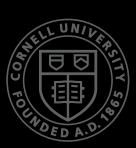


CORNELL
TECH

Computer Science Coaching Toolkit Card Deck

Written by: Meg Ray, Kelly Powers, Joe Melendez and Liz Gallo
Designed by: Niti Parikh

This Computer Science Coaching Toolkit Card Deck was developed out of five years of learning and practice from a CS instructional coaching program at Cornell Tech. Diane Levitt started the Teacher in Residence (TIR) program in 2015 with the purpose to make computer science teachable. We have published multiple research papers about our work in conjunction with our research partners. This card deck represents the expertise of our Teachers in Residence and what we have learned and experienced from practice. These cards also bring together some of the most important ideas about CS education and teacher development from the larger CS education community. You can see the names of these researchers, teachers, and organizations in (italicized parenthesis) on cards related to their work. We highly recommend that you follow their work. This deck is intended for anyone who works with CS teachers in a coaching capacity, whether or not they have the title "coach." It can be used as a tool for the development and growth of coaches and as a planning tool for coaches of all experience levels. Additional resources and full citations for many of the cards are available on the website.

Thank you to all of the teachers, administrators, and students who have participated in the TIR program, to our funders and partners at the Siegel Family Endowment and the Robin Hood Learning + Tech Fund, and to Cornell Tech for its unwavering commitment to K-12 education.

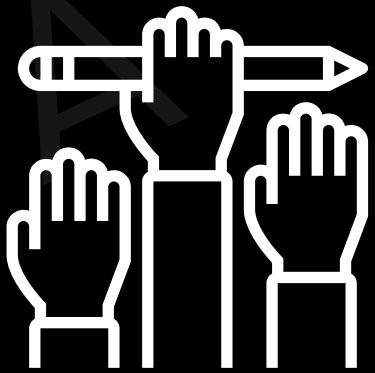
ABBREVIATIONS

CS - Computer Science
CT - Computational Thinking
UDL - Universal Design for Learning
CRE - Culturally Responsive Education
PBL - Project Based Education
TIR - Teacher in Residence
PLC - Professional Learning Community

CONTENTS

	Total Cards
Co-Planning	34
Classroom Implementation	24
Reflection & Feedback	25
Coach's Role	18
Coaching Lenses	17
CS Teacher Moves	16
CS Teaching Strategies	24

Theme:
Access, Equity , and Inclusion



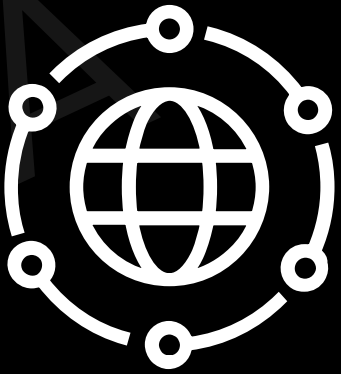
tech.cornell.edu/impact/k-12

Effective CS teachers proactively advocate for equity and inclusion in the CS classroom. They work towards an intentional, equity-focused vision to improve access, engagement, and achievement for all of their students in CS.

Author: CSTA Standards for CS Teachers

These cards include explicit strategies, tools, and/or pedagogies used towards achieving the goal of CS Education for all students in a classroom/school. (CSTA)

Theme:
Community Engagement

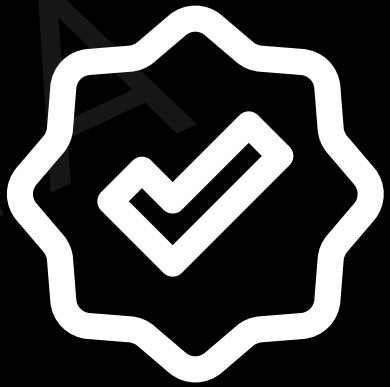


tech.cornell.edu/impact/k-12

CS education is the strongest when presented as interconnected across different stakeholders and aspects in our society. Engaging stakeholders means both building CS community and utilizing community resources. Some stakeholders include: families, classrooms, local communities, business professionals, CS educators, university partners, and others.

These cards have been identified to guide CS educators and coaches to engage with community beyond the classroom. This includes cards related to engaging families, leveraging community resources for richer CS education, and utilizing professional networks.

Theme: **Evidence and Research Based**

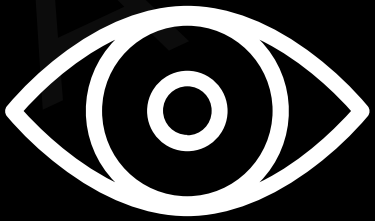


tech.cornell.edu/impact/k-12

Teaching and coaching strategies that demonstrate some evidence of potential for efficacy in one or more research studies are considered evidence-based or research-based.

These cards include coaching or teaching practices that have some form of research backing them up, though it may not yet be comprehensive.

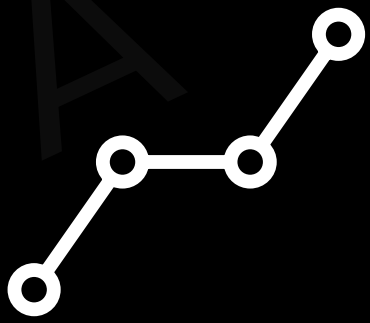
Theme:
Look Fors



"Look Fors" are specific events, words, moments, or actions that a coach can attend to in relation to CS teaching growth and proficiency.

These cards include prompts that can be used to guide the coach's attention during the coaching cycle.

Theme:
Professional Growth



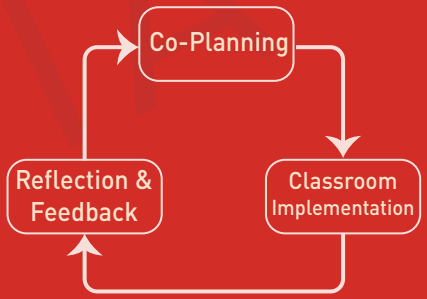
tech.cornell.edu/impact/k-12

Teaching a new discipline and integrating CS into one's domain will require continued professional growth. Exposing the teacher to the broader CS education community and developing a professional development plan with the teacher is essential in building their own understanding of the importance of Computer Science for all.

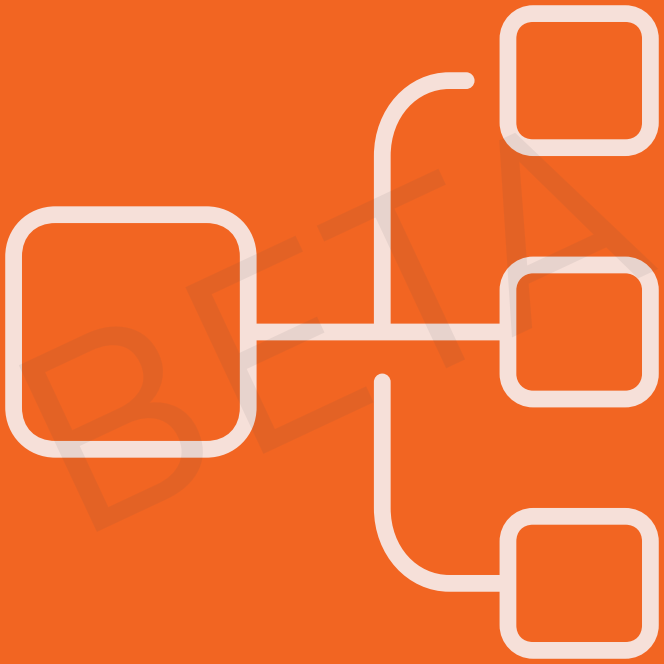
These cards include strategies like communities of practice and references to help a coach guide a teacher to develop a professional development plan.

Co-Planning

Co-planning is a primary way that coaches can build relationships with teachers and help build their content knowledge and pedagogy. It occurs prior to classroom implementation in the coaching cycle as a time to choose curriculum, plan, modify, create curriculum resources, and review how lessons will be enacted. It is also the time to build in actions to improve from prior reflection session.



Coaching Cycle


BETA

A Step Ahead

What: **When co-planning computational thinking or coding activities, offer the teacher one additional step or idea beyond what the students will learn.**

Why: This will improve their confidence of CS knowledge when teaching students.

How: **Continually push teacher's knowledge beyond what they will need to teach their students by practicing additional skills.**

Affective Skills

What: **The affective domain involves our feelings, emotions, and attitudes.**

Why: The affective domain includes the manner in which we deal with things emotionally, such as feelings, values, appreciation, enthusiasms, motivations, and attitudes.

How: **Identify in each lesson the affective objective for students. How do you want to students to feel, what do you want them to value, what attitude do you want them to demonstrate?**

Breaking Barriers

What: **Students' learning barriers must be identified and addressed.**

Why: Identifying barriers that are inhibiting students from learning is the first step in thinking about strategies to remove them.

How: **Decompose the barrier. What are students struggling with and why do they believe students are struggling with that? Then work with teachers to creatively brainstorm solutions to help students overcome those barriers.**

Build it First

What: Use time during a coaching session to practice the CS activity that students are doing.

Why: Since many teachers are new to CS, they may not have experienced the content of the lesson before.

How: Have teachers do the whole learning activity that students would do, noticing their points of frustration or confusion. Use this experience to co-plan the flow of the lesson, clarify instructions, and identify potential misconceptions.

Career Connections

What: **Build teachers' awareness of STEM and CS careers and the intersection of computing across all fields.**

Why: Career pathways in CS and STEM are new and rapidly changing.

How: Talk with teachers about CS careers. Share resources that highlight CS careers and integration across different fields to increase student awareness.

Content Area Connections

What: **Connect CS to other content areas.**

Why: This demonstrates relevance of CS to the students' whole school experience.

How: **Find opportunities to integrate CS content into other content areas. Ideas include connecting CS to grammar and syntax in Language Arts, algebraic/boolean equations in math, experimentation in science, language acquisition in foreign language, storytelling in history.**

Continuous Learning

What: Discuss with teachers how you pursue further learning in CS and teaching.

Why: It is important that teachers see how you as a CS educator continue to learn about CS content and teaching pedagogies.

How: Identify and share key resources, and list the professors, leaders, organizations and research that you follow to inform your practice. Share stories of how these resources have helped you to improve your craft, acquire new lessons, and learn effective CS teaching strategies.

CS Content Standards

What: Connect units, topics, lessons, and activities to relevant CS standards.

Why: Teachers must be familiar with CS standards because they are guideposts for what students should be able to master through various benchmarks in their learning journey.

