Computer Science (CS) Implementation Guidance

Landscape of Computer Science | **CS for AZ: Leaders** | **CS Toolbox: Curricula and Tools**

Document prepared by Janice Mak

Note: The following document contains examples of resources for elementary school, middle school, and high school. These resources are not explicitly recommended or endorsed by the Arizona Department of Education. The absence of a resource from this does not indicate its value.

Table of Contents

Overview and Purpose

The primary purpose of this document is to introduce LEAs to resources that support implementation of the new Arizona K-12 Computer Science Standards. Whether integrating CS and computational thinking across the curriculum or adopting it as a stand-alone course, there is a need to consider CS implementation within the K-12 system. As such, resources and guidance are outlined in the sections below that address the needs of the following stakeholders: school/LEA leadership; counselors and educators. An additional section includes considerations when adopting CS curricula and tools.

Introduction - Landscape of Computer Science in Arizona

Computing underpins every other science, technology, engineering, and mathematics (STEM) and non-STEM fields as a highly versatile and sought-after skill-set that is essential in today's information economy. As the foundation for all computing, computer science is defined as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications and their impact on society" (Tucker et al., 2006, p. 2).

This demand drives the economy and creates a direct impact on the job market; the U.S. Department of Labor estimates 1.1 million computing-related job openings in the U.S. by 2024 (Department of Labor Bureau of Labor). However, more than two-thirds of these jobs could go unfilled due to the insufficient pool of college graduates with computing related degrees[1].

As computing has become an integral part of our world, public demand for computer science (CS) education is high. Nine in ten parents would like their child's school to offer CS (Gallup, 2015). In fact, most Americans believe CS is a critical skill to learn (Horizon Media, 2015). Many of today's students will be using CS in their future careers across all fields (Change the Equation, 2015). There is a need, therefore, to provide equitable access and opportunity for every student, equipping them with the CS and computational thinking skills in their K-12 education. This necessitates thoughtful and intentional planning of inclusive CS learning pathways. At the early grades from K-2, CS might include integrated lessons that can be taught with (plugged) or without (unplugged) a computer. Unplugged lesson examples can be found here, here, and here. From grades 3-8, CS may continue to be addressed in an integrated way with other subject areas or taught as a part of a

[1] Bureau of Labor Statistics has additional information on occupational projections to 2024 https://www.bls.gov/opub/mlr/2015/article/occupational-employment-projections-to-2024.htm

STEM, technology, or computer class. At the high school level, CS may start off with an exploratory type of course that may then lead to Advanced Placement (AP) or Career and Technical Education (CTE) options in CS.

Arizona Challenge[2]

- Arizona currently has 9,301 open computing jobs (2.7 times the average demand rate in Arizona).
- The average salary for a computing occupation in AZ is $87,434, which is significantly higher than the average salary in the state ($48,160).
- The existing open jobs alone represent a $813,227,447 opportunity in terms of annual salaries.
- Arizona had only 814 computer science graduates in 2017; only 16% were female.
- Only 1,108 exams were taken in AP Computer Science by high school students in Arizona in 2018 (487 took AP CS A and 621 took AP CSP).
- Only 24% were female (21% for AP CS A and 26% for AP CSP); only 252 exams were taken by Hispanic or Latino students (69 took AP CS A and 183 took AP CSP); only 23 exams were taken by Black students (4 took AP CS A and 19 took AP CSP); only 3 exams were taken by American Indian or Alaska Native students (2 took AP CS A and 6 took AP CSP); 1 exam were taken by Native Hawaiian or Pacific Islander students.
- Only 65 schools in AZ (20% of AZ schools with AP programs) offered an AP Computer Science course in 2017-18 (12% offered AP CS A and 15% offered AP CSP), which is 13 more than the previous year.

To address these challenges, there have been efforts within the state to increase the number of K12 CS teachers and supporting CS teachers with high-quality curricular and professional development support. The State Board of Education adopted K-12 Arizona Computer Science Standards in October 2018 and a new CS endorsement pathway in May 2019. There is a need now to further move the trajectory of CS education forward in the state by building the capacity of educators in applying CS content, skills, and practices to their teaching. In order to build this capacity, schools and local

[2] Data is from Conference Board for job demand, the Bureau of Labor Statistics for state salary and national job projections data, the College Board for AP exam data, the National Center for Education Statistics for university graduate data, the Gallup and Google research study Education Trends in the State of Computer Science in U.S. K-12 Schools for parent demand, the 2018 Computer Science Access Report for schools that offer computer science

education agencies will need to adopt a broad framework to provide all students with access to high quality opportunities to learn CS.

Recommendations

Given the landscape of computer science in Arizona, there are four recommendations to prioritize when providing equitable access to K-12 CS learning opportunities. The four recommendations, adapted from State of the States Landscape Report: State-Level Policies Supporting Equitable K-12 Computer Science Education (2017), include the following:

- **Build a broad base of leadership and ownership among key stakeholders.** In order to implement a sustained and scaleable CS initiative, the process must be developed by and informed by an inclusive representation of stakeholders who are committed to ensuring access and opportunity for every student K-12 to have a high quality CS education.
- **Develop short-, medium-, and long-term strategies, with a view to coherence and sustainability.** Ensuring input from diverse stakeholders will help to develop long-term goals that are supported by actionable short- and medium-term strategies. It is also critical to note the importance of administration, school boards, and district leadership in ensuring ongoing support for implementation (e.g. financial support, scheduling).
- **Use data to monitor progress, inform decision-making, and drive continuous improvement.** In order to ensure that local education agencies (LEA) are broadening participation to K-12 CS education in a way that increases access to every student, it is critical to identify and use data to inform the development of their strategies and goals. Suggested sources of data include parent input surveys, student and teacher input, enrollment in CS courses at schools across all grade levels within the LEA, College Board data for the AP exam, and state test results.
- **Leverage expertise.** Local education agencies can make use of a plethora of resources to guide their implementation of CS. Organizations such as the National Center for Women and Information Technology (NCWIT) offer support for increasing access to CS to every student by engaging various stakeholders including school counselors in the efforts of broadening participation. CSforAll has developed the Strategic CSforALL Resource & Implementation Planning Tool (SCRIPT) to guide LEAs as they work toward implementing a CS vision.

