Northern Lights Collaborative
for Computing Education

CSLearning AID Checklist:

A Research-Informed Tool for Inclusive K-8 CS Integration

COLLEGE OF EDUCATION AND HUMAN DEVELOPMENT

Authorship and Acknowledgments

The **Northern Lights Collaborative for Computing Education** team was responsible for synthesizing the initial criteria, grounding the checklist in research, and coordinating iterative refinements. Their work guided alignment with Universal Design for Learning (UDL), culturally responsive pedagogy, and computer science education frameworks.

Justin Cannady designs inclusive computing learning experiences for educators and students. He previously served as Senior Manager of Online Learning at the National Math and Science Initiative (NMSI), where he led the creation of professional learning for 6,000+ educators. A former secondary math and computer science teacher, Justin holds degrees from the University of Montevallo and the University of West Florida, and is a National Board Certified Teacher.

Jennifer Rosato leads the Northern Lights Collaborative at the University of Minnesota, advancing inclusive computing education across the state. With over 20 years of experience in K–12 and higher education, she has directed multiple grants and projects to broaden participation in computing. She is a CSforAll-MN steering committee member and former chair of the CSTA Board of Directors.

Paul Schonfeld supports school districts in building equitable K–12 computer science pathways through coaching, professional development, and district planning. He previously taught computer science, physics, and engineering for nine years and coached robotics. He holds degrees in Physics/Geology (Lawrence University) and Professional Development (MEd, University of Wisconsin–La Crosse).

Tom Cozzolino works with schools and districts to expand equitable and engaging K–12 computer science programs. Over his 14-year career, he has taught elementary and middle school, facilitated MakerSpaces, and led after-school robotics. He holds a Master's in Differentiated Instruction, helping him scaffold computing for diverse learners and integrate CS into other subject areas.

Acknowledgements

Additional colleagues provided ongoing feedback through informal reviews, conference presentations, and collaborative workshops. Their insights kept the checklist relevant, teacher-friendly, and adaptable.

Liz Walsh, *Learning Manager*
Raspberry Pi Foundation

Christa Treichal, *President*
Cooperative Ventures

Jean Salac, *Assistant Professor of Computer Science*
Carleton College

Megan Dieger, *Researcher and Program Evaluator*
Deiger Research & Evaluation Consulting

Cindy Miller, *Library Media Specialist and Computer Science Lead*
Duluth Public Schools

Funding Support

This material is based upon work supported by the National Science Foundation under Award No. 2401154. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Suggested Citation

Cannady, J., Rosato, J., Schonfeld, P., & Cozzolino, T. (2025). CSLearning AID Checklist: A Research-Informed Tool for Inclusive K-8 CS Integration (White paper). *Northern Lights Collaborative for Computing Education, University of Minnesota.* https://northern-lights.umn.edu/resources

Table of Contents

Executive Summary

The Need. In elementary and middle school, integrating CS into other subjects has great potential to broaden participation in computing (Liao et al., 2024). Early CS exposure can boost student interest and confidence, especially for learners from historically marginalized communities (Madrigal et al., 2020). However, many young students still lack high-quality CS learning opportunities due to limited curricula, lack of inclusive content, and insufficient teacher support (McGill & Thigpen, 2024). This gap disproportionately affects girls, students of color, those with disabilities, and low-income students.

Our Solution: CSLearning AID Checklist. The CSLearning AID Checklist (formerly the K-8 CS Lesson Checklist (Cannady, 2025)) is introduced as a practical response to these challenges. It is an evaluative tool with 45 clear criteria organized into five categories that together address lesson **C**omposition, **S**tudent **Learning**, **A**ssessment, **I**nclusion, and **D**igital Accessibility (**CSLearning AID**). The checklist provides research-based guidance to help educators design or select lessons that are equitable, culturally responsive, and aligned with best practices.

Development and Theoretical Grounding. This white paper outlines the checklist's rigorous development process, which was iterative and collaborative. It draws on established frameworks, including Universal Design for Learning (UDL) (Israel et al., 2022), culturally responsive-sustaining pedagogy (Kapor Center, 2021), and the Teacher Accessibility, Equity, and Content (TEC) Rubric (Weintrop et al., 2020), so that the criteria represent proven strategies for inclusion and high-quality CS integration. Multiple phases of refinement, with input from experts and teachers, led to a tool that is both grounded in theory and educator-approved.

Structure and Use. Readers will find an explanation of the checklist's structure demonstrating how to apply the checklist to real CS-integrated lessons. These examples illustrate how the criteria can highlight strengths or reveal areas for improvement in lesson plans, ensuring lessons are inclusive of diverse student identities and abilities.

Implementation Guidance. The document provides targeted guidance for different stakeholders (teachers, curriculum developers, PD facilitators, and researchers) on how to implement the checklist in practice. Whether it's for lesson planning, curriculum design, or teacher training, the checklist is presented as a flexible tool for improving teaching practice and student outcomes in CS.