How: **Constantly review standards with the teacher and reflect on how the lesson content is moving students towards mastery of a standard. Ensure that you look at local, national and course specific standards.**

CS Integration

What: **Design CS integration lessons across disciplines.**

Why: Integrated lessons and projects provide more powerful and deeper learning experiences for our students.

How: Brainstorm ideas with the teacher to plan or integrating or collaborate with other academic disciplines. Bring examples of subject integrated resources. Developing teacher agency and confidence over time will result in teacher driven ideas/lessons for integration naturally.

CS Vocabulary

What: **Develop CS fluency with vocabulary.**

Why: Teachers need to understand and use the CS vocabulary fluently.

How: **Work with teachers to build and reinforce vocabulary around CS. Have them practice, out loud or in writing, how they will use the vocabulary during class.**

Curriculum Mash-up

What: **Combine different CS resources and pieces of curricula to create lessons and units for CS.**

Why: Some CS lessons and resources are stronger for one component and weaker for another. Nothing is perfect.

How: Co-plan a "mash-up" lesson or unit together that combines lessons from different CS resources and/or curricula. Bring in multiple sources and have the teacher talk through their choices for what they want to incorporate, at what point in the unit or lesson, and why.

Curriculum Evaluation

What: **Use critical questions to ask the teacher to evaluate for quality CS curriculum.**

Why: Teachers need to become critical thinkers about the CS curriculum they are presenting to students including how to evaluate curriculum, lessons and projects and determine the appropriateness for CS lessons

How: **Evaluate the CS curriculum thinking specifically about the accessibility, equity and learning concepts and practices. Ask critical questions such as: How would you modify this curriculum for your class? Will this engage all of your students? Does this approach make sense for your students? How does this align to CS standards? Is the pacing right for my students? Is there a level of rigor for my students?**

Decide Ahead

What: **Ask the teacher how comfortable they feel with you working and intervening in their class.**

Why: Teachers have different comfort levels with having other adults with them while they teach. Determining a plan and setting expectations ahead of time helps to ensure a positive coaching relationship.

How: **Prior to observing and implementing, discuss with the teacher their comfort with the coach being present and intervening within the classroom. Reassure teachers that a coach's role is to support their work and not to officially evaluate them.**

Demystifying Computing

What: Increase the teacher's understanding of concepts and vocabulary related to Computing Systems and Networks and the Internet.

Why: Help teachers understanding of how computing systems and networks work will boost their confidence to teach CS instruction.

How: Talk about computing systems and networks as you are using them, using CS vocabulary, then explain what you mean without jargon. Give specific and concrete examples of how computers work. When troubleshooting a network issue, explain to teachers the different parts of a network and how they work together.

Describe Your Role

What: **Clearly identify your role as a CS coach.**

Why: Establishing clear expectations for roles and what will happen during coaching creates a foundation for a good working relationship.

How: As a coach you remain in a non-evaluative position. The goal is to work together to bring CS to their class. Describe the process of the coaching cycle the teacher will experience. Clearly describe your role as a coach to support, guide and help grow with the teacher throughout that cycle.

Embrace Challenges

What: **Things don't always go as planned and making mistakes is natural.**

Why: Coaches use the CT approaches like perseverance and debugging to demystify overcoming challenges as they arise, helping build the teacher's comfort level in doing so.

How: When you come across an error or something goes wrong, don't try to hide it. Normalize errors and model debugging skills and perseverance. Make explicit connections to how the teacher might do this with their class.

Exemplars

What: **Have the teacher create exemplars.**

Why: Creating an exemplar can help teacher skill development and boost their confidence as they model for their students.

How: Ask teachers to create their own best version of the project students will be working on in order to demonstrate how the students' project will look.

Instructional Tools

What: Identify instructional tools to help all students.

Why: There are many different instructional tools that specifically address the diverse needs of learners.

How: **Explore options like digital or paper graphic organizers, vocabulary charts, dictionaries, visual task lists, flow charts, and many more.**

Lesson Assessments

What: **Discuss assessments with teachers at the beginning of the planning.**

Why: Look at the assessments to ensure that the instructional content will get students to where we want them to be.

How: **Check the plan to assess students' learning at the end of the lesson. Determine if it is aligned to the CS objectives and activities.**

Lesson Objectives

What: Objectives identify what the learner should be able to do by the end of the lesson.

Why: Setting clear lesson objectives helps keep the lesson focused and connects the learning experience to the assessment.

How: Use the ABCD model of setting objectives.

Audience - Who are your learners?
Behavior - What will they be doing? What is their task?
Condition - What tools or resources will they use? When should the task be complete?
Degree - To what degree of mastery should the task be completed?

Paint a Picture

What: **Describe what CS education looks like.**

Why: To effectively explain what CS education is, one should have a clear and concise understanding of what it is and most importantly what it looks like.

How: **Paint a picture of what CS education looks like, sounds like, feels like. For example, the teacher could observe the peer coach's classroom or the peer coach can describe what CS instruction looks like.**

Real Life Connections

What: **CS is everywhere in our daily lives.**

Why: Connecting CS to our daily lives brings relevance to the instruction and help students understand how CS impacts everything we do.

How: **Identify ways that students will use CS in their everyday lives, including jobs and current events.**

Student Hat
Teacher Hat

What: During a coaching session, ask the teacher to work through the lesson as if they are a student. Together, look for difficulties that students may experience in this lesson and brainstorm solutions.

Why: Let teachers experience the same struggles and successes as the students in order to understand the student experience.

How: **The teacher acts like the students and works through the lesson. Together you look for difficulties that the real students might experience in this lesson and think of solutions.**

Students' Culture

What: Take into account students' culture when planning CS lessons.

Why: Students become more engaged when they feel they have a connection the content they are learning.

How: **Find opportunities to integrate appropriate cultural and background knowledge into CS instruction.**

Talk Through the Lesson

What: **Talk through the lesson sequence and logistics with the teacher.**

Why: Talking through a lesson will help teachers become fully prepared and helps you identify any gaps where the teacher needs support.

How: **Ask the teacher to walk through the all of the lesson activities and the details of each step.**

Teacher & Coach: Pair Program

What: Use pair programming when developing a student programming activity.

Why: Practicing pair programming will allow the teacher to know how to use this move in the classroom with students and build projects.

How: **Use one device to complete a programming assignment with the teacher. The teacher will be the driver and you will take the role of the navigator. Practice positive communication and perform the roles of driver/navigator. Switch roles every five minutes. Make a plan to use pair programming with students.**

Technology Practice

What: **Encourage the teacher to practice with the technology.**

Why: Teachers often feel most uncomfortable because they do not know how to use the technology.

How: **Sit next to teachers while they use the technology like their students would. Allow time for the teacher to tinker with the technology they will use in their lessons.**

Technology Procedures

What: **Review how the students will get the technology they need for the lesson.**

Why: Reviewing student procedures is often the most challenging task for classes full of students and teachers.

How: **Review, plan, and implement the procedures for how students will access the technology needed for the lesson. Think about how will things stay charged, how will devices be distributed, and what materials students will need.**

The Teacher's "Why"

What: Inform your coaching plan using information learned while discussing the teacher's whys.

Why: It is important to understand a teacher's own motivation for teaching in general and then connect that to their reason for teaching CS.

How: **Meet with the teacher to learn more about them. Develop a set of questions to learn about their journey to the teaching profession. Take notes as you listen. Ask the teacher why they are teaching CS? Ask a follow up question to understand if they feel motivated or pressured to teach CS. Show compassion for the demands placed on teachers and partner with them.**

Troubleshooting Tech

What: **Look for and solve technology issues.**

Why: As a coach, be prepared to resolve and model resolution of tech issues. Tech issues can make teachers nervous and be a barrier to teaching CS.

How: **Help teachers see how you work to solve and debug problems in order to build their capacity to do it themselves. Do small things to get the teacher and class up and running for teaching. Know how to contact an IT person to troubleshoot and debug tech challenges. Try to solve as many technical problems as possible.**

Use, Modify, Create Curriculum

What: **Gradually move teachers from using pre-planned lessons to developing their own.**

Why: Scaffolding ownership of curriculum will help teachers feel confident in teaching lessons to their students.

How: Utilize the CS strategy "Use, Modify, Create" as a way to develop teacher ownership of CS curriculum. When teachers start out, have them use pre-written curriculum. Then have them modify the curriculum slightly. Then, encourage them to create their own projects and lessons. (I. Lee)

What's Working

What: Leverage teachers' existing routines and protocols.

Why: Transferring routines across subject areas helps maintain consistency and builds on the teacher's success in other subject areas. Leverage these already existing routines to help with CS instruction.

How: **Identify teachers' typical routines and protocols. Find ways to incorporate these into CS lessons.**

Who Are We Designing For?

What: **Draw out the teacher's knowledge of individual students to inform lesson design.**

Why: Teachers' knowledge of their students is needed to build an inclusive and positive classroom culture for learning.

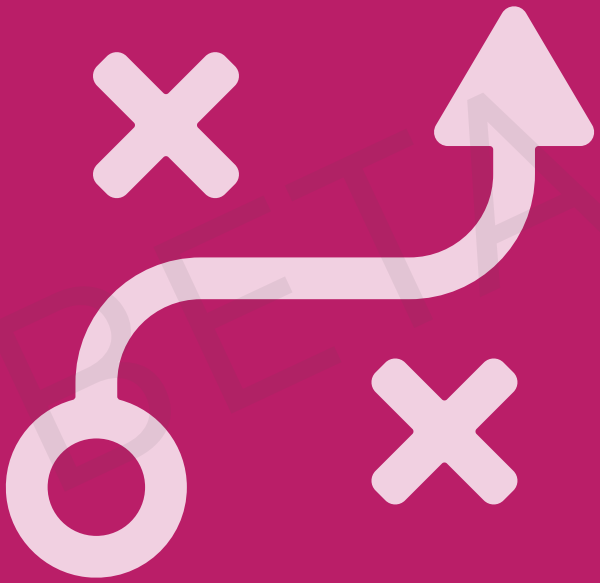
How: Encourage teachers to creatively gather information on student cultures represented, languages spoken in their classroom, strengths and weaknesses, and how students interact with each other. Once collected, the coach and teacher may collaborate on a plan to build an inclusive and positive classroom culture for learning in CS.

CS Teaching Strategies

CS Teaching Strategies are research- and evidence-based best practices for CS instruction. They have been spotlighted by CS Ed communities and researchers as being effective for improving students' learning of CS.