A Brief Introduction: Arizona K-12 Computer Science Standards

Arizona's K-12 Computer Science Standards are based on the concepts and practices defined in the K-12 Computer Science Framework. The Framework presents a vision of students who are "computationally literate creators who are proficient in the concepts and practices of computer science" (K-12 Computer Science Framework).To be computationally literate in this sense means to know computer science, as distinguished from computer literacy[3], educational technology[4], digital citizenship[5], and information technology[6]. Computer science is focused on understanding how and why computers work, the creation of new technologies, and the impact of these creations on society. In order to fulfill this vision, therefore, it is critical to deepen educators' knowledge of and capacity for teaching computer science. This includes developing proficiency in the computer science concepts[7] that underlie what they teach. Educators also engage in computational thinking[8] and professional computer science practices[9]. They model these practices as they support their students to do the same.

While Arizona's K-12 Computer Science Standards are grade-level specific, it is helpful for educators to not only demonstrate fluency with computer science concepts for the relevant grade band, but also possess familiarity with both the preceding and following grade bands[10]. Concepts include[11]: Computing Systems[12], Networks & The Internet[13], Data &

[3] **Computational literacy:** the general use of computer and programs (e.g. computer applications, productivity software, Internet search, digital presentation)
[4] **Educational technology:** the application of computer literacy to school subjects (e.g. collaborative editing of an online document)
[5] **Digital Citizenship:** the appropriate and responsible use of technology (e.g. select an appropriate password)
[6] **Information technology:** the industrial applications of computer science (e.g. installation of software)
[7] Computer science concepts are defined in the K-12 Computer Science Framework.
[8] **Computational thinking:** The human ability to formulate problems so that their solutions can be represented as computational steps or algorithms to be executed by a computer. [Lee, 2016]
[9] Computer science practices are defined in the K-12 Computer Science Framework.
[10] For example, a middle grades computer science teachers should demonstrate deep knowledge of the concepts taught in grades 6-8, and they should also have familiarity with the progression of concepts from upper elementary to early high school.
[11] Computer science concepts are defined in the K-12 Computer Science Framework. Standards aligned to the Framework, such as Arizona Computer Science Standards, CSTA K-12 Computer Science Standards, and other comprehensive state-adopted standards, articulate the current body of knowledge and skills.
[12] **Computing Systems:** People interact with computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively.
[13] **Networks & the Internet**: Networks connect computing devices to share information and resources, and facilitate innovation by providing fast, secure communication.

Analysis[14], Algorithms & Programming[15], Impacts of Computing[16]. CS educators also apply and model professional CS and computational thinking practice in flexible and appropriate ways. Practices[17] include: Fostering an Inclusive Computing Culture[18], Collaborating Around Computing[19], Communicating About Computing[20], Recognizing and Defining Computational Problems[21], Developing & Using Abstractions[22], Creating Computational Artifacts[23], and Testing and Refining Computational Artifacts[24].

---

[14] **Data & Analysis:** Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions. As the amount of digital data expands, so does the need to process data effectively.

[15] **Algorithms & Programming:** Algorithms and programming empower people to communicate with the world in new ways and solve compelling problems. The development process involves choosing which information to use and how to process and store it, breaking large problems into smaller ones, recombining existing solutions, and analyzing different solutions.

[16] **Impacts of Computing:** Computing affects the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions.

[17] **Practice:** "The seven core practices of computer science describe the behaviors and ways of thinking that computationally literate students use to fully engage in today's data-rich and interconnected world" K-12 Computer Science Framework

[18] **Fostering an Inclusive Computing Culture:** The classroom teacher is personally able to incorporate perspectives from people of different genders, ethnicities, and abilities. To do this, educators must first understand the personal, ethical, social, economic, and cultural identities/contexts in which they personally operate.

[19] **Collaborating Around Computing**: Work effectively with colleagues to plan and reflect on lessons and to create complex artifacts. Collaboration requires educators to navigate and incorporate diverse perspectives, conflicting ideas, disparate skills, and distinct personalities.

[20] **Communicating About Computing:** Communicate with diverse audiences about the use and effects of computation and the appropriateness of computational choices. Write clear comments, document work, and communicate ideas using precise language and multiple forms of media.

[21] **Recognizing & Defining Computational Problems**: Define problems, break them down into their component parts, and evaluate each part to determine whether a computational solution is appropriate.

[22] **Developing & Using Abstractions:** Identify patterns and extract common features from specific examples to create generalizations, in order to simplify the development process and manage complexity.

[23] **Creating Computational Artifacts:** Create artifacts that are personally relevant or beneficial to their community and beyond, by combining and modifying existing artifacts, or by developing new artifacts. Examples of computational artifacts include programs, simulations, visualizations, digital animations, robotic systems, and apps.

[24] **Testing & Refining Computational Artifacts:** Test and refine computational artifacts in a deliberate and iterative process. Respond to the changing needs and expectations of end users and improve the performance, reliability, usability, and accessibility of artifacts.

CS for AZ: Leaders

Vision

It is essential for LEA leaders to gather stakeholders to develop a broad base of ownership of the vision for CS within the organization. These stakeholders could include representatives from curriculum and instruction, information technology, career and technical education, parent group, educator group, library media specialists and community organizations. Together, the leaders with stakeholders can work to develop a vision for CS within their LEA that could include grappling with the following questions: Why do we want to teach CS/develop a CS pathway? What value does CS have for our students? Do we see the value of CS for every student?

Data Collection and Goal-Setting

Conducting a survey of the landscape of CS within the district is a potential next step. This could include taking inventory of all the CS teaching currently occurring within the LEA. Surveying elementary site leaders to see whether CS is being taught across all grades through a STEM elective or pull-out computer class, for example, provides critical information pertaining to access and opportunity for students to learn CS early on. Follow-up questions regarding whether every student at the school is receiving instruction in CS or a sub-population is also important to know in order to have the baseline information to expand opportunities to more students. At the high schools, dual-enrollment, AP course enrollment, and achievement data by demographics, gender, and socio-economic status are all critical to guide next steps as the LEA works to increase access to CS courses for all students. Joint Technical Education Districts (JTEDS) provide valuable data on students enrolled in and graduated from Career and Technical Education (CTE) coursework in computer science areas. Based on this, data tracking regarding which certificates students earned and what they are currently doing could be useful as well. Overall participation in CS across the LEA also yields useful information and can help the leaders set relevant participation goals. Data can also be collected from parent, student, and staff surveys with questions about their views of and experience with CS. Responses from these surveys will provide a context-specific perspective regarding the launch or development of CS pathways within the LEA.