Future Directions. A preview of upcoming initiatives is included, such as the development of a supportive toolkit, a bank of vetted CS-integrated lessons, and expanded professional development offerings. The white paper concludes with concrete recommendations for adopting the checklist and advancing equity-focused CS integration in K-8 education.

Introduction

Early introduction of computing concepts in K-8, especially when integrated with core subjects, has been shown to increase students' interest and confidence in computer science (Liao et al., 2024). This is particularly impactful for students from underrepresented groups, helping to inspire them and build foundational skills before high school (Grabarczyk et al., 2022). Early integrated CS experiences can lead to better outcomes in later CS courses, creating a pipeline of more diverse students pursuing computing.

Despite growing awareness, significant inequities remain in access to quality CS education. Historically marginalized groups face systemic barriers to participation in CS learning (McGill et al., 2022). These barriers include stereotypes about who belongs in CS, a lack of inclusion and sense of belonging in CS classrooms, and a scarcity of engaging, culturally relevant curriculum materials. In K-8 settings, CS is often offered sporadically or not at all in the curriculum, leaving many students without exposure.

Educators are crucial to bridging these gaps, but they often lack support and training for teaching CS in equitable ways. Many K-12 CS teachers feel poorly equipped to employ culturally relevant pedagogy and believe that too few existing CS curricula are culturally relevant to their students (White et al., 2025). Educators at the elementary level have limited CS content knowledge and struggle to find age-appropriate, interdisciplinary CS lessons (Liao et al., 2024). Time constraints and accountability pressures in other subjects further complicate efforts to integrate CS into daily teaching (McGill & Thigpen, 2024). These factors contribute to hesitation or inconsistency in implementing CS-integrated lessons.

The lack of accessible, high-quality instructional resources is a commonly cited barrier to CS integration (Margulieux et al., 2024). Teachers and curriculum designers need concrete guidance on what *good* CS-integrated lessons look like. The community has called for tools that simplify the process of identifying or developing such lessons, so that even educators new to CS can confidently include computing in their teaching. In short, a gap exists in providing educators with an easy-to-use framework to evaluate and improve lesson plans for equity and inclusion in CS.

To address these needs, the **CSLearning AID Checklist** was developed as a solution grounded in research and educator feedback. It is designed to fill the resource and guidance gap by providing a structured set of criteria that define key elements of inclusive, high-quality CS-integrated lessons. By using this checklist, teachers and curriculum developers can incorporate lessons that are aligned with best practices in accessibility and equity, thereby promoting broader participation in computing. The checklist serves as a practical aid to simplify lesson selection or creation, reduce the burden on teachers, and promote equitable access to CS learning opportunities. This white paper will detail the background of the checklist's creation, its contents, and how it can be effectively utilized in K-8 education.

Development Process and Theoretical Foundations

Developing the **CSLearning AID Checklist** required balancing two priorities: grounding the tool in established research and ensuring it was usable for classroom educators. From the outset, the design team aimed to merge theory with practice by starting with frameworks that define inclusive, high-quality instruction, and then engaging teachers and experts to confirm that the criteria were practical, relevant, and easy to apply. The development process was therefore both research-informed and iteratively co-designed, moving through multiple phases of testing and refinement to produce a checklist that is comprehensive, reliable, and classroom-ready.

Research-Based Foundations

The development of the CSLearning AID Checklist was anchored in established educational frameworks and research on inclusive pedagogy. The project team began by examining existing evaluation rubrics and theories to inform the criteria. Key foundations included:

- *Teacher Accessibility, Equity, and Content (TEC) Rubric:* a rubric for evaluating computing curricula that emphasizes teacher support, equity, and content alignment (Weintrop et al., 2020).
- *Evaluative Rubric for CT Integration:* a research-based rubric for integrating computational thinking in elementary curricula, contributing criteria related to how well lessons blend CS concepts with other subject standards (Sullivan et al., 2022).
- *Teacher Beliefs and Pedagogy Research:* insights from studies on teachers' beliefs about coding and computational thinking helped shape criteria addressing teacher mindset and readiness, ensuring the checklist's usability for those new to CS (Rich et al., 2021).
- *Universal Design for Learning:* principles that guide the design of instruction to meet the needs of all learners. UDL inspired criteria for providing multiple means of engagement, representation, and expression in CS-integrated lessons. Criteria were added to help make

sure learning goals were student-friendly and to encourage using students' first languages or varied modalities in instruction (Israel et al., 2022).

- *Culturally Responsive-Sustaining Pedagogy:* Drawing on the Kapor Center's Culturally Responsive-Sustaining CS Education Framework and related equity literature, the checklist incorporates practices that value students' cultural backgrounds, promote inclusion, and challenge inequities. This theoretical grounding shaped the Equity and Inclusion criteria, ensuring lessons honor student identities and provide culturally relevant contexts (Kapor Center, 2021).

Iterative Co-Design Approach

The checklist was not created in isolation. It went through multiple phases of design and refinement, with continuous input from educators and experts to ensure relevance and clarity. Each phase of development built upon the last, expanding the checklist's scope and improving its usability.