These cards are strategies teachers can adopt to make CS accessible for all students and to ensure all students are learning CS.

tech.cornell.edu/impact/k-12



Collaborative Culture

What: **Continue to design and develop a culture for collaborative discussions.**

Why: Collaborating around computing is a CS practice and a learning objective. Provide supports to help students master this objective.

How: Have a "direct talk" routine for student collaboration (accountable talk is one). Provide students with scaffolded supports to collaborate with each other (for example, try sentence stems).

Commenting

What: **Commenting is writing non-executable notes directly into a program.**

Why: Commenting, a form of collaboration, is a practice used in the field of computing. It helps students organize their program and is a way students can communicate their thinking to their teacher and partners on a team project.

How: **Set expectations for students to document their thinking using commenting within their code. Ensure that you are teaching and modeling this within their instruction.**

Concept Before Code Protocol

What: Teach abstract concepts before students begin to code.

Why: Abstract coding concepts are disconnected and irrelevant to learners when there is no context for the code they are using. Time to "play" with CS concepts collaboratively in a unique, non coding, activity has been shown to improve concept understanding.

How: Teach the concepts of CS before you teach students how to code.

1.) Define the concept

2.) Make an analogy that will relate to students' background knowledge.

3.) Engage students with concepts collaboratively in unplugged or digital activities prior to coding.

4.) Apply the concept to coding.

(S. Grover, et al.) (P. Bagge)

Concrete-Representational-Abstract (CRA)

What: **Guide students through a sequence of connected learning activities:**
Concrete - hands-on materials
Representations - pictorial representation, drawing
Abstract - writing the code
The teacher explicitly bridges connections between each of these.

Why: CRA helps build students' understanding of complex concepts by creating explicit connections from the concrete to the abstract.

How: When teaching computing concepts, begin with concrete representations that use hands-on materials. Then, move to a representational understanding using pictoral and pseudocode. Finally, move to abstract coding experience.

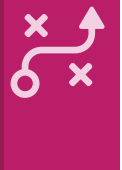
Exploded Code

What: **Provide students with some or all of the code scrambled.**

Why: "Exploded code" allows students to demonstrate their conceptual understanding by creating a program that they might not be ready to create from scratch.

How: Provide students with all or some of the code to complete the task un-assembled. In block-based languages, individual blocks are "exploded" around the code editor. In a text-based language, you might offer completed lines of code in a series of unordered comments. Students use the provided code blocks or snippets to complete a program. (Creative Technology Research Lab, UF)

Flowcharts



What: **Use flowcharts to develop student understanding, thinking, and approach to problem solving.**

Why: Understanding the flow of a program is an important step in learning. Creating flowcharts helps students focus and organize processes using words or images that are easily understood by them and/or their peers.

How: **Using flowcharts, students identify and create a visual representation of a program's process and any conditionals it may have. Students can use specific symbols and conventions to make flowcharts to plan and represent algorithms.**

Kinesthetic Learning Strategies

What: Teach and practice CS concepts using movement and manipulatives.

Why: CS is a very "sit in your seat, look at your computer screen and work" subject. However, there are many ways to get students moving.

How: **Include ways students can engage with CS concepts and activities through movement to express their learning. Unplugged activities, ed tech products that encourage movement and using manipulatives are all great strategies.**

Model Problem Solving

What: **Problem solving is a skill that should be taught and developed.**

Why: Coaches need to expose teachers to explicit problem solving strategies in the context of CS.

How: Model strategies to solve problems with students such as talk through how you are solving a problem. Identify where in the lesson they can use explicit strategies to develop problem solving skills. Use Polya's four step approach to solving problems:

1. Understand the problem
2. Make a plan
3. Carry out the plan
4. Look back, iterate, and go back to step one.

Multiple Entry Points

What: **Lessons designed with multiple entry points provide access and engagement with CS concepts based on each learner's strengths and needs.**

Why: To ensure CS instruction is accessible to all students, teachers must be aware of and use diverse techniques and resources that allow all types of learners an entry point to create and succeed.

How: Offer options for how students can engage with a new concept based on student preferences and learner needs. For example, when starting a new project, students may explore the task by using and modifying a completed program, building a program with disconnected code, debugging a "broken" version of the program, using video aids, using physical cards to scaffold learning, or by creating and adding advanced features to the program. (Creative Technology Research Lab, UF)

tech.cornell.edu/impact/k-12

Parsons Problems

What: **Parsons Problems are lines or blocks of scrambled code that need to be correctly ordered.**

Why: Parsons Problems reduce cognitive load, student frustration over syntax errors, and promote logical thinking.

How: Provide students with blocks or lines of code that are disconnected. Challenge students to unscramble the code to form and test a solution. This strategy may be applied in any programming language.

Peer Feedback

What: **Peer feedback happens when students communicate with one another about their work, orally and/or in writing.**

Why: Feedback gives students an opportunity to practice communicating about computing. Receiving and applying feedback helps students build collegial relationships. Feedback during user testing and iteration are core to CS.

How: **Design a peer feedback protocol to teach and develop students' ability to communicate and constructively evaluate each other's progress on a project. Feedback should be planned to happen during a project so that students can take action on their project to incorporate suggestions. Consider having students use an assignment rubric to review as they assess and give each other feedback.**

PRIMM

What: **PRIMM, (Predict, Run, Investigate, Modify, Make) is a protocol for learning programming concepts.**

Why: The guided process of PRIMM allows students to deepen their learning and interact with completed programs when they are still beginners.

How: Have students read a program, predict what it will do and converse about their predictions. After they run the program, students should discuss the outcome. Students then participate in a guided investigation of the code prior to tackling modification task challenges. Finally, have students create their own program. (S. Sentance)

Pseudocode

What: **Use pseudocode when students are planning and organizing pre-code.**

Why: Pseudocode helps students to formulate and organize their thoughts before starting to code. It removes the barrier of needing to know exact coding syntax.

How: **Have students use a combination of natural language and programming language to sequence out an algorithm when planning a program. Students can express their understanding of a program by writing pseudocode for that program.**

Reading Code

What: When students read code, they create a mental model of what will happen when a program runs.

Why: By reading code, students build both their reading comprehension and CT skills, like pattern recognition. Teaching a strategy to read and investigate code provides needed scaffolding to be able to understand what the code is supposed to look like and do.

How: **Have students practice reading code. Ask students to predict what will happen when the code is run. Have them translate it to pseudocode. Ask students to look for patterns, structure, and syntax. Have them identify the purpose of each line of code.**

Reference Materials

What: **Programming is not about memorizing specifics, it is about being able to solve a problem using a deep understanding of concepts that transfer between languages.**

Why: Providing reference materials and teaching students how to use them helps reduce cognitive load and helps students focus on concepts rather than the details.

How: Programming is not about memorizing specific lines of code. Create or find reference materials appropriate for your students.These could include cheat sheets, documentation of syntax, symbols, code snippets, and built in autofill tools in the code editor. It is more important that students know how to apply, access, and use reference materials for deeper understanding rather than memorizing code.

Scaffolded Constructionism

What: Scaffold constructivist learning approaches.

Why: Based on Papert's research, creating physical and digital products helps students construct their knowledge about CS. The focus on building artifacts in a constructivist learning environment gives us constructionism. We can give all students access to this type of rigorous experience by building scaffolds to independent learning.

How: **Create opportunities for students to construct their own knowledge through open-ended play with physical materials and creating both digital and physical artifacts. Make these experiences accessible to all students by pairing them with explicit instruction, scaffolding exploration, and the artifact development process then by guiding reflection and synthesis. (M. Ray, M. Israel, et al.)**

Subgoal Labeling

What: **Subgoal labeling is a method used to add labels to chunks of code on worked examples that describe it's purpose or goal.**

Why: Subgoal labels on worked examples have been used to reduce the learner's cognitive load and aid in the learning of how problems are both constructed and solved.

How: Decompose worked examples into code chunks and add subgoal labels. Use code comments or visual text aids to define the purpose of each segment. After some practice deconstructing code and creating code in novel problems, give students a worked example and have them create subgoal labels.

Teaching CS Vocabulary

What: Teach and reinforce the use of CS vocabulary.

Why: Setting expectations for using CS academic vocabulary builds and develops content knowledge in CS.

How: Teach CS vocabulary using explicit instruction strategies. Scaffold and reinforce students' use of academic language during CS lessons. While engaging in CT/CS activities, remind students to put their CT/CS "hats on". In this role, students are expected to use CS academic vocabulary words.

Tech Helpers

What: **Define and share student tech helper roles.**

Why: Having student tech helpers builds a collaborative classroon and a sense of community while reducing teacher overload.

How: Create a student team to assist with devices, materials and helping other students. Provide guidelines and scaffolding to student helpers to prevent them from doing the work for others. Rotate students to allow all students the opportunity to take on a tech helper role. Post the current team of assistants visibly on the board or in a handout.

That Could Be Me

What: Students need to see themselves as potential innovators, collaborators, and contributors in the field of CS.

Why: Stereotypes, attitudes, and access influence the lack of diversity in CS fields and a student's interest in pursuing advanced education in Computer Science. Students need to see themselves as part of a CS community and have opportunities to engage with exciting CS projects.

How: **Dedicate class time to show current news of diverse professionals and students of all ages involved with the application of CS to solve a problem in society or the workplace. Highlight technological advances in fields that show diverse teams collaborating on difficult problems such as disease detection, firefighting, artistic productions, farming, disasters, environment, etc.**

TIPP & SEE

What: TIPP & SEE is a metacognitive strategy used in conjunction with Use, Modify, Create while learning to program.

Why: Students using the TIPP & SEE Model have been shown to use more programming elements and write longer programs.

How: Students preview a project, using 'TIPP': examine the title and wonder what the title implies, read the author's instructions and purpose and then play the project. Next, students apply the 'SEE' strategy:
See the project code, examine events and then explore the project by making modification. TIPP & SEE can be used with Scratch and can be adopted to be used with other programming platforms. (J. Salac, et al.)

Tracing Code

What: **Tracing code is a strategy used for exploring states of a program while executing.**

Why: By tracing code, students will learn to comprehend changes in program variables over time and build their debugging skills when issues arise.

How: Make sure that students first understand the problem they are solving by reading the problem. Next, ask students to find where the program begins. Start tracing each line, conceptualizing what is being run one line at a time. Have students use a pen and paper to record values of code variables as code executes. Creating a table to trace the value changing as the code runs is a strategy used for students to conceptualize how code is running.

