”	**Sequencing**: Understanding chronological order. **Storytelling**: Creating step-by-step narratives. **Planning**: Organizing events logically.
	Creating Computational Artifacts		

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Culture (CU)	PK.IC.CU.01 Identify a technology tool used at home or school and say what it helps people do.	Match picture cards of old vs. new tools (letter vs. email, map vs. GPS) and discuss their purposes: “This helps people talk to friends far away.”	**Impact Analysis**: Understanding how tools help people accomplish tasks. **Comparison**: Recognizing how technology changes over time. **Communication**: Explaining technology’s role in daily life.
	Communicating About Computing		
Social Interactions (SI)	PK.IC.SI.01 With guidance, discuss being kind and using gentle hands when sharing devices.	Pass a stuffed toy while saying something kind before handing it on: “You’re a good friend,” “I like playing with you.”	**Social Skills**: Practicing respectful interaction. **Empathy**: Considering others’ feelings when sharing resources. **Collaboration**: Working together positively.
	Collaborating Around Computing		
Internet Safety, Law, & Ethics (SLE)	PK.IC.SLE.01 With guidance, ask permission before taking a friend’s photo with a tablet.	Pretend to take a photo with a cardboard “tablet” only after asking permission: “May I take your picture?” Wait for “yes” before pretending to click.	**Consent**: Understanding the importance of asking permission. **Respect**: Considering others' privacy and feelings. **Ethics**: Learning appropriate behavior with technology.
	Developing a Productive Computing Environment		

Subconcept	Pre-Kindergarten (PK) Objective	Unplugged Idea	Computational Thinking Connection
Emerging Technologies & Artificial Intelligence (ET.AI)	PK.ET.AI.01 With guidance, recognize that some toys or devices can “listen” or “talk back” (e.g., a smart speaker) and describe what they do.	Pass around a toy robot pet; ask how it “hears” commands and how to treat it kindly. Discuss: “What does it do when you talk to it?”	**AI Awareness**: Understanding that some devices can respond to voice or touch. **Input/Output**: Recognizing that devices receive information and give responses. **Interaction**: Learning appropriate ways to communicate with technology.
	Communicating About Computing		

Pre-K Vocabulary Guide

Simple Terms for Complex Concepts:

Technology Words:

- **Algorithm** → “Step-by-step directions” or “Recipe for doing something”
- **Computer** → “Smart machine that helps us” or “Electronic helper”
- **Program** → “Instructions for the computer” or “Computer's to-do list”
- **Bug/Debug** → “Mistake” and “Fix the mistake”
- **Code** → “Computer language” or “Secret instructions”

Action Words:

- **Sequence** → “Put in order” or “What comes next?”
- **Pattern** → “Things that repeat” or “Same thing over and over”
- **Sort** → “Put similar things together” or “Make groups”
- **Data** → “Information” or “What we learned by counting”
- **Save** → “Keep safe” or “Put away to use later”

Problem-Solving Words:

- **Decompose** → “Break into smaller pieces” or “Take apart to understand”
- **Abstract** → “Simple way to show something hard” or “Like a symbol”
- **Model** → “Pretend version” or “Copy that helps us learn”

Social Words:

- **Network** → “Connected together” or “Ways to talk to each other”
- **Privacy** → “Keeping things private” or “Things just for you”
- **Permission** → “Asking first” or “Getting the okay”

Teacher Implementation Tips

Making Computational Thinking Explicit:

1. **Use thinking language**: “We're decomposing this big job into small steps, just like computer scientists do!”
2. **Connect to familiar experiences**: “This is like following a recipe - we need the steps in the right order.”
3. **Celebrate mistakes**: “You found a bug! Good debugging! How can we fix it?”
4. **Make patterns visible**: “Look! We made a pattern with our data - red, blue, red, blue!”
5. **Encourage precise communication**: “Give your robot friend very clear directions so they know exactly what to do.”

Assessment Opportunities:

- Can children follow 2-step sequences independently?
- Do they recognize when something isn’t working as expected?
- Can they sort objects using consistent criteria?
- Do they ask permission before “taking photos” in role-play?
- Can they give clear directions to peers in robot activities?

Extension Activities:

For Advanced Learners:

- Increase sequence length to 3 to 4 steps
- Add conditional language: “If the path is blocked, go around.”
- Create more complex sorting rules: “Sort by size AND color.”

For Emerging Learners:

- Start with 1-step directions.
- Use more physical movement and gestures.
- Provide additional visual supports and peer partners.