Along with examining participation data, understanding teacher certification and endorsement requirements to teach CS is key in order to identify potential teachers who may be able to lead CS classes in the coming year. Closely connected with staffing is the need to examine the resources and funding within the LEA to support the initiation or expansion of CS. This includes various sources of Title funding to potentially support PD efforts, CTE funding, internet access, devices, and administrative support as the program develops. Example case studies from Code.org listed below and linked here serve as interesting starting points:

- Large Urban School District

- Medium sized Exurban District

- Rural School District

Based on the landscape data, the LEA may then set goals to reach targets in number and types of CS offerings, enrollment data, staffing goals, and fidelity of implementation. For example, if a middle school finds that CS is only taught to students who select the STEM or CS elective, then they might set a goal to seek ways that CS can be integrated into other curricular areas. Schools can use Bootstrap Algebra to integrate CS into algebra and Project Growing Up Thinking Scientifically (Project GUTS) to introduce modeling into science. Alternately, a school might find that the gender balance is skewed with many more male than female students. This school could set a goal to achieve CS class enrollment with balanced gender representation. In another school, data may indicate the desire of the parent and community to participate in CS efforts. This particular school may choose to partner with parents who are software engineers or community volunteers to bring CS to their setting and coach robotics clubs, host CS club meetings, or even teach particular skills in their classes.

Ultimately, the goal is to develop a CS program where all students regardless of grade level, gender, race and socioeconomic status are represented and achieving at comparable levels of CS achievement. Inherent to developing an inclusive CS program is to consider how diverse learners including English Learners and students with disabilities will access the course content and deliberate integration of tools and resources to support all students. Collecting and analyzing a robust base of data and input from the community will inform specific, context-based, and relevant goal-setting. Resources such as Guide to Inclusive Computer Science Education , You Can Actively Recruit a Diverse Range of Girls into High School Computing Classes, Computer Science Is for Everyone: A toolkit for middle and high schools to increase diversity in computer science education, and Bridging the Encouragement Gap in Computing are critical to ensure that stakeholders are equipped to broaden participation to computing pathways for all students.

Plan

Based on the values, vision, and landscape survey, the stakeholders can then work to plan for equitable access to CS pathways within the LEA. This includes establishing short-term, medium-term, and long-term milestones on the way to attaining goals within the areas of CS participation, CS course offerings, professional development opportunities, and funding.

An example template with additional guidance in the areas of establishing the CS landscape survey, supporting teachers with curriculum and professional development can be found in Appendix A.

CS Toolbox: Curricula and Tools

Elementary School

Organization	Description of Curriculum
BootUp	FREE project-based, interest-driven K-6 teacher lesson plans featuring block-based programming languages in Scratch and ScratchJr. BootUp's entire curriculum, with 100 lesson plans, is FREE.
Carnegie Learning	Zulama is a standards-aligned K-12 curriculum that gives students a strong digital foundation. Students ideate, research, prototype, and program their own computational apps, games, and simulations. Courses were written by faculty at Carnegie Mellon and MIT. All content is fully customizable, including assessments. Rigorous yet fun content that includes assessments, lesson plans, and more.
Code Avengers	Over 100 interactive junior lessons that teach computer science through storytelling. They weave in science, math and social studies to build broad and deep understanding of computational thinking and coding. $12.50 - $25/student. Bulk discounts available.
Code Monkey	More than 10 game and project-based courses that teach text-based coding in CoffeeScript and Python. Includes freestyle game and challenge-creation platforms as well as game design, cross-curricular and Hour of Code courses. Complimentary demos and teacher resources (including fully-detailed lessons). Starts at $10/student. Bulk-discounts available.
Codesters	3-lesson intro and project platform, FREE, 2 40-hr Intro. to Python courses and a 20-hr Intro to Game Design course, $20/student for the bundle (Python 1, Python 2, and Game Design) or a la carte pricing.
Computer Science Fundamentals	6 courses, 15-30 hrs each, blending online tutorials with “unplugged” activities, FREE.
Computer Science Foundations by Sphero	Computer Science Foundations by Sphero is a supplemental, standards-aligned curriculum designed to be taught in the classroom with our robots. Across 3 courses and 72 lessons,

	teachers and students will explore STEAM principles through creative coding. Courses 1-3 $499.99 or a la carte pricing for each course.
Google CSFirst	CS First provides free, easy-to-use computer science enrichment materials that engage a diverse student population in grades 4-8 (ages 9-14). Facilitators use the video content to teach kids coding basics with Scratch, a block-based coding tool. CS First is available online at http://g.co/csfirst and can be used by anyone, in any setting (in school, after school, or outside of school).
KinderLab Robotics	KIBO is the screen-free robot kit for kids that lets 4- to 7-year-olds create, design, decorate and bring their own robot to life! KIBO is an easy and fun way to bring robotics and coding to your young learners and spark their interest in STEAM. KIBO is available in 4 different robot kits ($199, $299, $399 and $499). Other items available include a la carte programming blocks and modules, curriculum and teacher materials. Compare KIBO robot kits!
Kodable	More than 150 easy to follow lessons that foster creativity and include gamified practice and creative design from Sequence to JavaScript and Swift. Teacher videos and resources explain new concepts before each unit. Progress tracking allows teachers to monitor CS standards and outcomes with weekly updates and reports. Pricing begins at $1,250/site and $30/student for classroom pricing. District Discounts available.
WeDo robotics LEGO Education	From simple drag and drop interfaces to developing more complex algorithms, teachers are empowered to encourage students to explore the world around them and develop their own ideas and theories, making science come to life through coding.
Lightbot	LightBot is a puzzle game based on coding; it secretly teaches you programming logic as you play!
PLTW Launch	PLTW Launch’s 28 interdisciplinary modules bring learning to life. The program empowers students to adopt a design-thinking mindset through compelling activities, projects, and problems that build upon each other and relate to the world around them. And as students engage in hands-on activities in computer science, engineering, and biomedical science, they become creative, collaborative problem solvers ready to take on any challenge.