Use - Modify - Create

What: **Use - Modify - Create is a three stage progression used to engage students in CS learning.**

Why: When students review someone else's finished work or modify existing work, it has been demonstrated to reduce anxiety and increase confidence.

How: First have students interact with a finished program and probe them with inquisitive questions. Then challenge students to make modifications to the provided program and make it their own (remixing). Finally, students create their own project that mimics the original program. (I. Lee, et. al.)

Worked Examples

What: Introduce new CS concepts using worked examples.

Why: Worked examples that are scaffolded reduces the cognitive load on learners when learning new concepts. It shows students why we use specific code and how it works within the context of a completed program.

How: **Show a completed program with detailed comments. Examine the completed program with the new concept in pairs or as a class and discuss it. Students may then reference the worked example while using a new CS concept.**

Zone of Proximal Development (ZPD)

What: **The ZPD is a range of skills that students can perform with the guidance of a peer and/or scaffolds provided by a teacher.**

Why: To maximize student engagement, students must be able to apply content learned. Problems should not be overly complex, causing some individual students to become overwhelmed or too simple, causing boredom. Teachers need to understand what their students can do independently so they can help students grow to their optimal learning potential.

How: **Show students a completed project. Have students work within their level of skill, applying concepts learned by making modifications to the project. To reach their highest potential for learning, scaffold additional activities and structure peer collaboration so that students can build their own version of the project.**

Coaching Lenses

These are themes that have been pulled from across different coaching strategies. They are core coaching values, mindsets and concepts, pedagogical frameworks and approaches, and strategies that are found across different coaching models.

These cards offer a guided approach for navigating through the card deck. They have been designed to focus attention on specific lenses and meant to help decompose and inform the coach's work. They may be used throughout the different stages of the coaching cycle and may guide all of the work in a specific direction OR to help meet the goals of the school/teacher the coach is working with.

tech.cornell.edu/impact/k-12

Coaching Questions

What: Often the coach guides communication with a set of carefully crafted questions.

Why: Conversations between teacher and coach persist throughout the coaching cycle. To sustain dialogue with the teacher, a coach will need to both craft and anticipate questions to ask the teacher to guide constructive conversations and advance teacher learning.

How: **Focus on engaging teachers in dialogue to provoke deep thought and reflection. Encourage teacher-led dialogue and lean on their expertise to respond to questions. Avoid closed questions and keep creativity and positivity high by listening to the teacher and using language that invites dialogue.**

How might I construct questions to prompt critical thinking and open communication about teaching CS?

Culturally Responsive Education

What: Culturally Responsive Education (CRE) is a pedagogy in which one recognizes, responds to, and celebrates how culture plays a role in shaping the thinking process of groups and individuals.

Why: Acknowledging and responding to fundamental cultures allows education to be equitable and accessible to all students. Students become active participants in their learning when they are allowed to share viewpoints and perspectives based on their own cultural and social experiences.

How: **Offer additional resources and training in CRE to help teachers identify their own cultural context and how it informs their teaching. Then prompt teachers to allow students to express their personal, family and cultural experiences and then ask how teachers might connect their CS lessons to students' community and prior knowledge. (Brown University)**

How might I lead teachers to professional development that will help them learn about culturally responsive education? How might I help teachers go deeper than the surface level to learn more about their students and their cultures?

Curiosity

What: Be open to learning new things from the stakeholders with whom you work.

Why: Every moment is a new learning opportunity for you as a coach. The stakeholders with whom you work may have new insight or a different experience that may help you further improve upon your own practice.

How: Set your intention to approach a coaching session with curiosity. Try to notice when you might be assuming intentions behind actions or assuming the outcome of a lesson. Step back and remind yourself that you do not know the answer. Ask a question or make an "I wonder" statement about what is happening.

I wonder what factors contributed to..... ?
I wonder how students might respond to.... ?
I wonder how this team will address this challenge... ?

Empathy

What: Coaches understand and acknowledge the complexities of the teacher's role.

Why: Being open-minded and empathetic towards the teacher as an equal colleague and human will result in a more effective learning and coaching experience.

How: **Holding teachers in unconditional positive regard requires coaches to widen the aperture for possible explanations of challenging behaviors. Coaches consider not just internal motivations, but external circumstances and the impact of the system on teachers. Coaches practice empathy by accepting the teacher's feelings and reactions as valid and responding with compassion and respect.**

How might I consider outside factors may be affecting the teacher today? How might I consider how the system the teacher is operating in creates obstacles for the teacher? How might I take the teacher's feelings about this into account during our conversation?

Flexible Coaching Cycle

What: The coaching cycle is meant to be a flexible structure that allows you to respond to the needs of a teacher or a school.

Why: Coaching only works if it can work within the constraints of a school and if it is centered around relationships, not a rigid structure.

How: A typical coaching cycle is iterative, and looks like this: Co-planning -> Classroom Implementation -> Reflection & Feedback -> Repeat. However, a coach must be flexible and responsive to teachers' and schools' needs. All of these steps may look different through time, teachers and schools.

How might I develop a coaching cycle that is responsive to the constraints of the school and impacts its teaching of CS?

Gender, Equity & Inclusion

What: Help teachers create equitable and inclusive learning opportunities for students of all genders in CS by attending to language, representation, and teaching behaviors in the classroom.

Why: Gender inclusive teaching practices can build the confidence of all students. Research shows that girls' participation in CS drops dramatically by middle school and that girls are underrepresented in high school CS classes. Transgender and non-binary students remain invisible in CS education because they have not yet been counted.

How: Gender equity and inclusion can be promoted in many ways: through visual representation, curriculum content, gender inclusive language, and critique of gender stereotyping. During classroom implementation, teachers can make sure to call on all genders equally and assign challenging assignments equally. Coaches should look for these types of practices in order to provide feedback. This can help teachers reflect on how their own attitudes about gender contribute to the teachers' expectations of students and instructional behaviors in the classroom. Gender inclusive coaches reflect on their own biases and attend to how gender identity influences their coaching.

How might I attend to gender inclusivity throughout the coaching cycle? How might I reflect with teachers on gender inclusivity in the CS classroom?

Honor Expertise

What: Leverage the expertise that the teacher has already developed.

Why: The coach/teacher partnership implies a hierarchy of expert/novice which may impact the coaching relationship. How the coach honors the expertise and strengths the teacher brings to the partnership will influence the effectiveness of the coaching experience.

How: **Coaches must always value the strength and expertise that teachers bring to the partnership. Continuously be on the lookout for these things in teachers and build off of those areas to improve CS instruction.**

How might I acknowledge and leverage teacher expertise?

Individual & Systems Change

What: Coaches work with individuals and administration to create meaningful change within the education system.

Why: In order to make room for CS, individual schools and larger school systems must shift. Systemic change is difficult and requires community buy-in, perseverance, and time. Coaches can play a key role in facilitating this shift.

How: **Coaches are in a unique position to work across a system. Coaches can spread best practices and consistency, remove barriers to CS instruction and teacher development, mediate between and connect stakeholders, and influence school culture.**

How are my actions/interventions creating and influencing positive change at a school level?

Inquiry-Based Learning

What: Inquiry-based learning is an approach that encourages students to construct their own knowledge through teacher-guided exploration and questioning.

Why: Since CS is new to most teachers, they often want to just provide students with facts so everyone "gets it right". An inquiry-based learning approach is key to developing computational thinking skills and leads to higher order thinking skills and deeper levels of learning.

How: **Coaches can use and model inquiry-based strategies as they support teachers learning through the coaching cycle. Coaches can work with the teacher to implement inquiry-based learning strategies in CS instruction.**

How might I allow teachers to construct their own knowledge about CS teaching? How might I support teachers to have effective use of inquiry-based instruction in their classes?

Project-Based Learning

What: Project-based learning (PBL) is a student-centered approach to teaching that allows students to learn problem-solving techniques and subject content through the exploration of real-world challenges.

Why: Through PBL students solve big, authentic problems by collaborating with peers and using resources and knowledge to develop a solution. This also ties to one of the main practices of CS education which is creating computational artifacts that are meaningful to the student.

How: **Coaches support teachers throughout the entire PBL process from planning through implementation and assessment. Coaches work with teachers to ensure that students are experiencing solving a real world and meaningful driving questions that includes computer science.**

How might I scaffold planning and implementation of PBL in CS experiences? How might I help teachers ensure that their PBL experiences are tied to real world problems?

Racial Literacy

What: Racial literacy refers to educators having the knowledge, skills, awareness and dispositions to talk about race and racism effectively.

Why: Racial literacy allows coaches to competently address structural racism in education and in the field of CS. It makes coaching relationships more trusting and effective when coaches are able to examine their own implicit bias and openly address differences in racial identity with courage and compassion.

How: All educators must assess their comfort and facility in addressing issues of race in the classroom, and this is especially true for coaches. Racially literate coaches surface differences in racial identity in the coaching relationship and invite teachers to do so as well when those differences play out in our practice or conversations. Coaches should actively seek training or support from experts in diversity and inclusion or anti-racist education, rather than relying on colleagues. (E. Aguilar)

How might I discuss and respond to topics of race in a mindful manner with both teachers and administrators? How might I remain aware of how my racial identity influences my coaching? How might I coach across racial differences with humility and grace?

tech.cornell.edu/impact/k-12

Self-awareness

What: It is vital for coaches to reflect and iterate on their own coaching, be aware of their own implicit biases and counteract them, and practice stress reduction and emotion regulation during coaching sessions.

Why: Coaching requires a high amount of emotional intelligence in order to keep the focus on teacher learning and growth.

How: Coaching should be both reflective, iterative, and intentional. Coaches should make time to reflect after each coaching cycle and make adjustments based on direct and indirect feedback gathered throughout the cycle. Coaches can use these reflections to examine their own implicit bias, becoming more aware of their own triggers, and practice mindful stress reduction and emotion regulation during coaching sessions. This type of internal work, allows coaches to focus on teacher growth.

How do I feel right now? Why am I having this reaction? What situations really upset me? Am I making any assumptions? Did I communicate what I meant?

Storytelling

What: Leverage storytelling to build rapport and promote learning, increase confidence and critical thinking.

Why: Storytelling is a primary way that people learn, communicate, and make meaning. Storytelling can be used to help you better understand a teacher, to communicate ideas and concepts, to process experiences, and to internalize learning.