- **Phase 1 - Initial Development.** The Northern Lights Collaborative team synthesized 45 preliminary criteria from research and existing rubrics. These were initially framed as binary checkpoints and grouped into three broad categories reflecting accessibility, equity, and content. Four curriculum experts reviewed and rated each criterion's importance, leading to a streamlined draft of 41 criteria. This version established a foundation but required further validation in classroom contexts.
- **Phase 2 - Expanding Inclusion and Structure.** Guided by UDL and culturally responsive pedagogy, the criteria that addressed student-friendly goals, multilingual supports, and diverse learner accommodations. The pilot with ten K-8 teachers (representing urban, suburban, and tribal schools) highlighted redundancy and challenges in locating where certain criteria applied. In response, the checklist was reorganized into five categories:

Composition, Student Learning, Assessment, Inclusion, and *Accessibility.* This restructuring simplified the tool, clarified scope, and aligned criteria with lesson components teachers naturally consult.

- **Phase 3 - Pilot Testing and Final Refinement.** The near-final checklist was applied to four diverse CS-integrated lessons spanning subjects and grade levels. The Northern Lights Team used the checklist on each lesson to test clarity and consistency. This process identified wording refinements and led to further precision. For example, splitting a broad prerequisite knowledge criterion into separate items for subject-area content and CS concepts. Importantly, the *Accessibility* category was refined into *Digital Accessibility*, capturing formatting and design practices essential for learners using assistive technologies. By the end of Phase 3, the checklist was finalized with 45 criteria across the five categories, tested for usability and efficiency (most lessons could be reviewed in under 30 minutes).

Throughout all phases, continuous feedback loops were used to keep the checklist relevant and teacher-friendly. Experts, practicing teachers, and evaluators influenced revisions at every stage, balancing theoretical grounding with practical classroom realities. The result is a research-informed, classroom-tested checklist that reflects both the rigor of academic frameworks and the lived expertise of educators.

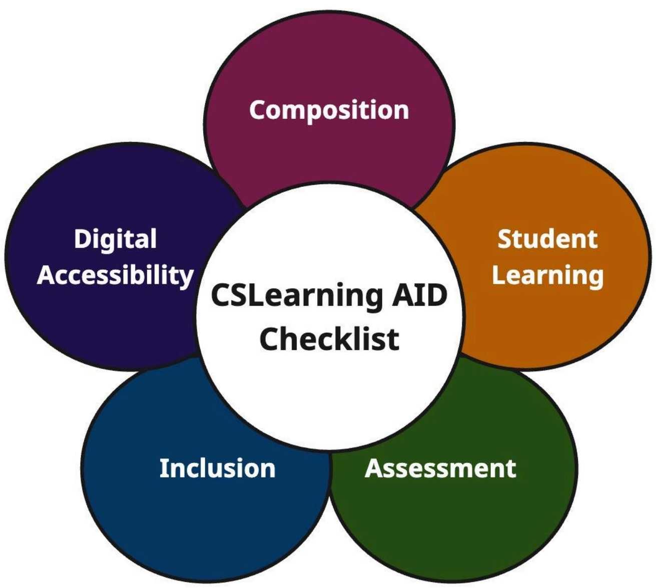
CSLearning AID Checklist

The CSLearning AID Checklist is designed as a practical tool that bridges research and classroom realities. Rather than serving as an abstract framework, the checklist is meant to be picked up and used directly by educators, reviewers, and curriculum developers. Each item reflects careful refinement through multiple cycles of design, testing, and evaluation. By foregrounding accessibility, inclusion, and clear alignment to lesson components, the checklist supports teachers in designing and analyzing lessons with an eye toward both equity and usability. The following section introduces the organizational categories that emerged through this process. The complete checklist is included in **Appendix A** for direct use by educators and reviewers.

Categories

The resultant **CSLearning AID Checklist** consists of 45 criteria, organized into five primary categories. Each category represents a critical dimension of a well-designed CS-integrated lesson. The criteria are phrased as concise checkpoints that collectively paint a picture of the lesson's quality in terms of content, pedagogy, and inclusion. By checking a lesson against all 45 criteria, educators can evaluate its strengths and identify areas for improvement in a systematic way. The five categories of the checklist are:

Composition. Composition criteria focus on lesson structure and organization necessary for effective teaching. These criteria address measurable learning objectives, grade-level identification, standards alignment, materials, definitions of key terms, prerequisite knowledge, lesson outline and timing, references, assessments, and student-facing elements such as "I Can" statements, directions, and activity guides. Together, these checkpoints verify that the building blocks of a lesson are in place, complete, and usable by both teachers and students.



Student Learning. The Student Learning category focuses on the quality of the learning experience. Criteria examine whether lessons include questioning prompts, address common misconceptions, and provide scaffolding that gradually builds understanding of computing concepts. They also check for accommodations and modifications, the use of varied instructional strategies, and opportunities for student collaboration. These elements collectively support student engagement and learning during instruction.