ScratchEd	A 6-unit intro to Scratch, FREE.
Tynker	1 FREE Starter course, 6 Visual Coding courses, 1 Lego WeDo coding course, 6 STEM courses. Each visual coding course has around 16 1-hour lessons. STEM courses include coding projects and quizzes in English, Math, Science, and Social Studies. Prices vary, district pricing available.
UC Davis C-STEM Center	Integrating coding into math education. Coding, robotics, math activities, and teaching resources for grades 1-6. Coding in Blockly and user-friendly C/C++ Interpreter Ch. Free teaching resources available including PDF textbooks. School site licenses $600.
VidCode	First course and teacher resources are free. Over 12 creative coding courses, 10 hours each, that teach JavaScript and computational thinking. Includes cross-disciplinary, interaction design, and game design courses. Unlimited classroom accounts starting at $299/year https://www.vidcode.com/pay
Wonder Workshop	The Learn to Code curriculum for K-5 contains 6 levels. Each level has five 50-minute lessons and one Assessment lesson that is 2 hours for the first 2 levels and 2 days for the other levels. The Applied Robotics curriculum for K-8 contains 3 units. Each unit has 10 lessons that range from 40-60 minutes. Learn to Code Challenge Cards + Curriculum Guide: $99.99. Cue Applied Robotics Curriculum Guide: $79.99. Student Design Process Notebooks: $16.99. Code to Learn Cross-Curricular Library Subscription: $60

Middle School

Organization	Description of Curriculum
Bootstrap	Composed of 4 research-based curricular modules for grades 6-12(Algebra, Reactive, Data Science, & Physics) that can be integrated into Social Studies, Science, Math, Intro and even AP CS Principles courses. Materials reinforce core concepts from mainstream subjects, enabling non-CS teachers to adopt our introductory materials while delivering rigorous and engaging computing content.
CodeAcademy	Code Foundations, Computer Science, Data Science and more.
CodeHS	CodeHS helps schools and districts build a comprehensive Middle School to High School computer science pathway starting from introductory level block-based programming courses all the way to AP level text-based courses in many languages.
Codesters	Codesters combines a fun online coding platform for students, a powerful learning management system for teachers, and built-out coding lessons so you can start teaching kids to code in your school today.
Creative Computing Curriculum	The Creative Computing Curriculum, designed by the Creative Computing Lab at the Harvard Graduate School of Education, is a collection of ideas, strategies, and activities for an introductory creative computing experience using Scratch.
Computer Science Discoveries	Computer Science Discoveries is appropriate for 6 - 10th grade students and can be taught as a semester or year long introductory course (3-5 hours per week of instruction for 9+ weeks). The course takes a wide lens on computer science by covering topics such as programming, physical computing, HTML/CSS, and data. The course inspires students as they build their own websites, apps, games, and physical computing devices. Our curriculum is available at no cost for anyone, anywhere to teach.
Edhesive	Each a full-year blended course, Explorations in Coding I and II cover foundational concepts and skills of C and build understanding of programming through coding explorations, practices and projects in block-based and text-based Python. Supplemental

	online and unplugged activities. Starts at $500/site for up to 50 students; unlimited school license available.
Edison Robots	The Edison robot is a powerful, engaging tool for teaching kids STEM, computational thinking and computer programming in a hands-on way.
Finch Robot by BirdBrain Technologies	Inspire and delight students learning computer science from kindergarten to college with the programmable Finch Robot 2.0.
Google CSFirst	CS First provides free, easy-to-use computer science enrichment materials that engage a diverse student population in grades 4-8 (ages 9-14). Facilitators use the video content to teach kids coding basics with Scratch, a block-based coding tool. CS First is available online at http://g.co/csfirst and can be used by anyone, in any setting (in school, after school, or outside of school).
Grasshopper (Google)	Learn with fun, quick lessons on your phone that teach you to write real JavaScript.
Khan Academy Computing	A variety of free, easily accessible lessons, videos, and text-based resources at all levels. This curriculum can also take students to more mature levels of understanding by offering resources related to object-oriented programming, interactive web page development, simulation and game development, information theory, algorithms, and cryptography.
LEGO Mindstorms EV3	The LEGO MINDSTORMS Education EV3 Core Set comes with a curriculum pack and includes 48 tutorials to help you and your students learn the basics of LEGO MINDSTORMS Education EV3.The 48 step-by-step tutorials are designed to help educators and students master basic and advanced programming as well as hardware and data logging functions.
Microsoft MakeCode	Microsoft MakeCode brings computer science to life for all students with fun projects, immediate results, and both block and text editors for learners at different levels.

Middle Years Computer Science (MyCS)	How do computers work? What do computer scientists do? What does it take to make a computer or a computer program work? We answer these questions and more with MyCS: Computer Science for Beginners.
NCLAB	Karel Coding, 3D Modeling, and Python Courses
STEM: Explore, Discover, Apply National Integrated Cyber Education Research Center (NICERC)	STEM: Explore, Discover, Apply engages middle school students through a series of hands-on projects that help improve their problem-solving and critical-thinking, leadership, and team-building skills. All projects seamlessly integrate the engineering design process which allows students to creatively explore STEM through design.
PLTW Gateway	The PLTW Gateway curriculum is targeted for middle school students and comprises a number of engineering, biomedical, and CS units. The CS units include such concepts as algorithms (linear, conditional, and repetitive), abstraction, programming, and data, as well as tracing and debugging, pair programming, and computational thinking skills. Students who experience PLTW Gateway learn to become independent learners and engage in a curriculum that is hands-on and student-driven, as the teacher becomes a facilitator of learning. In addition to CS knowledge and skills, PLTW Gateway fosters transferable skills, including problem solving, persistence, creativity, collaboration, and communication. Average cost to train a teacher = $1200
Project Growing Up Thinking Scientifically (Project GUTS)	Project GUTS — Growing Up Thinking Scientifically — is an integrated science and computer science program for middle school students serving schools and districts internationally. Growing up thinking scientifically means learning to look at the world and to ask questions, developing and using computer models that help answer questions through scientific inquiry, and using critical thinking to assess which ideas are reasonable and which are not. To grow up thinking scientifically means knowing science to be a computing-rich, dynamic, creative endeavor, a way of thinking, rather than a body of facts.
UC Davis C-STEM Center	Integrating coding into math education. Coding, robotics, math activities, and teaching resources for grades 1-6. Coding in Blockly and user-friendly C/C++ Interpreter Ch. Free teaching resources available including PDF textbooks. School site licenses $600.