How: **Storytelling is especially useful to highlight coaches' own CS teaching experience, and how to managing difficult situations and model resilience and persistence. Coaches also invite teachers to tell stories about their experiences teaching CS, what happened, what was taught, what was the outcome.**

How might I create opportunities to share stories and invite storytelling during the coaching cycle?

Student Impact

What: While coaches may not work directly with students, their ultimate goal is to positively impact student learning and well-being through improved teacher practice.

Why: The purpose of education is student learning and growth. The outcome of any support and development work must have student's best interests at the center.

How: **Pay attention to student impact and incorporate student learning data into your coaching practice. Collect anecdotes of individual student impact and survey impact across groups.**

How is the coaching work that I am doing evident in student work, student behaviors and affecting student attitudes toward computing?

Teacher Voice & Choice

What: When the coach acts as a guide and a teacher as an expert.

Why: The teacher being coached will ultimately make their own decisions about planning and implementing CS in their classroom.

How: The teacher is the expert on their students, the curriculum, and class culture. The coach has experience and resources, but must work to combine that with the teacher's expertise and autonomy. Rather than imposing the coach's instructional decisions, and acting as the knower of all, the coach helps the teacher make their own instructional decisions.

How might I act as a guide to support autonomous teacher growth? How might I respect and honor the teacher's decision making for their own class?

Translanguaging

What: Translanguaging is a pedagogical approach that encourages students to use all of their language and communication strategies to make meaning, learn, and express themselves. In CS, code is a form of linguistic communication.

Why: Use of translanguaging in the classroom recognizes the value of linguistic diversity and fosters deeper understanding and more meaningful expression of learning from bi/multilingual students. It also supports students in acquiring and comprehending English while maintaining their native language.

How: Coaches work with teachers to find opportunities to encourage students to work and represent in the language with which they are most comfortable. Coaches walk through this process with teachers and support them where they need help. Offer additional resources and training in translanguaging to the teacher. (O. Garcia & L. Wei)

How might I support teachers in using translanguaging with students even if the teacher does not speak their language?

Trusting Relationship

What: Coaching centers on cultivating a trusting, collegial relationship between the coach and teacher.

Why: Teaching practices are deeply tied to the individual teacher's own values and identity. Teachers can only find space for growth and change when they feel safe and courageous enough to do so through a trusting relationship with their coach.

How: **Coaches build a supportive partnership with teachers through conversations that reveal the thoughts, experiences, and feelings behind their teaching practice. Coaches should be clear about the information that gets shared with others while assuring that other information is kept confidential. Coaches address difficult classroom issues and critical feedback with honesty and care.**

How does this action/strategy/intervention build or maintain our trusting relationship?"

Universal Design for Learning

What: Universal Design for Learning (UDL) is a framework to improve and optimize teaching and learning for all people based on scientific insights into how humans learn.

Why: UDL increases access and inclusion for all students, including students with a range of learning needs.

How: Coaches can integrate UDL processes and approaches into support materials and coaching conversations, for example highlighting ideas that UDL focuses on like identifying barriers to learning within the curriculum and classroom. Coaches can also focus on supporting the teacher in integrating UDL principles into CS lessons. (CAST)

How might I use strategies from the UDL framework to shape the coaching cycle? How might I lead teachers to use UDL in their classroom?

Coach's Role

A coach's role extends beyond the typical coaching cycle to expand CS education's value within the system and ensures sustainable, ongoing success of CS education. A coach must identify their role within the system and seek opportunities to create systemic change, this including professional learning opportunities for themselves.

These cards assist coaches in enacting the many layers of their role outside of the coaching cycle in relation to school and broader education systems. Not all roles in this big idea will apply to all types of coaches.


BETA

Administrator Connections

What: Create cross-school administrator relationships.

Why: Seeing what other local school administrators are doing with CS programming and teaching offers administrators a concrete model of CS in action and allows school leaders to share ideas and best practices.

How: Prepare a data sheet and if possible contact sheet of schools committed to and implementing CS in their school day for all students. Consider scheduling a meeting between administrators from these schools. Have school administrators visit another school and observe CS education in practice.

Administrator's Capacity

What: **Build administrators' capacity for supporting CS.**

Why: Familiarizing administrators with CS content and teaching practices gives them ownership. It allows them to better communicate with the school community and better support teachers.

How: **Invite administrators to see CS teaching in action in their schools. Use the time to point out examples of strong CS teaching and CS specific strategies. Familiarize administrators with CS standards and learning goals.**

Building Mentors

What: **Developing mentor teachers is a desirable goal.**

Why: Developing teacher mentors will work toward building a sustainable CS ecosystem without you present and give other teachers an accessible colleague to go to about CS instruction.

How: **Develop mentor teachers that are eager to take a leadership role around CS education for the school. With administrators' support and approval, develop mentor teachers to co-lead PLC's, and PD's. This person can become a point person to disseminate information and learning opportunities about CS to their colleagues.**

Class Culture

What: **Do an initial general observation of the teacher and the class to understand how the culture works in the class.**

Why: By observing the teacher you can learn what works within their class culture and be responsive to that when integrating CS practices.

How: **Observe the teacher and class to understand the class culture. Look for general areas, such as classroom tone, routines, management, strengths and growth opportunities in their teaching practice. This is to inform your coaching plan, not to provide feedback to the teacher.**

Coaches' Learning

What: **Be a lifelong learner.**

Why: CS coaches must be open to new ideas, concepts, and approaches to CS instruction.

How: **Attend to your professional growth at different levels including: CS content knowledge, CS pedagogy, inclusive frameworks, and coaching practice. Attend conferences, read journals and build your professional learning network. When possible, contribute to the field by presenting at these conferences or speaking to others in the field.**

CS for All Staff

What: **Advocate for CS for all staff members in the school building.**

Why: All staff members, paras, specialists, English language learner teachers, special educators, play a big role in the classroom and student learning. Offering CS learning opportunities to them increases access and availability of CS education to all students.

How: **Advocate for all teachers/staff that interact with students to support CS learning, including special educators, teacher assistants, specialists, ENL specialists, and para professionals have access to CS learning opportunities. These may include PD, joint coaching sessions, professional learning communities, planning time and offering bite-sized coaching during classroom implementation.**

CS Implementation Planning

What: **Co-design a CS implementation plan with clear expectations.**

Why: Involving school administrators with high level decision making and keeping them informed helps them to create a school culture that involves CS and sustain the program.

How: **Co-design an implementation plan with administrators and school appointed leaders. Include regular updates on CS implementation and progress towards any school-wide goals. Share successes with the school community.**

CS is Different!

What: **Acknowledge that teaching CS is different from other subject areas.**

Why: Teaching CS, using CS pedagogy, and utilizing inclusive frameworks is new to a lot of teachers. Teachers without a CS or education background are attempting something that can be challenging and intimidating.

How: **Give administrators context by communicating the unique challenges of learning to teach CS and highlight teacher successes. Advocate for the resources and environment that teachers need to develop new skill sets.**

Use the coaching cycle to build teachers' confidence over time by going slowly with teaching new content knowledge, pedagogies and frameworks. Validate their struggles, find connections to the familiar, and acknowledge accomplishments throughout.

CS Vision

What: **Work with stakeholders to build a vision for CS.**

Why: Having a CS vision for the school will guide your CS work with teachers and throughout the school's system. It will help you determine if you are building teachers' specific capacity or working toward a system change.

How: Visualize with stakeholders what CS could look in this school. Ask them what their ideal state and their minimum state of success looks like for CS education. Do they envision a school where CS is a standalone course, or do they envision CS integrated into every discipline? There are many planning tools available to develop a CS/CT vision for the school.

Family Engagement and Inclusion

What: **Consider families as advocates for the development and success of school's CS programs.**

Why: By exposing parents to CS, they will have a better understanding on why CS education is crucial for their children.

How: Encourage schools to host family nights and invite families into the classroom during CS instruction. Provide resources for schools to plan family events that introduce families to computational thinking games and activities, and engage parents in supporting CS instruction and exposing them to career pathways with CS.

Learning Journey

What: **Share your CS learning journey with teachers you coach.**

Why: Understanding and preparing to teach CS is a huge undertaking for teachers. Acknowledging and sharing your stories of failures and growth shows compassion and helps develop the coach/teacher relationship.

How: Start your conversation with some background or a story. Highlight CT approaches you developed to persevere, and resources you learned about during your journey. Emphasise that learning is process and it will take time to develop skills as a CS teacher.

Positive Culture of CS

What: **Arrange to work with groups of teachers during team planning meetings.**

Why: By attending meetings the coach has the opportunity to build a relationships with the team, showcase their availability, and highlight CS.

How: **At meetings, highlight teachers successes and present opportunities to include CS. When possible provide CS professional development for all the teachers.**

Professional Learning Communities

What: Look for opportunities within structure of school outside of coaching cycle to keep CS learning visible.

Why: Coaching cycles are limited in time and may be focused on specific teachers. To build momentum for CS at a school, time outside of the coaching cycle will contribute to building a CS sustainable program.

How: **Ask school leaders to form a Professional Learning Community for CS that meets frequently throughout the year to discuss CS in classes, in events and the CS culture of the school. The PLC time can also be used to increase CS knowledge of content or pedagogy to be shared with all members of the staff.**

Professional Learning Networks

What: **Connect teachers with CS Professional Learning Networks (PLNs).**

Why: There are so many great resources created by teachers and researchers. Helping teachers recognize leaders and build a robust professional network that they can learn from is important. CS is a new content area that is always changing so learning from others in the field is helpful.

How: **Talk to teachers about experts that you follow, that you trust, share their social media accounts, websites and other information about them. Encourage them to join groups and communities to discuss CS.**

Program Needs Assessment

What: **Assessing the school's program and identifying needs and gaps related to CS education.**

Why: Support the school in providing all students a consistent CS education throughout their school experience.

How: Examine the whole picture of CS implementation, including any plan and its implementation. Identify strengths and gaps in CS content or pedagogy across grades, and work with the school to make a plan to build on strengths and build on gaps.

Promote CS Careers

What: **Highlighting CS across careers and real-world applications.**

Why: Teaching about CS for problem solving skills is important. But it is also important to help students see how CS and CT play an essential part in all careers.

How: **Share resources about careers in computing with teachers and help teachers highlight CS careers for students. Also help teachers make real-world connection across the CS they are teaching.**

Purpose Statement

What: **Create your own purpose statement as a CS coach that you can refer to during coaching conversations.**

Why: Creating and describing your purpose and your motivation will provide teachers a context for your role in the school.