Assessment. Assessment criteria verify that a lesson includes multiple and varied ways for students to demonstrate learning. This category covers universally designed assessments, summative and formative assessments, and the presence of rubrics. It also looks at whether students have opportunities to apply skills in authentic contexts, use computing terminology, communicate ideas through open-ended prompts, engage in peer feedback, and reflect metacognitively on their learning. By meeting these criteria, lessons promote accountability for student outcomes while also encouraging deeper engagement with computing.

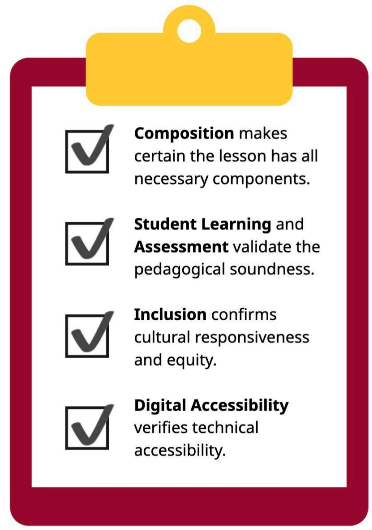
Inclusion. Inclusion criteria address whether lessons embody principles of equity and culturally responsive teaching. They ask whether lessons provide multiple entry points for engagement, authentic real-world contexts, and opportunities for students to incorporate their identities, interests, and cultures into the learning process. They also check for representation of diverse cultures and perspectives, strategies that support multilingual learners, and extensions for students who meet expectations and are ready for deeper exploration. These criteria help make sure that CS-integrated lessons welcome every student to participate and thrive.

Digital Accessibility. With Digital Accessibility, lesson materials follow accessibility best practices for digital documents and presentations. Criteria include consistent heading styles, unique slide titles, numbered or bulleted lists, appropriate typography (sans serif, adequate size and spacing), sufficient color contrast, descriptive alt text for images, captions for videos, informative link text, properly structured data tables, and logical reading order on slides. By embedding these practices, lessons become accessible to learners with diverse needs and align with broader commitments to inclusive design.

Checklist Structure

Binary Criteria with Limited Exceptions. Each of the 45 criteria is reviewed using a set of options: *Yes, No, Partial, N/A - Lesson Type,* and *N/A - Missing Component*. The development team's intention, however, was to keep the tool as close to a binary Yes/No format as possible. The additional options were introduced to handle practical review situations: *N/A - Lesson Type* was used when certain criteria did not apply (for example, a lesson devoted solely to a unit project without standalone slides); *N/A - Missing Component* captured cases where materials were incomplete (such as missing assessments or student guides); and "Partial" was used sparingly, usually to indicate disagreement between reviewers or to flag criteria where the indicators needed refinement. By anchoring evaluation in Yes/No judgments but allowing these limited exceptions, the checklist balances clarity with the flexibility required for real-world lesson review.

Criteria, Indicators, and Notes. To support consistency and transparency, each checklist item is structured around three elements: the criterion (a concise checkpoint), the indicators (examples of what meeting the criterion looks like), and space for reviewers to mark whether the criterion was met and provide notes. This structure encourages reviewers to ground their judgments in evidence rather than impression, while leaving space to document nuances, clarify partial fulfillment, or capture contextual considerations. The five categories work together to create a holistic evaluation. Taken together, the **CSLearning AID Checklist** provides a systematic, classroom-tested way to identify both the strengths and areas for growth in CS-integrated lessons.



Implementation Guidance

The **CSLearning AID Checklist** was designed with multiple audiences in mind, and its practical value lies in the different ways it can be taken up by classroom teachers, curriculum developers, professional learning providers, and researchers.

- For **classroom teachers**, the checklist can serve as both a planning and reflection tool. During lesson planning, it helps teachers identify missing elements such as explicit CS learning goals, formative checks for understanding, or multilingual supports. After instruction, the same criteria can be used for reflection, allowing teachers to iteratively refine lessons and set professional learning priorities. The checklist also provides support when evaluating third-party materials, surfacing gaps such as insufficient scaffolds for English learners or limited assessment of CS concepts. By highlighting these areas, teachers can make targeted adaptations rather than needing to overhaul entire lessons. Furthermore, the Inclusion and Digital Accessibility categories encourage proactive planning for diverse learners by emphasizing multiple modalities, extension activities, and structured opportunities for collaboration.
- For **curriculum developers**, the checklist can function as a set of design specifications. Using the criteria as requirements during development ensures that inclusion and

accessibility goals are addressed from the start. In addition, the checklist supports quality assurance by enabling systematic review of materials, ensuring internal consistency across units, and providing traceability between criteria and specific lesson artifacts. For existing materials, the tool can guide retrospective improvement, helping designers focus on targeted updates such as adding student-friendly objectives, accessible formats, or formative assessment prompts without requiring a full redesign.