VidCode	First course and teacher resources are free. Over 12 creative coding courses, 10 hours each, that teach JavaScript and computational thinking. Includes cross-disciplinary, interaction design, and game design courses. Unlimited classroom accounts starting at $299/year https://www.vidcode.com/pay
Zulama Game Design Fundamentals	Created with faculty at Carnegie Mellon University's game design program, the Zulama Modern Learning Computer Science and Game Design Platform translates the University's professional Master's program to middle and high school. • 11 long courses • 9 short courses • 2 full-year computer science courses • Professional Learning • Assessments • Lesson Plans • Standards Correlations • Ongoing Support • Implementation Training

High School

Organization	Description of Curriculum
AP Computer Science A: College Board	AP Computer Science A introduces students to fundamental topics in CS, such as problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing. It is meant to be the equivalent of a first-semester course in CS. This engaging course underscores the importance of communicating solutions appropriately and in ways that are relevant to current societal needs.
Beauty and Joy of Computing	The Beauty and Joy of Computing (BJC) is an introductory computer science curriculum developed at the University of California, Berkeley, intended for non-CS majors at the high school junior through undergraduate freshman level. It was one of the five initial pilot programs for the AP CS Principles course being developed by the College Board and the National Science Foundation.
Bootstrap	Consists of four research-based curricular modules for grades 6-12(Algebra, Reactive, Data Science, & Physics) that can be integrated into Social Studies, Science, Math, Intro and even AP CS Principles courses. Materials reinforce core concepts from mainstream subjects, enabling non-CS teachers to adopt our introductory materials while delivering rigorous and engaging computing content
Carnegie Mellon University CS Academy (CMU CS Academy)	The future of the field depends on introducing students to computer science well before they reach college. Often, though, students have limited — or no — access to a CS curriculum in high school. The CS Academy aims to create an entirely free, online, interactive high school computer science curriculum. It comes with teacher training, an online interactive textbook, online technical support and more.
Carnegie Learning	Zulama courses were written by faculty at Carnegie Mellon and MIT. All content is customizable and includes assessments, lesson plans, and more. Example courses: Introduction to Computer Science Through Game Design Advanced Placement (AP) Computer Science Principles Unity 3D Programming.

CMU CS Academy	A novel, world-class, online, interactive high school computer science curriculum that is entirely free.
CS50: Harvard University	CS50 AP, Harvard University's introduction to the intellectual enterprises of computer science and the art of programming for students in high school, which satisfies the College Board's new AP CS Principles curriculum framework.
CS Principles: Code.org	A year-long course that can be taught as an AP or non-AP computer science course. Computer Science Principles introduces students to the foundational concepts of computer science and challenges them to explore how computing and technology can impact the world. More than a traditional introduction to programming, it is a rigorous, engaging, and approachable course that explores many of the foundational ideas of computing so all students understand how these concepts are transforming the world we live in.
Edhesive	3 full-year, blended courses: Intro to CS, AP CS Principles, AP CSA - Java; Intro CS also available as half-year course. Programming languages include Python, Scratch, Processing and Java. AP CSP is College Board-endorsed curriculum, and both AP CSP and AP CSA prepare students for the AP Exams.
Exploring CS	Exploring Computer Science is a year-long, research-based, high school intro-level computer science curriculum and teacher professional development program that focuses on broadening participation in computing.
Mobile CSPrinciples	Mobile CSP is a College Board-endorsed AP Computer Science Principles curriculum and professional development based on the theme of mobile computing.
NICERC Cyber and Computer Science	The National Integrated Cyber Education Research Center (NICERC) provides free classroom curriculum and PD opportunities for educators across the country. Currently, NICERC offers curricula for students in grades 6–12 on such topics as STEM (grades 6–8), cyber engineering (grades 9–11), physics and advanced mathematics (grades 11–12), and CS (grade 12).
PLTW High School Computer Science	Computer Science Essentials (CSE), Computer Science Principles (CSP), Computer Science A (CSA).

Computer Science Unplugged

"Computer science unplugged," or learning computer science concepts without a computer can be a very useful, engaging, and powerful way to engage students in learning computer science concepts and practices. The table below includes several resources to support teachers in integrating 'unplugged' CS lessons.

CS Unplugged https://csunplugged.org/en/ https://classic.csunplugged.org/	A collection of free teaching materials that teaches computer science through engaging activities that do not require the use of a computer. No programming is required to engage with these tasks.
Computer Science-in-a-Box: Unplug Your Curriculum https://www.ncwit.org/resources/computer-science-box-unplug-your-curriculum-2018-update	Computer Science-in-a-Box: Unplug Your Curriculum introduces fundamental building blocks of computer science -- without using computers. Use it with students ages 9 to 14 to teach lessons about how computers work, while addressing critical mathematics and science concepts such as number systems, algorithms, and manipulating variables and logic.
CS Fundamentals Unplugged https://code.org/curriculum/unplugged	A compilation of all the unplugged lessons to teach the fundamentals of computer science. Use any of these lessons as a stand-alone or as complementary lessons for any computer science course. Ages 4+
Programming Unplugged: Bridging CS Unplugged Activities Gap for Learning Key Programming Concepts (AlAmer et al., 2015) https://ieeexplore.ieee.org/document/7478218	Read about the integration of newly developed CS unplugged activities to teach key programming concepts to middle and high school girls.
Technocamps: Computer Science 101 - An Unplugged Introduction https://education.microsoft.com/technocamps	Skype lesson designed to introduce students to computer science through a series of thought-provoking questions and hands-on activities. This session is ideal as a starting point for any students before they begin coding, but also appropriate for those who are already coding.

Computational Thinking Resources

Computational thinking, or the thought processes involved in formulating problems so that their solutions their solutions can be represented as computational steps or algorithms that can be carried out by a computer (Cuny, Snyder, & Wing, 2010; Lee, 2016) is a practice highlighted in the K-12 Computer Science Framework. Computational thinking is most effectively taught in the context of computer science lies at the heart of the computer science practices, as seen in practices 3-6 (K-12 CS Framework).