How: **Write a brief statement that captures your core values and beliefs about CS education. Include what drives your passion and motivates you to inspire and develop teachers' content and pedagogy. In your coaching practice, when you make decisions or face changes, return to this statement for guidance. Share this statement with teachers and schools when you begin your work together.**

School Priorities

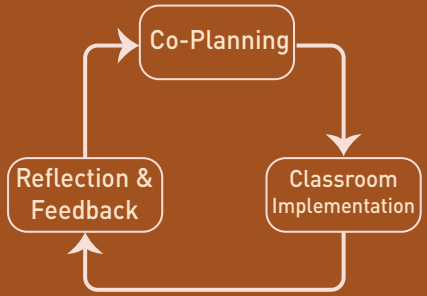
What: Aligning CS education with school initiatives and agendas will help schools prioritize CS.

Why: Get to know the school's goals, initiatives and students' needs. Determine how initiatives may compete with time for CS instruction. Think about how the CS program could support those goals and needs.

How: **Annually and sometimes more often, schools have initiatives implemented for ELA/math, social emotional learning, parent engagement, etc. Determine how these initiatives/opportunities will "compete/cooperate" with time for CS instruction. At a later date look for opportunities to integrate CS and align CS with the goals of competing initiatives.**

Classroom Implementation

Classroom implementation is a time when teachers teach a CS lesson with the coach present. A coach, while ensuring a joyous and positive collaborative time with the teacher, focuses on building teacher's CS practice and pedagogy. A coach actively engages during lesson implementation through modeling, co-teaching, or observing to later provide feedback.



Coaching Cycle

Bug-in-the-ear

What: **Prompt the teacher discreetly during instruction.**

Why: When teachers are prompted with practical strategies, they are more likely to implement them in the moment and in the future.

How: **Prompt the teacher discreetly during the class period with a CS Teacher Move that they can enact in the moment.**

Co-Teaching

What: **Co-teaching is when you and the teacher share the responsibility for teaching the class.**

Why: Co-teaching allows for modeling and scaffolding for the teacher while encouraging them to own the instruction. Establishing a partnership in the classroom maintains the teacher's position even when a content expert is present.

How: **Co-teach the lesson with the teacher based on the co-teaching roles you established in the co-planning session. The co-teaching models that lends themself to coaching include leading and assisting or co-presenting equally.**

I Do, We Do, You Do

What: Introduce the gradual release model. "I do, we do, you do," for new teaching concepts.

Why: The gradual release model builds teacher confidence and identity as a new CS teacher.

How: First, model teaching a new concept (I do). Then the teacher and you co-teach the new concept (We do). Next, the teacher teaches the new concept independently and receives feedback from you (You do).

Intervene

What: **Interventions are sometimes necessary to clarify CS concepts.**

Why: Building deep CS understanding takes time. Intervening when a teacher is explaining a concept incorrectly is important to ensure concepts are learned correctly.

How: **Politely intervene in a way that you and the teacher agreed upon during co-planning. Use light sentence openers, such as "Ms/Mr teacher, I've seen that done..." Address the students and teacher by decomposing the concept in a manner that clarifies it well for both the students and teacher.**

Look For: Classroom Culture

What: **Look for evidence of a classroom culture that supports CS attitudes and dispositions.**

Why: A CS classroom culture embraces mistakes and develops a positive problem solving mindset.

How: How does a teacher foster CS attitudes and dispositions? Is there room for collaborations, mistakes, perseverance, joy in problem solving? Look for procedures rules and norms and how they affect learning.

Look For: Collaborative Learning

What: **Look for collaborative learning opportunities.**

Why: Collaborating is a core practice of CS, ensuring different perspectives and inclusiveness. Teachers may be new to or need support to implement collaborative learning structures.

How: Look for the teacher modeling and developing collaborative learning norms. Are students taking turns talking about CS content? Are all students having an input on the CS project they are working on? Does the teacher provide a clear structure for collaborative work? Is there fair grouping? Are there roles? Does the teacher model accountable talk? Does the teacher provide access to collaborative groups for emerging bilinguals and students with disabilities?

Look For:
Content Accuracy

What: **Look for evidence of accurate and appropriate CS content.**

Why: Quality CS education happens when teachers combine accurate content and appropriate pedagogy.

How: Listen for the CS content teachers give to students. Is the content presented in a way that is age-appropriate for students? Are they using CS vocabulary correctly and incorporating it into their instruction? Are concepts misleading in ways that students will need to unlearn? Is it being taught in the correct context? Is there important information missing? Are teachers addressing common misconceptions? Does it include too much technical detail that detracts from the learning objectives?

Look For: Extended Learning

What: Notice what students do when they are done with the learning activity.

Why: Once students complete the requirements of an activity they may need to be encouraged to achieve beyond the lesson requirements using extensions.

How: **Look for protocols that offer options for once an activity is completed. These may include continuing practice, or deeper dives into content, or creativity and personalization.**

Look For: Instructions

What: **Clear instructions set the stage for success in CS learning activities.**

Why: How instructions are communicated to students can support or hinder CS learning.

How: Are instructions comprehensible to the students in the class (language, images, format, multiple modalities)? Are instructions broken down into steps (appropriate to the grade level)? Are instructions available to students to refer back to throughout the activity? Do students understand what is expected of them? Are students able to follow the instructions and complete the steps?

Look For: Productive Struggle

What: **Productive struggle gives students space to grapple with problems and make mistakes in a safe setting.**

Why: Struggle is productive when it is intentionally scaffolded and does not push students into frustration and stress levels that interfere with learning. It supports self-efficacy and reinforces problem solving skills. It breaks down learned helplessness behaviors and facilitates deeper learning.

How: **Does the teacher tend to solve students' problems? Are students given time and resources to problem solve before receiving help? Is the teacher supporting students by prompting them to describe their thinking, try out different solutions, and tolerate ambiguity? Are students unengaged due to boredom or frustration?**

Look For:
Student Engagement

What: **Student engagement is the degree to which all students are actively participating in their own learning process.**

Why: Students learn more when they are actively engaged in learning. High active engagement builds all students' interest and motivates them to advance their learning.

How: How does a teacher engage every student in learning a new CS concept? Does the teacher provide "multiple entry points?" Is the lesson teacher focused or student focused? How does the teacher solicit student participation? Do students share their thinking, ask questions, work collaboratively? Do students share out, write in journals, complete exit slips to self-report their learning and level of engagement?

Look For: Student Talk

What: **Student talk is the way students are talking about the CS content.**

Why: Students learn when they are engaged in collaborative, social learning experiences.

How: Are students given the opportunity to interact with each other during the lesson? Are they using appropriate CS vocabulary, sharing their thinking about the work, giving helpful feedback to each other, collaborating towards a common goal?

Look For: Student Thinking

What: **Making student thinking visible informs teaching practice.**

Why: Students' understanding of concepts becomes clearer when given the opportunity to "show" their thinking in different ways.

How: **Do students have opportunities to explain their thinking verbally, in writing, or through drawings? Is the teacher prompting students to identify the CS concept they are using and how? Is there time for students to reflect at the end of the lesson?**

Look For: Student Voice and Choice

What: Throughout the lesson, there should be opportunities for student voice, meaningful choices, and personal expression.

Why: Ensuring that some student choice and voice is built into a lesson to increase student engagement and agency toward CS.

How: **Do students have the opportunity to provide feedback and input and ask questions? Do students have the opportunity to make important and meaningful choices? Are there opportunities for personalization?**

Look For:
The Third CS Teacher

What: The "Third Teacher" is the physical or digital space in which the students learn.

Why: The walls of a classroom impact student relatedness/connectedness to the field of computing and as a resource to support CS learning.

How: **Look for learning supports about CS around the space in word walls, anchor charts, visual cues, posters, books. Look for how the physical space is set up to leverage CS experiences for the students. Look for equitable representation visible in the room.**

Model Teaching

What: When modeling CS select a specific piece of the lesson to teach.

Why: Modeling teaching of CS content gives teachers the experiences of observing best practices.

How: Model a specific teaching practice or skill by purposefully teaching a targeted portion of the lesson. The teacher observes you intently while you teach the targeted CS Moves or CS Strategy.

Observing

What: **Observe the teacher and students during a CS lesson.**

Why: Observations paired with meaningful feedback are used to help teachers grow in their practice and provide the coach with insight into their teaching.

How: **Observe a class session and take notes. Observe both the teacher and the students. Have "look fors" in mind that are aligned with the teacher's goals and stay open to what is happening in the room.**

One At a Time

What: Introduce discrete CS Teacher Moves one at a time.

Why: Exploring too many CS Teacher moves in a lesson may be overwhelming and impact teacher success.

How: **Explore CS teacher moves aligned to the student lesson goals one at a time.**

Positive Problem Solving

What: **Seek opportunities to model positive problem solving approaches.**

Why: Modeling positive approaches in front of students and with students, when addressing a problem, impacts the student culture of collaborative problem solving in the classroom. It is okay to make mistakes in front of students.

How: **When an error occurs with tech or in a program, interject to share your own joy in debugging and persevering through a tough problem. Express to the teacher and students how much you love to solve problems. Students' approaches and mindsets begin to change when the teacher models this behavior regularly and encourages students to joyfully debug using perseverance and peer collaboration.**

Stick to Your Role

What: Your role is to be a guide and actively help the teacher grow.

Why: Being a guide will build the teacher's own competence, confidence, agency and ownership.

How: Do not feel that you have to teach the lesson even if it is not going well. Provide the teacher with CS Moves and guide the teacher with advice and suggestions. Look at the teaching as an opportunity for growth and development.

Student Input

What: **Interact with students during CS lesson implementation to solicit their input.**

Why: When we have the students' perspective, we can see if it matches with the teachers expectations to see if the CS lesson is effective.

How: Let the teacher know that during classroom implementation, you intend to interact with students to gather information to discuss during reflection. Choose students randomly. Ask students what they are doing, to share their thinking with you and describe their artifact. Write down these notes to share with the teacher to discuss and give feedback.

Student Work Time

What: Identify strategic tasks to perform during independent student work time.

Why: Student work time is another opportunity for you and teachers to continue building best practices.

How: Coaches attentively observe the teacher and students to watch and listen to how the teacher interacts with students during student work time. The coach models inquiry questioning techniques with the teacher to use when responding to student questions and to determine where students are in their problem solving process.