- For **professional development providers**, the checklist is particularly powerful as a shared framework for practice-based learning. Workshops can be structured around the criteria, asking participants to analyze sample lessons, identify gaps, and propose concrete revisions. Instructional coaches and professional learning communities can use the checklist to structure observations and peer reviews, creating a shared vocabulary for equity, accessibility, and quality in lesson design.
- Finally, for **researchers**, the checklist offers an operational framework for describing and analyzing instructional materials. It can be used to characterize the features of CS-integrated lessons, examine relationships between lesson design and teacher practice, or explore associations between specific lesson features and student outcomes. In this way, the checklist provides not only a practitioner-facing tool but also a research instrument with potential for advancing the evidence base on equity-focused CS integration.

Future Directions

Looking ahead, the Northern Lights Collaborative plans to create supporting resources for the **CSLearning AID Checklist**. One planned effort is the development of a companion toolkit, which will include templates, examples, and an interactive version of the checklist linked to improvement resources for criteria marked "No." In addition, a curated repository of vetted CS-integrated lessons annotated against the checklist will provide educators with ready-to-use models and enable comparative study of instructional practices.

Professional learning opportunities are also being considered for development. Institutes, online courses, and mentoring networks are envisioned to help educators apply the checklist in varied contexts and to build long-term capacity for inclusive CS instruction. Parallel to these practitioner supports, ongoing research will continue to test the checklist in classrooms, gather data from its use, and refine its criteria to align with evolving standards and best practices. Finally, the checklist may inform broader policy and systems alignment by serving as a model for district-or state-level processes around lesson adoption, review, and equity-focused instructional improvement.

Conclusion and Recommendations

The **CSLearning AID Checklist** provides a concise, research-informed framework for evaluating and improving K-8 CS-integrated lessons. By organizing 45 criteria into five categories, it reduces ambiguity for teachers, supports curriculum developers in designing accessible and equitable materials, and offers professional learning providers a common language for coaching and training. Ultimately, the checklist is intended to promote consistent, high-quality practices that expand equitable participation in computing.

To maximize its impact, several recommendations emerge. Curriculum providers should consider incorporating the criteria directly into development workflows and quality assurance processes, so that inclusion and accessibility are non-negotiable from the outset. Schools and districts can encourage routine use of the checklist in collaborative planning and materials selection, embedding equity checks into everyday practice. Professional development providers can situate practice-based learning around the checklist and associated examples, helping teachers learn not only the *what* but also the *how* of equitable lesson design. Finally, researchers can use the criteria to describe instructional materials, inform study designs, and build new evidence on the relationship between inclusive design and student outcomes.

The **CSLearning AID Checklist** is more than an evaluative tool. It can serve as a bridge between research and practice, offering educators concrete guidance for designing lessons that are rigorous, culturally responsive, and accessible to all learners. Its adoption at multiple levels of the education system has the potential to transform K-8 CS integration from a patchwork of uneven opportunities into a coherent, equity-driven movement that broadens participation in computing for every learner.

References

Cannady, J. L. (2025). Iterative Development of an Evaluative Checklist for Integrated Computer Science Lessons in K-8. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*, 1403–1404. https://doi.org/10.1145/3641555.3705210

Grabarczyk, P., Nicolajsen, S. M., & Brabrand, C. (2022). On the Effect of Onboarding Computing Students without Programming-Confidence or -Experience. *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research*, 1–8. https://doi.org/10.1145/3564721.3564724

Israel, M., Kester, B., Williams, J. J., & Ray, M. J. (2022). Equity and inclusion through UDL in K-6 computer science education: Perspectives of teachers and instructional coaches. *ACM Transactions on Computing Education*, *22*(3), 1–22. https://doi.org/10.1145/3513138

Kapor Center. (2021). *Culturally responsive-sustaining computer science education: A framework*.

Liao, Y.-C., Kim, J., Ottenbreit-Leftwich, A. T., Karlin, M., & Guo, M. (2024). Voices of Elementary Computer Science Teachers: Computer Science Integration Rationales and Practices. *ACM Trans. Comput. Educ.*, *24*(4), 43:1-43:26. https://doi.org/10.1145/3688854

Madrigal, V., Yamaguchi, R., Hall, A., & Burge, J. (2020). Promoting and Supporting Computer Science Among Middle School Girls of Color: Initial Findings from BRIGHT-CS. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 247–253. https://doi.org/10.1145/3328778.3366855

Margulieux, L. E., Liao, Y.-C., Anderson, E., Parker, M. C., & Calandra, B. D. (2024). Intent and Extent: Computer Science Concepts and Practices in Integrated Computing. *ACM Trans. Comput. Educ.*, *24*(3), 35:1-35:23. https://doi.org/10.1145/3664825

McGill, M. M., Davis, S., & Reyes, J. (2022). Surfacing Inequities and Their Broader Implications in the CS Education Research Community. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, 294–308. https://doi.org/10.1145/3501385.3543969