Computational thinking in the K-12 CS Framework	Computational Thinking. https://k12cs.org/computational-thinking/ Chapter 5 - Practices Including Computational Thinking https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf
Computational Thinking for Educators	Computational Thinking Online Course for Educators https://computationalthinkingcourse.withgoogle.com/unit
Computational Thinking (Wing, 2006)	Article by Jeanette Wing https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf
Computational Thinking Competencies	This set of competencies are intended to help educators build the skills necessary to support all learners to be computational thinkers so that they can harness the power of computing to innovate and solve problems. Jointly developed by the International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA), these competencies are intended to support educators across all disciplines and with students of all ages as they integrate computational thinking. https://www.iste.org/standards/computational-thinking
Computational thinking with Scratch	A definition of computational thinking derived from the Scratch community that involves three key dimensions: computational concepts, computational practices, and computational perspectives. https://scratched.gse.harvard.edu/ct/defining.html
Computational thinking and CS Unplugged	Computational thinking as seen in the CS Unplugged curriculum. https://csunplugged.org/en/computational-thinking/

References

AlAmer, R. A., Al-Doweesh, W. A., Al-Khalifa, H.S.., & Al-Razgan, M.S. (2015, October). Programming unplugged: bridging CS unplugged activities gap for learning key programming concepts. In *2015 Fifth International Conference on e-Learning (econf)* (pp.97-103). IEEE

Bridging the Encouragement Gap in Computing. (2019). Retrieved from www.ncwit.org/PracticingEncouragement

Century, J., Lach, M., King, H., Rand, S., Heppner, C., Franke, B., & Westrick, J. (2013). Building an operating system for computer science. *Retrieved March 23*, 2015.

Computer Science is for Everyone: A toolkit for middle and high schools to increase diversity in computer science education. (2018). Retrieved from https://www.ncwit.org/resources/computer-science-everyone-toolkit-middle-and-high-schools-increase-diversity-computer

Computer science professional development guide. (2018). Retrieved from https://www.ncwit.org/resources/computer-science-professional-development-guide

CS Unplugged. (2016). Retrieved from http://csunplugged.org/

Cuny, J., Snyder, L., & Wing, J. M. (2010). Computational thinking: A definition. unpublished manuscript.

ECEP Alliance – Landscape Reports. Retrieved from https://ecepalliance.org/resources/landscape-reports

Guide to inclusive computer science education. (2019). Retrieved from https://www.ncwit.org/resources/guide-inclusive-computer-science-education-how-educators-can-encourage-and-engage-all

K12 Computer Science Framework. (2016). Retrieved from https://k12cs.org/

Lee, I. (2016). Reclaiming the roots of CT. *CSTA Voice: The Voice of K-12 Computer Science and Its Educators, 12*(1), 3-4. Retrieved from http://www.csteachers.org/resource/resmgr/Voice/csta-voice_03_2016.pdf.

Massachusetts K-12 Computer Science Curriculum Guide. (2017). Retrieved from http://masscan.edc.org/documents/publications/K-12_CS_Curriculum_Guide.pdf

National Center for Women & Information Technology. (2016). Retrieved from https://www.ncwit.org

SCRIPT Program. Retrieved from https://www.csforall.org/projects_and_programs/script/

State of the States Landscape Report: State-level policies supporting equitable K-12 computer science education. (2017). Retrieved from https://www.ecs.org/state-of-the-states-landscape-report-state-level-policies-supporting-equitable-k-12-computer-science-education/

Support K-12 Computer Science Education in Arizona. Code.org. Retrieved from https://code.org/advocacy/state-facts/AZ.pdf

Third Party Educator Resources (n.d.). Retrieved from https://code.org/educate/curriculum/3rd-party

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2006). *A model curriculum for K-12 computer science: Report of the ACM K-12 task force curriculum committee* (2nd ed.). New York, NY: Association for Computing Machinery.

Wing, J. M. (2006, March). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

You can actively recruit a diverse range of girls into high school computing classes: A workbook for high school teachers. (2019). Retrieved from https://www.ncwit.org/resources/you-can-actively-recruit-diverse-range-girls-high-school-computing-classes-workbook-high

Appendix A

LEA Computer Science Planning Toolkit

A strategic plan provides coherence to the overall effort of implementing K-12 computer science, offers the opportunity to involve a number of stakeholders, and creates a vision that can be publicly communicated. This toolkit helps local education agencies (LEAs) establish plans to address questions they will face as they ensure opportunities for all K-12 students to engage in high quality computer science education, such as:

How can teachers receive professional development?
What will they teach?
How will this effort be funded and communicated?

LEAs developing plans to address each of these questions will encounter key policy and implementation issues related to standards, certification, course pathways, graduation requirements, higher education entrance requirements, and professional learning, starting with designating ownership of creating and implementing the LEA's plan.

Developing an LEA Plan

The goal of this toolkit is to help an LEA create a strategic plan for both internal purposes and external communications. By defining implementation goals (e.g., number of schools or students reached), a timeline of events, and costs associated with the plan, your LEA will have a foundational document that will guide strategic decisions.

About This Toolkit

This toolkit was adapted from the State CS Implementation Toolkit that was created with participation from Code.org and the Expanding Computing Education Pathways (ECEP) Alliance, an NSF-funded Broadening Participation in Computing Alliance, reviewed and commented on portions of this toolkit.

How to Use This Toolkit

Tips

- You may develop your plan within this toolkit document and delete section and subsection prompts when done. Alternatively, you may use this blank plan.
- Your LEA team does not have to use every section or subsection of this toolkit; use them as they relate to your LEA's goals. *Use the table of contents below to skip to the desired section or subsection.*
- If your LEA has an existing template for strategic plans, copy and paste sections from this toolkit as desired.

- This document contains links to outside resources as well as links to aid you in transitioning back and forth between subsections and examples.

It is recommended that the LEA works with a variety of stakeholders in using this toolkit, including counselors, school leaders, curriculum departments, professional development departments, school boards, parent organizations, CTE departments, industry, and K-12 educators. A typical time commitment for developing the content in an initial plan may consist of 3 to 4 meetings of 3 hours each. Stakeholder groups will benefit from individuals who can facilitate meetings and/or draft the final written product, which should be reviewed and approved by the group of stakeholders.

The toolkit is divided into sections that, in some cases, are further divided into subsections. Each subsection includes resources, considerations and recommendations, and a table for listing goals and strategies, including their start/end date, responsible party/partners, and evidence of success. Descriptions and examples for completing a table are provided below:

This document could be used to develop and publish an initial plan for the LEA, but it is also intended to be used as a working document to monitor progress toward implementation of the plan. The column for tracking progress (either planning or acting) assists in this by encouraging stakeholders to revisit the plan often. These two stages are below.

- Planning: The team is beginning to define this strategy (e.g., researching needs) and the necessary action steps.
- Acting: Action on the strategy is currently underway.