Taking Notes

What: **Notes are an important tool for coaching practice and informally document the coaching work.**

Why: Referring to notes is more reliable than memory for debriefing purposes. Notes also allow you to refresh your memory from session to session, track teacher growth, and document your own coaching. This promotes follow-through for teachers and coaches.

How: Take notes on what happens during classroom implementation, take notes on what the teacher does, what the students are doing, the words they are using. Set up a notetaking system that works for you. Keep notes organized so that you can easily review them. Jot down notes during or immediately after the lesson. Preserve teacher's confidentiality by keeping notes in a secure location and, if digital, protect them with a password.

When to Interject

What: Identify appropriate moments to share additional, relevant information.

Why: Teachers are still developing their CS teaching skills and capacities. Sometimes the coach may have additional information or relevant CS Teacher Moves or Strategies that could elevate the current CS learning opportunity.

How: **If something (a fact, anecdote, or CS teacher move) comes to mind that can greatly enhance that CS learning opportunity; identify a proper time to share it with the class, without disrupting the teacher's flow.**

CS Teacher Moves

CS Teacher Moves are used by teachers and coaches to help improve instruction while teaching CS. These cards have explicit moves to create a culture of engaged CS learning and may be used at any time by a coach throughout the cycle.

These cards are specific actions to be used as reminders and introduce concrete methods for improving CS teaching and routines.

tech.cornell.edu/impact/k-12


BETA

Asking for Help

What: **Implement norms/routines/processes that give students guidance and support in asking for help**

Why: Having specific norms for asking for help in CS classes allows students to take action and influence their own learning.

How: Plan a strategy for students to visually signal their need for help. Things like hand signals, cue cards, colored sticky notes, cups, traffic light cards, emojis, or writing their name on the whiteboard will help students manage their needs.

Call and Response

What: **Choose a call and response phrase to introduce students to interesting CS concepts.**

Why: Use a call phrase that triggers a response to gain students' attention during CS lessons. This redirects the class to focus on the teacher and secretly exposes students to CS topics.

How: **Create a call phrase to expose students to people, things, and events associated with the field of CS. Use the call and response as a cue to stop working and focus on the teacher. For example, When I say "Ada", you say "LoveLace."**

Celebrate CT Approaches

What: Give praise for practicing CT approaches.

Why: When the teacher praises the approaches to problem solving, it reinforces the mindset and work ethic students should have during CS lessons.

How: **Display the approaches so that students are continuously reminded of the behavioral mindsets they should be using. Celebrate the perseverance you see, the collaboration, creativity, and tinkering techniques. Consider awarding CT badges or stickers for students engaging in CT approaches frequently.**

Create CT Cues

What: **Create and use nonverbal cues to communicate CT strategies a student can use for problem solving.**

Why: Nonverbal cues to students allow them to move forward on an assignment without a loss of momentum.

How: A cue may be a gesture, a picture, or a sound that is intended to redirect students to perform a CT skill. This might be a visual in the classroom, or hand motions, such as: decompose, by using ASL sign language for "break apart" by pulling two hands apart; or reevaluate, using thumbs down or sideways.

Facilitating Student Talk

What: **Encourage and teach students to listen to and learn from other students.**

Why: If the teacher paraphrases, the class will lose accountability for listening and thinking on their own.

How: When you ask a student to answer a question in front of other students, do not repeat or rephrase their response. If it needs to be clarified or repeated have another student rephrase a prior students answer. Teach students a routine of add on or build on to the prior student's response.

Getting Unstuck

What: **When students get stuck, they need to develop strategies to move forward on a problem.**

Why: Explicit strategies for problem solving helps students regulate their emotions, continue their learning and persist when they become stuck and unable to move forward in their work.

How: **Encourage a student to step away from the problem and take a short brain break. Have the student collaborate with a peer to describe what they want their program to do, where they feel stuck, and then have them read their code aloud. To become unstuck, ask the peers to collaborate, tinker, debug, and persevere.**

Hands Behind Back

What: During student work, the teacher should assist students without touching their work.

Why: When teachers touch student work by editing their code or taking control of their keyboard, students tend to back away from ownership of their work.

How: Teachers can remind themselves not to touch student work by placing their hands behind their back. In addition, teachers can teach students to use this strategy when they are providing peer assistance.

Hands Off

What: **Teachers use a keyword or hand signal to get students' attention.**

Why: "Hands Off" is a management tool for moving students from screen work to listening to the teacher.

How: Create a call and response such as "hands off", to bring their attention to the teacher and as a signal for students to adjust their screens. Students respond by turning a tablet or phone face down on the desk, putting laptop screens at a 45 degree angle known as “clamshell”, “Pac-Man”, or “half mast". Make sure students are "hands off" and remove their hands from the keys. Younger students can put their hands in the air, on their heads, or on their laps.

Least to Most Prompting

What: When helping a student solve problems, gradually increase the level of help you provide.

Why: Promote students to be independent learners. In CS, there is a tendency for teachers to over-prompt and over-scaffold. This strategy helps you meet students where they are.

How: **When a student first asks for help, refer them to classroom resources, respond with a question, or give them a generalized prompt. If the student continues to ask for help, gradually increase the amount of help you give, saving things like direct instructions as a last resort. (M. Israel, et al.)**

Lingering Questions

What: **Elicit lingering questions about an assigned task.**

Why: Prior to students moving on to an independent or collaborative activity, engage students in identifying any lingering questions they may still have to ensure they know what they are expected to do.

How: Strategies for eliciting lingering questions include asking "What questions do you have?" and directing all students to write down one question anonymously. Then, draw questions to answer or take a question from one student, having other students respond with the answers.

Reflection

What: **Build in time for a CS/CT reflection.**

Why: Reflection strategies include having students share out strategies to solve a problem, writing a daily reflection, drawing a picture, or completing an exit ticket.

How: **Build time into the lesson for the students to synthesize new concepts and skills. Reflection strategies include having students share out strategies to solve a problem, student writing in a daily reflection, draw a picture, or complete an exit ticket.**

Student Deliverables

What: Support student monitoring their own progress towards the daily deliverable.

Why: To create independent learners, build student strategies to help students achieve lesson goals.

How: Make student expectations clear by posting the "deliverable" of a lesson visibly on the whiteboard and assignment sheet. Refer to the written expectations when a student asks what they are supposed to do!

The Hook

What: **Get students active and engaged right way with a short hook activity.**

Why: A hook in the beginning of a lesson helps students transition and improves motivation by engaging in fun experiences related to the learning objectives.

How: Start a lesson with an engaging activity that lasts a few minutes. Hooks may activate prior knowledge and be used to preview new concepts. Hook structures and format are designed to be highly engaging for students in your classroom.

The Power of Paraphrasing

What: **Paraphrasing helps clarify what students are understanding.**

Why: Students' understanding of CT and CS concepts may quickly be checked by asking a student to paraphrase what they are expected to do.

How: **Check for understanding. Have students repeat back their understanding of a CS concepts or instructions for a task. If one student doesn't have a clear understanding of what to do, there is a chance that many do not.**

Think Time

What: **Use "Think Time" strategies to engage all students.**

Why: Think Time allows all students to participate and raises rigor for all students.

How: Develop a thinking culture in your classroom by adopting Think Time routines. Provide all students wait time to think independently. Think time routines include, "Think, Pair, Share", "Everybody Writes", "Predict, Observe, Explain (POE)"

Value Every Student's Ideas

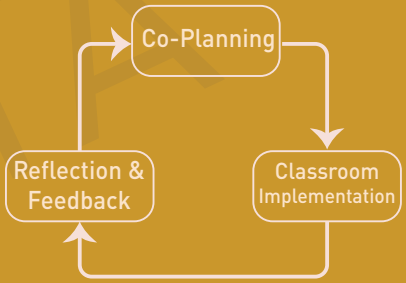
What: Acknowledge and value each student's ideas, questions, and reasoning.

Why: Due to time constraints, pacing, or class momentum, a teacher may not be able to tend to each student's question or explore that student's reasoning in depth.

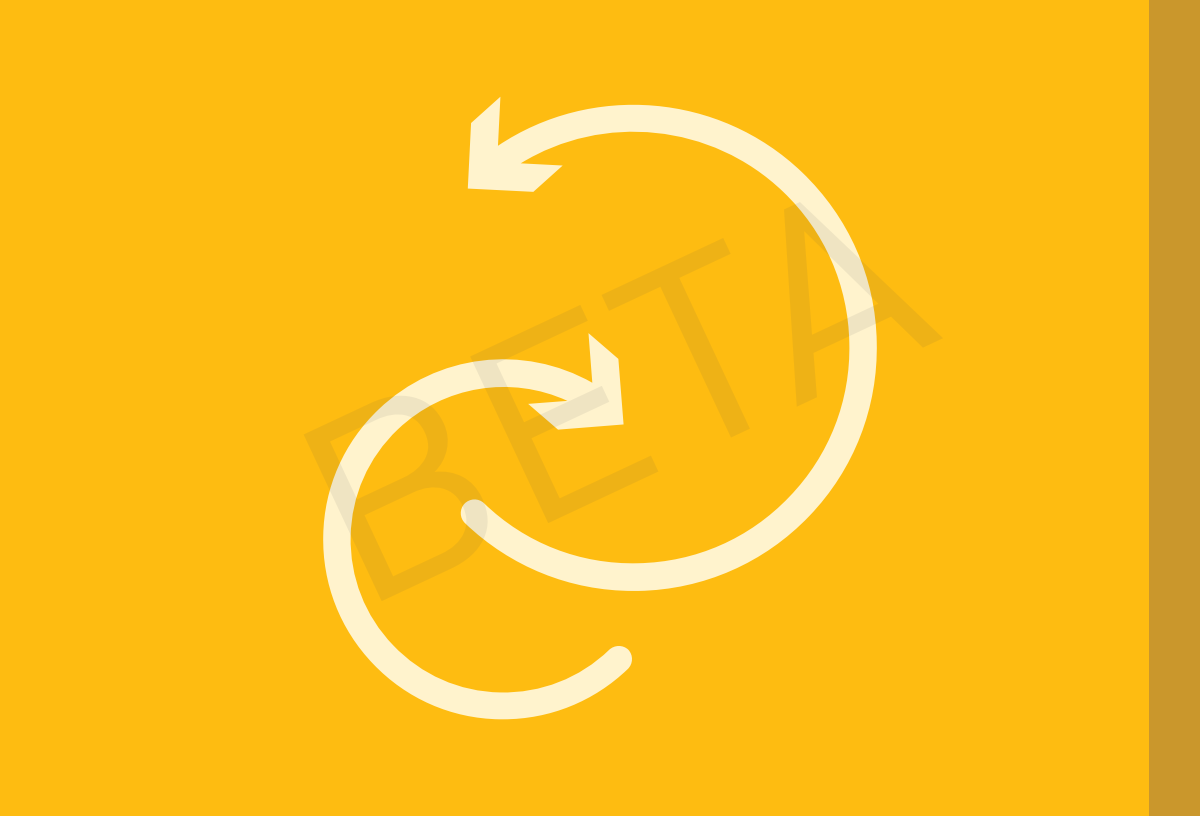
How: **Seek opportunities to explore a student's thinking through journaling, prompts, and in large or small groups to accomplish two goals: 1) acknowledge that the student's idea is valued and 2) allow students to contribute to the refining of each other's ideas and reasoning collaboratively.**

Reflection & Feedback

Reflection and feedback is a time when both the coach and teacher communicate about a lesson taught and then collaboratively design a set of actions or next steps to take during the next co-planning or teaching session.