McGill, M. M., & Thigpen, L. (2024). Extrinsic Barriers to Integrating Computer Science in Elementary School Subject Areas in the United States. *Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research*, 1–10. https://doi.org/10.1145/3677619.3678116

Rich, P. J., Larsen, R. A., & Mason, S. L. (2021). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education*, *53*(3), 296–316. https://doi.org/10.1080/15391523.2020.1771232

Sullivan, F. R., Duan, L., & Pektas, E. (2022). Design of an evaluative rubric for CT integrated curriculum in the elementary grades. *6th International Conference on Computational Thinking Education and STEM Education (CTE-STEM)*. https://doi.org/10.34641/ctestem.2022.475

Weintrop, D., Coenraad, M., Palmer, J., & Franklin, D. (2020). The teacher accessibility, equity, and content (TEC) rubric for evaluating computing curricula. *ACM Transactions on Computing Education*, *20*(1), 1–30. https://doi.org/10.1145/3371155

White, S. V., Koshy, S., & Scott, A. (2025). Advancing culturally responsive-sustaining computer science through K-12 teacher professional development strategies. *Journal of Research on Technology in Education*, *57*(1), 163–177. https://doi.org/10.1080/15391523.2024.2399280

Appendix A: CSLearning AID Checklist

ID	Criteria	Description	Indicators
Composition			
CN01	**Learning Objectives**	The lesson plan contains clear, measurable learning objectives (lesson goals).	- The lesson plan includes measurable objectives that clearly define what students should know or be able to do by the end of the lesson. - The objectives use Bloom's verbs (e.g., 'analyze,' 'create,' 'demonstrate') to articulate clear expectations. - The objectives are aligned with standards and can be assessed at the end of the lesson.
CN02	**Grade Level**	The lesson plan identifies the grade level.	- The grade level is explicitly identified in the lesson plan (e.g., 'This lesson is designed for Grade 3').
CN03	**Standards**	The lesson plan identifies the national and/or state standards.	- The lesson explicitly lists the relevant national and/or state standards (e.g., CSTA, NGSS, Common Core) aligned with the lesson's objectives and activities. - The standards are clearly labeled and mapped to the specific grade level.
CN04	**Materials Needed**	The lesson plan provides a list of the materials needed.	- The lesson includes an itemized list of all materials and quantities needed for both the teacher and students. - Presentation aids and student materials are accessible from the lesson plan. - The list specifies any digital or physical resources required, including links to online resources if applicable.
CN05	**Content Definitions**	The lesson plan includes definitions of content concepts to reduce confusion related to terminology.	- The lesson provides clear definitions of non-CS content concepts (e.g., emotional literacy, historical events, scientific terms) that students need to understand to succeed in the lesson. - The lesson does not rely on external resources for definitions.
CN06	**CS Definitions**	The lesson plan includes definitions and examples of CS concepts to support teacher learning.	- The lesson includes clear, well-explained definitions of CS-related concepts (e.g., loops, conditionals, variables) tailored to support teacher understanding of the computer science content. - Definitions are accompanied by examples or sample code to make the CS concepts tangible and easier to understand for teachers.

ID	Criteria	Description	Indicators
CN07	**Content Prior Knowledge**	The lesson plan outlines any prerequisite content knowledge or skills needed for the lesson.	- The lesson plan clearly lists any non-CS content knowledge or skills that students should have before starting the lesson (e.g., familiarity with basic mathematical operations, and understanding of specific literary works). - The lesson plan suggests ways to review or refresh these content areas if necessary.
CN08	**CS Prior Knowledge**	The lesson plan outlines any prerequisite CS knowledge or skills needed for the lesson.	- The lesson outlines CS knowledge or skills students need beforehand (e.g., 'Students should be familiar with basic Scratch programming, including using motion and looks blocks'). - Prerequisites are detailed to allow teachers to gauge whether students require a refresher. - The lesson plan suggests ways to review or refresh these content areas if necessary.
CN09	**Suggested Implementation / Outline**	The lesson plan contains suggested implementation for how an educator should teach the lesson.	- The lesson plan includes a step-by-step outline or script for how to teach the lesson (e.g., introduction, guided practice, independent work). - Provides instructional strategies and methods (e.g., group discussions, demonstrations, questioning techniques).
CN10	**Timing**	The lesson plan includes a timeline or estimated duration for each section of the lesson.	- The lesson includes estimated durations for each section (e.g., 'Warm-up: 5 minutes; Activity: 20 minutes; Discussion: 10 minutes').
CN11	**References**	The lesson plan contains related references, attributions, and acknowledgements (if needed).	- The lesson plan includes attributions for any external resources, research, or materials used (e.g., books, websites, articles). - Acknowledgments are provided for any contributors, tools, or collaborations used in developing the lesson. - Proper citations are included for materials used in accordance with copyright or open-source licenses.
CN12	**Content Assessment**	The lesson plan contains assessment(s) that provide teachers opportunities to assess CS and subject-area student learning.	- Formative and summative assessments are included to check for understanding throughout and at the end of the lesson.