Table of Contents *(Each section and subsection below are links)*

1. Current Landscape and Strategic Goals

What is going on in our LEA, what are our goals, and how close are we to meeting them?

The development of an LEA plan benefits from collecting data and administering a survey to understand the current landscape of computer science education in the LEA. The data can be used to inform the overall goals and metrics for the state plan.

Landscape Report

In this section, you will plan the development of a landscape report. The objective is to gather relevant data on the current policy and implementation landscape for K-12 computer science and use them to define the goals and metrics for the LEA plan. This section includes examples of data that could be collected to determine a baseline for computer science implementation.

Considerations and Recommendations

- Where will the data for the landscape report come from?
  - Combine survey results, data from professional development, and data for the most comprehensive view of computer science education in the LEA. Data may come from a variety of sources, including:
    - teachers,
    - schools,
    - districts,
    - and national organizations such as Code.org, College Board, Computer Science Teachers Association (CSTA), National Math and Science Initiative (NMSI), and Project Lead The Way (PLTW).
  - A survey may be developed to collect data from districts and schools. See the survey at code.org/yourschool for an example of questions that can be used to determine computer science activity at individual schools. These Google/Gallup state-specific surveys also include example questions. Use statewide course codes to determine the prevalence of computer science courses, enrollment, and teacher certification areas. However, be sure to validate that

they represent computer science courses, rather than general computer or digital literacy. See Chapter 1 of the K–12 Computer Science Framework for a definition of computer science.

Suggestions for Data to Include in the Landscape Study *(this list is not inclusive of all relevant data)*

- How many elementary, middle, and high schools offer computer science in your LEA?
    - Which elementary schools offer CS as an integrated part of the school day? How are teachers teaching CS? Is it integrated as part of the core curriculum or is it part of a STEM or technology elective course? Are there after-school clubs that offer a CS-focus such as robotics?
    - Which middle school offer CS? How is CS offered at the middle school? Is it integrated into other courses or is it a stand-alone course? If it is a stand-alone course, is it offered as an elective and what are the demographics of the course if it is an elective? Are there after-school clubs or during the school day programs that offer a CS-focus such as Girls who Code?
    - How many high schools offer AP CS A or AP CS P and what is the enrollment in these courses? Although AP CS A may comprise only a portion of the high school computer science courses being offered, data from the College Board can be used to extrapolate the level of current high school computer science implementation in your state.
        - AP CS exam participation per state over time (by gender, minority status)
        - Barb Ericson’s annual AP CS A data review
        - College Board AP data

- Teachers
    - How many elementary, middle, and high school teachers are teaching computer science in each district? What courses or topics are being taught?
    - How many teachers have the CS endorsement?

- Students
    - What are the demographics of students that have access to computer science or are enrolled in computer science courses?
    - Are the students enrolled in and completing computer science courses representative of the student population in the LEA?
    - Where does computer science currently fit into your graduation requirements?
- Partners
    - What organizations, projects, or providers of computer science curriculum and/or professional development are already acting in the LEA? Which ones could be scaled up to cover part or all of the LEA?
- Infrastructure

- How is computer science defined in your state and LEA? A sample of district and school leaders can be polled to describe what they consider to be a computer science course.
- What are the current computer science course codes at the state level? How are they implemented? What is the content required of each course? (e.g., are any digital literacy courses being taught under a computer science course code or vice versa?)

<table>
<tr><td colspan="6">Landscape Report (See <u>example</u>)</td></tr>
<tr><td colspan="6">Goals

1.</td></tr>
<tr><td rowspan="2">Strategies</td><td rowspan="2">Start/End</td><td rowspan="2">Responsible Party/Partners</td><td colspan="2">Progress</td><td rowspan="2">Specific Evidence of Success or Completion</td></tr>
<tr><td>Planning</td><td>Acting</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Strategic Goals

In this section, you will determine the strategic goals for the state plan by defining the overarching vision, goals, and timeline. This subsection includes a high-level description of the goals that an LEA should consider. When setting goals, LEAs should rely on the landscape report and articulate specific goals related to both implementation and policy. An initial step in defining strategic goals is to identify stakeholders and the facilitator for the development of the LEA plan for achieving those goals.

Considerations and Recommendations

- What types of goals should be included?
    - Include implementation goals. An example of an implementation-related goal for high school is to establish a computer science course pathway within a defined number of years. An example of an implementation goal related to diversity is a 30% increase in the percentage of female students in computer science classes.
    - Note: The table below does not have a column for "Specific Evidence of Success or Completion," because goals identified here should be specific, clear, actionable and based on related subsections of the plan. Further, these goals should be copied and pasted into the tables of the corresponding subsections in this toolkit.

Strategic Goals (See example**)**					
Overarching Vision Statement for Computer Science Education *Describe your LEA's vision for computer science including the LEA's beliefs and rationale.*					
Goal(s) *The goals should correspond with the vision you identified above. Arrange the goals chronologically. These goals should be copied and pasted into the corresponding sections of this toolkit.*	**Related Subsection of Plan**	**Start/End**	**Responsible Party/Partners**	**Progress**	
				Planning	**Acting**

2. Diversity

How will we ensure that all students have access to and are engaged in K-12 computer science?

The result of equitable access should be computer science classrooms that are diverse in terms of race, gender, disability, socioeconomic status, and English language proficiency. LEAs play a significant role in building access to and equity within systems for implementation of high quality computer science education. In this section you will list some strategies for broadening participation in computing. Access for all students to computer science education and diversity in available courses should be integral to *all* aspects of your state strategy for K-12 computer science education. **Goals and strategies related to diversity should be considered in every part of the LEA plan.** As you work on different subsections in the plan, come back to add strategies in this section. You will find that those strategies relate to other parts of the plan.

Guidance Resources

- Chapter 2 of the K–12 Computer Science Framework addresses equity issues in computer science, provides recommendations, and gives examples of programs working to increase equity. Also see abridged web version.
- Computer Science is for Everyone slides here and accompanying toolkit here contains recruitment road map and the strong case for why we need to actively recruit to reach all learners.