Coaching Cycle


BETA

5 Whys

What: **Ask 5 questions that all start with "Why?"**

Why: This helps you go deep into exploring underlying causes of a specific problem.

How: Explain the 5 "Why" strategy to teachers and use it with them routinely. Repeat the question "Why?" to get at the root of an issue. Each answer forms the basis of the next question.

Action Steps

What: **Identify specific actions to take next.**

Why: Defining a specific action is a key component of coaching and leads to ensuring continuous improvement.

How: **Both the teacher and coach leave the session with specific steps to take prior to the next lesson or session. Use clear language in the action steps and identify what the teacher will do and what the coach will do.**

Active Listening

What: **Engage in active listening.**

Why: Understanding the teachers' perspective is important to a collaborative coaching relationship.

How: Listen to the teachers reflection about how the CS lesson went. Ask the teacher to expand what they are saying by asking them clarifying questions, or prompting them provide a bit more detail, or an example. Validate their point of view by verbalizing this back to them.

Brainstorming

What: **Brainstorm creative solutions to improve lesson delivery and student outcomes.**

Why: Brainstorming collaboratively leads to creative solutions and nourishes a collaborative relationship.

How: Identify an area of the lesson delivered, or classroom scenario to revise and improve. Collaboratively jot down ideas of possible solutions. Purposefully hold back on sharing ideas until the teacher has shared a few. Then add on to the teacher's ideas. Finally choose a solution and make a plan to test the solution during the the next lesson.

Closing a Debrief Session

What: Close a debrief session with self-reflective questioning.

Why: Closing questions help to solidify and confirm teacher understanding.

How: Check for understanding using questions: What did you learn from our conversation? What are some big takeaways? What would you do differently next time?

Coaching Report

What: **A report based on the debrief can be shared with the administration.**

Why: Because coaching is a significant investment of time and resources for schools, some administrators may want to keep a record of the coaching program. Coaching is non-evaluative, and for growth to occur, trust must be maintained in the coaching relationship. It is important to find ways to meet the competing needs of different stakeholders.

How: Create a short write-up that states the coaching activities the teacher participated in and the topics that you and the teacher discussed without any element of evaluation. Create this document with the teacher or send it to the teacher to check over prior to sharing it with administrators. Alternatively, send a weekly email outlining who you worked with, general themes, and student "wins" from the week.

CS Teacher Growth

What: Teachers evaluate their own personal progress using the field's teaching standards.

Why: CS teacher standards are clearly defined as a resource for developing CS teachers. Self monitoring in CS teacher standards achievement is key in goal setting and self-assessment.

How: **Review and reflect on the CS Teacher Standards and state standards. Have the teacher self-assessment their growth, independently or with you.**

From Critique to Question

What: **Ask the teacher how they will improve something specific rather than telling them to do so.**

Why: Empower the teacher to take personal responsibility to improve their craft.

How: Use a question that starts with 'How might we?" to guide teachers toward approaching a particular area for improvement. Discuss ideas from CS Teaching Strategies or Moves. Construct a plan with the teacher to implement an action during the next lesson.

Guide on the Side

What: Investigate using questioning to address and unravel issues.

Why: Guided questions are asked to lead a teacher to reflect upon a broader picture when tackling an issue in a lesson or classroom.

How: Introduce an issue through an investigation. As a coach, guide an investigation to address and resolve an issue with increasingly more specific questions. Lead teachers to identify their own understanding of an issue and have them arrive at a strategy to fix it.

I, the Learner

What: **Identify where you may have been confused as a learner.**

Why: Emphasizing your misunderstanding will create an opportunity to see how the students may also been confused.

How: If a teacher is not presenting themselves well, express where you may have been confused if you were a learner. Then dig into how the teacher may fix it.

Just the Facts

What: Directly address in a sensitive way when something does not go right in the classroom.

Why: Sometimes difficult conversations must be had about teaching style, equity, and teacher moves.

How: Share truthfully what you observed in the classroom, without ascribing motives or blame. Prioritize the most important problem to address. Frame this as an opportunity for the coach and teacher to solve together.

Lean on External Sources

What: **When providing feedback, lean on an external source.**

Why: This reinforces feedback and makes it feel less personal, removing potential tension.

How: Refer to the CS teacher standards, a school leader or an evidence based strategy to point out how a particular learning or teaching strategy may be applied. Encourage the teacher to explore piloting this strategy with their students.

Not About You

What: **By eliminating "you" from feedback, it comes across as less personal.**

Why: The teacher can sometimes hear constructive feedback as criticism of their own character, which gets in the way of professional growth.

How: **Offer constructive feedback about the work, not the person. "The handout does not match the correct vocabulary we talked about" sounds different than "you did not use the correct vocabulary on this handout."**

Opening a Debrief Session

What: **Initiate a debrief session with self-reflective questioning.**

Why: Allow teachers to share and unpack their thoughts before listening to your feedback. This allows them to reflect without being swayed by your feedback first.

How: **Give teachers a lot of time to think about and talk out what happened during the lesson without jumping to solutions. Ask questions like: How do you feel the lesson went? What do you think went well? What happened during the each part? How are you feeling about the students' understanding?**

Plan to Iterate

What: **Lessons may not work for every classroom. Prepare teachers to iterate based on reflections.**

Why: Sometimes you think you have a solid plan for a lesson until you teach it and realize that something is off. That's okay, as long as you reflect on it and make a plan to correct it.

How: **When a lesson doesn't go as planned, identify what went wrong and make a plan to revise, reteach, or modify. Discuss and think of the students who might have struggled with this lesson. What worked well for them? What didn't?**

Professional Learning Roadmap

What: Support the teacher in creating a personalized plan for professional learning.

Why: The field of CS continues to evolve. Teachers of all experience levels will need to continue to develop. Self assessment allows the teacher to create a roadmap for professional growth and learning.

How: **Using teacher standards as a resource helps teachers see areas for growth to plan professional learning. Expose teachers to external opportunities including conferences, communities of practice, workshops, books, and research opportunities.**

Recognition & Praise

What: **Praise teachers for successes big and small.**

Why: Praising teachers helps build relationships with them. It also boosts their confidence to keep expanding on their practice.

How: Praise the effort that got them to the result. Recognize them for the work they put into getting the result. Be specific, be realistic. Praise something obscure that they do not hear all of the time. Let your praise stand alone, not connected to a favor or constructive criticism. Praise people behind their backs, to their colleagues.

Repair Relationships

What: When necessary, you must work to actively repair relationships.

Why: Nobody is perfect. People don't always say or do the right thing. People have different lenses for interpreting meaning that can lead to misunderstanding. Or a mistake could have been made that leads to hurt feelings. It does not matter, it is important for you to take initiative to address feelings and maintain these professional, learning relationships.

How: **Notice if a teacher becomes "closed off", withdrawn, or passive aggressive. Rather than letting these feelings fester, take steps to address them. First examine your own feelings and implicit biases and reflect on your actions as the coach. Then have an open conversation with the teacher. Depending on your professional relationship, this might include sharing your own experiences and feelings, doing a personal check-in, surfacing cultural differences, reviewing roles, or apologizing and making amends when a misstep has occurred.**

Start Strong

What: Build on strengths to continue to improve all practices.

Why: Using the teacher's strengths is an effective way to encourage them to use their strengths to keep growing.

How: **In order to make improvements in the teacher's work, look at the things that went well and ask them how they can transfer what went well to an area that did not go well.**

Student Work

What: **Examine students' work samples from the lesson.**

Why: Examining student work lets you gauge student understanding and misconceptions.

How: Look for patterns in the work from all students or examine a sample of students' work from all levels, looking for common misconceptions and areas of strengths. The results of this should inform future planning.

Taking the Temperature

What: **Consider the teacher's emotional receptiveness.**

Why: Coaching isn't just about what you say, but knowing when and how to say it. Coaching is most effective when teachers are mentally and emotionally flexible.

How: **Gauge the teacher's receptiveness in the moment. Use this information to guide the conversation including how and when you provide different types of feedback, how much to push thinking and skills, or if you need to pause for a personal check-in.**

Teacher Goals

What: Have the teacher self-select goals to focus on.

Why: Clear goals help you gauge the teacher's understanding and promote growth in the teacher's learning.

How: Have the teacher self-assess their growth toward any existing goals. After reflecting on the lesson, ask the teacher what areas for growth they want to focus next time and/or together review the teacher standards to pick goals to focus on over time.

Teacher Talk > Coach Talk

What: **The teacher should be speaking more than you during a reflection session.**

Why: Learning and growth occurs from active participation and conversation.

How: Pay attention to talk time. Who is speaking more? Create room for the teacher's voice through asking open-ended questions and allowing think time. The teacher should be engaging in thinking work with you as a guide.

Wear Their Shoes

What: **Acknowledge the teacher, the whole person!**

Why: Having empathy for the teacher's situation and personal experiences is important to building an understanding and trustworthy relationship.

How: **Having empathy means being able to put yourself in someone else's shoes. Recognize the personal and professional demands that may affect their teaching. This will help you better identify judgments they may be making or feelings they are having.**

Write it Down

What: **Keep a shared document with the main points of the reflection that both you and the teacher can access.**

Why: When used as a communication tool, shared notes can be referred back to and can provide continuity and transparency for coaching sessions.

How: This is a separate document from your coaching notes or your raw observation notes. Create a shared document where you and the teacher record goals, action items, teacher takeaways and any feedback.