ID	Criteria	Description	Indicators
CN13	**Alignment**	The lesson plan provides an alignment between each assessment and learning goals (and/or standards).	- Each assessment is explicitly aligned with the lesson's learning goals and/or standards (e.g., 'This project assesses students' ability to use loops, aligned with CS Standard 3.3').
CN14	**Student Learning Goals**	The slides or student activity guide state the learning goals in student-friendly language (I Can statements).	- The student activity guide or slides clearly state the learning goals in student-friendly language using 'I Can' statements.
CN15	**Directions**	The lesson provides clear and simple directions for each activity or task.	- Students are provided with clear and concise directions for each activity or task. - The instructions are straightforward and simple to allow students to easily follow along and understand the steps required to complete each task.
CN16	**Navigation**	The student activity guide contains headings for each activity or task that align with the slides so that students can easily follow along with the lesson.	- The student activity guide contains well-defined headings for each activity or task that align with the corresponding slides.
CN17	**Notes**	The student activity guide has sections for students to make notes, answer questions, or reflect on their learning.	- The student activity guide includes dedicated sections for students to make notes, answer questions, or reflect on their learning.
Student Learning			
SL01	**Question Prompts**	The lesson includes student questioning and discussion prompts.	- The lesson includes open-ended student questioning and discussion prompts that stimulate critical thinking (e.g., 'How might changing this variable affect the outcome of the program?').
SL02	**Misconceptions**	The lesson provides explanations about common misconceptions and challenges that students have regarding the concepts.	- The lesson identifies common misconceptions and challenges related to the concepts taught, provides explanations or clarifications to address these misconceptions, and includes strategies or tips to help students overcome anticipated difficulties.

ID	Criteria	Description	Indicators
SL03	**Lesson Scaffolded**	The lesson is scaffolded to help students gradually build their understanding of computing concepts.	- The lesson is scaffolded to progressively build understanding (e.g., starting with guided examples before moving to independent projects). - Activities move from concrete to abstract.
SL04	**Accommodations / Modifications**	The lesson contains suggested accommodations, modifications, and other strategies to scaffold concepts for students.	- The lesson includes specific accommodations and modifications to support English Language Learners, students with disabilities, and other diverse learners (e.g., extended time, alternative assignments).
SL05	**Instructional Strategies**	The lesson includes at least 2 different instructional strategies (e.g. discussions, modeling, student activities, projects, etc.).	- The lesson incorporates at least two different instructional strategies (e.g., direct instruction, group discussions, hands-on activities).
SL06	**Collaboration**	The lesson includes an opportunity for students to collaborate.	- The lesson includes opportunities for student collaboration (e.g., pair programming, and group projects). - Collaborative activities are structured with clear expectations and roles.
Assessment			
AS01	**Universally Designed**	The lesson contains assessments that are accessible for all students, amenable to accommodations, and provides simple, clear, and intuitive instructions.	- Assessments are accessible to students of all abilities. - Instructions for the assessment are clear and straightforward. - The lesson plan includes guidance for accommodations. - The programming environment required for the assessment is accessible or has accomodations listed.
AS02	**Summative**	The lesson contains at least one summative assessment that measures all CS and non-CS learning objectives listed (not necessarily the learning standards).	- The lesson includes at least one summative assessment that measures all CS and non-CS learning objectives. - The tasks within the summative assessment are not vague or lack clear criteria for evaluating student performance.

ID	Criteria	Description	Indicators
AS03	**Rubrics**	The lesson plan contains a rubric for each summative assessment that help teachers assess student proficiency.	- The lesson plan includes a rubric for each summative assessment, with criteria for evaluating student proficiency at multiple levels (e.g., exemplary, proficient, developing). - Rubrics are easy to understand and include specific, measurable criteria that align with the learning objectives. - The rubrics cover both CS and non-CS skills and knowledge.
AS04	**Formative**	The lesson contains formative assessments that measure CS and non-CS learning objectives (not necessarily the learning standards).	- The lesson incorporates multiple formative assessments throughout. - Formative assessments are generally aligned with CS and non-CS learning objectives. - The assessments are varied in format (e.g., quizzes, discussions, in-class activities).
AS05	**Application**	The lesson provides opportunities for students to apply the skills that are being taught in both CS and non-CS content.	- The lesson provides practical activities that allow students to apply both CS and non-CS skills. - Application tasks are connected to the lesson's learning objectives. - Activities include opportunities for problem-solving, creative projects, or hands-on experiments that demonstrate applied learning.
AS06	**Terminology**	The lesson prompts students to use CS-specific vocabulary during the lesson or during an assessment.	- The lesson prompts students to use and correctly apply CS-specific and non-CS vocabulary during activities and assessments. - Vocabulary usage is integrated into discussions, written responses, or project-based tasks.
AS07	**Communicating Ideas**	The lesson provides opportunities for students to demonstrate their understanding through responding to open-ended prompts.	- The lesson includes open-ended prompts that allow students to express their understanding and reasoning in their own words. - Multiple formats (written, oral, digital) for communicating ideas are provided.
AS08	**Peer Feedback**	The lesson provides opportunities for students to receive peer feedback.	- The lesson provides structured opportunities for peer feedback, where students can review and comment on each other's work. - Clear criteria and guidelines are provided to help students give constructive and focused feedback.

ID	Criteria	Description	Indicators
AS09	**Reflection**	The lesson provides opportunities for students to reflect on their learning at a metacognitive level (i.e., reflection about the learning process, not the content).	- The lesson incorporates reflection activities that prompt students to think metacognitively about their learning process and progress. - Structured reflection opportunities (e.g., journals, discussion prompts) encourage students to identify challenges, successes, and areas for future improvement.
Inclusion			
IN01	**Multiple Entry Points**	The lesson provides multiple entry points to build knowledge and making meaning (e.g., storytelling, kinesthetics, and problem solving).	- The lesson provides various ways to access and engage with the content (e.g., storytelling, kinesthetic activities, problem-solving). - The lesson offers clear alternative pathways for students who struggle with the primary mode of instruction.
IN02	**Authentic***	The lesson provides authentic connections to real-world applications and/or real-world contexts.	- Real-world contexts and applications are integrated in the lesson. - Authentic examples and scenarios help students see the practical value of the skills being taught. Examples used in the lesson are not contrived, disconnected, or abstract.
IN03	**Student Identity**	The lesson provides opportunities for students to elevate their voice by including their identity (including isolated and intersectional), interests, and/or culture in their learning.	- The lesson includes activities that encourage students to incorporate their own identities, interests, and cultural backgrounds. - Students are given opportunities to share personal experiences or perspectives that relate to the content.
IN04	**Cultural Inclusion**	The lesson highlights the diverse cultures, perspectives, languages, and/or community values.	- The lesson highlights diverse cultures, perspectives, and community values through examples, discussions, and materials. - Content is presented in a way that respects and validates different cultural backgrounds. The content is not tokenistic or perpetrating cultural stereotypes.

ID	Criteria	Description	Indicators
IN05	**Multilingual Supports**	The lesson includes at least one strategy that supports multilingual learners.	- The lesson plan includes strategies to assist multilingual learners and language supports are integrated into the lesson. - Strategies should be easy to locate and pervasive throughout the materials.
IN06	**Extensions**	The lesson provides extensions that allow a deeper understanding of topics for students who meet the performance expectations.	- The lesson includes extension activities or challenges to provide additional content without disrupting the core lesson.
Digital Accessibility			
DA01	**Heading Styles**	The lesson plan and student activity guide use heading styles.	- The lesson plan and student activity guide use heading styles. - Heading styles should be consistent throughout documents.
DA02	**Unique Titles**	The slides use unique slide titles.	- Slide titles are unique (not repeated) and descriptive, accurately reflecting the content of each slide.
DA03	**Lists**	The lesson plan, slides, and student activity guide use numbered and bulleted lists where possible.	- Numbered and bulleted lists are used when applicable throughout the lesson plan, slides, and student activity guide to organize information clearly.
DA04	**Typography**	The lesson plan, slides, and student activity guide use a sans serif font, 1.5 line spacing, text left-aligned when possible. Font is at 11pt (docs) and 18pt (slides).	- All materials use a sans serif font with 1.5 line spacing and left-aligned text, ensuring clear and accessible content. - Font sizes meet or exceed the recommended standards (11pt for documents, 18pt for slides). - Font use and formatting are consistent throughout all of the lesson materials.
DA05	**Contrast Ratio**	The lesson plan, slides, and student activity guide use a contrast ratio of at least 4.5:1.	- The lesson plan, slides, and student activity guide consistently use a contrast ratio of at least 4.5:1 to ensure the text is legible against the background.

ID	Criteria	Description	Indicators
DA06	**Alt Text**	The lesson plan, slides, and student activity guide contain alt text for all images.	- Every image included in the lesson plan, slides, and student activity guide is accompanied by descriptive alt text. - Alt text is concise, accurate, and clearly conveys the purpose or content of the image for users relying on screen readers.
DA07	**Captions**	The lesson contain only videos that have captions available.	- All videos included in the lesson materials have accurate and synchronized captions.
DA08	**Link Text**	The lesson plan, slides, and student activity guide contain link text that is descriptive, informative, and indicate if a download action will be triggered.	- All links in the lesson materials are accompanied by descriptive text that clearly indicates their purpose or destination. - Link text is informative and provides context (such as if a download will occur) so that users know what to expect.
DA09	**Data Tables**	The lesson plan, slides, and student activity guide use tables to organize data, not for visual layout.	- Tables are used solely for organizing data with clear headers and logical structures that aid in readability.
DA10	**Reading Order**	The reading order for each slide is in the order it is meant to be presented.	- The content on the slides is arranged in a logical reading order that reflects the intended sequence of the lesson.