Considerations and Recommendations

- How will you keep diversity and equity at the *forefront* of your computer science education reform efforts?
    - Include diversity goals across multiple efforts in the LEA plan. For example, the students taking computer science, the teachers in pre-service programs, and even the people creating the state plan should represent diverse backgrounds. Computer science education funding in AZ is currently prioritized for schools and districts that do not yet offer a CS course grades 9-12.
- How will you ensure implementation is equitable?
    - Demographic data such as socioeconomic status, gender, and race/ethnicity should be regularly collected and reported for computer science courses.
- What professional development for reaching diverse audiences and ensuring equitable implementation will you put in place for key stakeholders and decision makers?
    - The National Center for Women in Technology (NCWIT) provides an inventory of resources that address increasing the engagement of women in computer science in K-12 education, higher education, and careers. NCWIT's Counselors for Computing resources provide ideas for engaging counselors and administrators.
    - The Alliance for Access to Computing Careers (AccessComputing) provides resources for increasing the participation of people with disabilities.
- Who are the local, state, and national experts from which you can draw research-based strategies and resources?
    - Find partner organizations that are uniquely positioned to inform diversity and equity work in your LEA and your LEA's plan.
- In which schools will the implementation efforts begin?
    - Consider schools that are not yet offering any CS courses or CS integration into other courses. Also consider how you will build the K-12 pipeline. This often means elementary and middle school feeder schools aligning their CS programs with the high school CS courses.

Diversity (See example)

Goals

1.

Strategies *It is expected that some of these strategies may overlap with ones in other sections. This table provides a central location for collecting all of the diversity-related strategies in the LEA plan.*	**Start/End**	**Responsible Party/Partners**	**Progress**		**Specific Evidence of Success or Completion**
			Planning	**Acting**	

3. Teacher Pipeline

Where will the computer science teachers come from?

LEAs can address the teacher pipeline by providing ongoing professional development for existing teachers.

Professional Development
The purpose of providing professional development for teachers of other subjects is to leverage the existing pool of teachers and provide a short-term approach for increasing the number of computer science opportunities in schools. In this section, you will develop a plan for rolling out professional development to help existing teachers, possibly teaching other subject areas, to teach computer science.

Guidance Resources
- Chapter 8 of the K–12 Computer Science Framework (p.173-176) addresses professional development. Also see abridged web version.
- Building an Operating System for Computer Science (University of Chicago), describes a comprehensive understanding of our nation's current high school computer science teaching population, the support they have, and contexts in which they teach.

- [Computer Science Professional Development Guide] is a PD toolkit that provides guidance to LEAs as they align their PD to their vision for CS
- [Computer Science Teachers Association: CS Standards for Teachers (2020)] are designed to provide clear guidance around effective and equitable CS instruction in support of rigorous CS education for all K-12 students.

Considerations and Recommendations

- Who will provide the professional development?
  - Identify existing computer science professional development providers in the state, including other state-level professional development providers that can provide professional development for computer science (whether or not they currently offer computer science). There are national providers of professional development that can be found at Code.org's 3rd party resources and the CSforAll Consortium.
- Are you focusing on one level of education (e.g., elementary, or high school), or all levels simultaneously?
  - Some LEAs have begun with high school teachers in order to serve students before they leave the K-12 system. Other states have focused on middle school as that is the greater area of need and an opportunity for getting students interested in further study in high school. Others focus on finding opportunities to integrate CS into the core curriculum or as part of an existing STEM or technology elective. Eventually, professional development should be available for elementary, middle, and high school teachers.
- How will professional development support the AZ K-12 Computer Science Standards and/or recommended curriculum options?
  - Professional development should promote and align to the AZ K-12 Computer Science Standards.
- How is high-quality professional development characterized?
  - High-quality professional development should align with the recommendations in the K–12 Computer Science Framework, AZ K-12 Computer Science Standards, and/or your LEAs's strategic goals to reach all students.
- Who is offering the professional development — state, district, non-profit provider, university, for-profit providers, or a combination?
  - A variety of providers, with common accountability measures, can serve a broader audience.
- How does your professional development strategy reach a diverse population?
  - Approach schools to ensure that a diverse student body is being reached.

How will you ensure that the professional development emphasizes equitable teaching practices?

  - Rely on providers that have shown a history of research-based and field-tested practices to improve equitable practices.
- What is the timeline for professional development?
  - Consider all providers and determine how many teachers can be trained each year. Consider how many years it will take to reach your LEA's goal and the milestones along the way.
- How will professional development opportunities be communicated to districts, schools, and teachers?

    - AZ lists professional development opportunities, as well as instructional resources, on a dedicated computer science page on their website. Computer Science Teachers Association - Arizona (CSTA AZ) also provides learning opportunities for teachers in the state.
- How much will professional development cost?
    - Current sources of state and federal funding may be accessed to pay for professional development. The Arizona Computer Science Professional Development grant is currently open and accepting applications on a rolling basis.

Professional Development						
Goals 1.						
Strategies	**Start/End**	**Responsible Party/Partners**	**Progress**		**Specific Evidence of Success or Completion**
			Planning	**Acting**	

4. Curriculum and Courses

What courses will teachers be teaching? What curriculum best aligns with the computer science goals in your district/organization?

Standards are an essential component of a larger education plan and provide a foundation with which to align the other components, such as curriculum, instruction, and policies such as graduation requirements. The development and selection of curriculum and courses plays a significant role in the access and equity movement in computer science education.

Curriculum

In this section, you will plan the process of planning the development and selection of curriculum and courses.

Considerations and Recommendations

- What are effective practices for creating and/or evaluating high-quality computer science curriculum?
    - Chapter 8 of the K–12 Computer Science Framework provides recommendations around curriculum, assessment, and course pathways. Also see abridged web version. Recommendations include how to create and evaluate curriculum that appeals to a broad student population.
    - Consider what role the state's computer science standards may play in creating and/or evaluating curriculum and how your timeline allows one to inform the other.
- What are options for high-quality computer science curriculum?
    - Code.org's 3rd party resources page provides detailed information about multiple providers.
    - LeadCS inventory of curriculum resources
    - List of endorsed curricula for AP CS Principles
- What can the state do to ease adoption at the LEA level?
    - If applicable, state-level course codes can simplify the adoption of courses at the LEA level and offer the state a way to measure course uptake.
    - Consider how you will communicate and recommend curriculum options to your LEA and schools.
- How will teachers be trained on curriculum?
    - Many curriculum providers also offer professional development.
- What are the costs for high-quality curriculum? Are there current or potential sources of funding that may be accessed to pay for curriculum?
    - There are a variety of free and adaptable resources available. A lack of funding shouldn't be an obstacle to the adoption of curriculum.

Curriculum (see example)

Goals

1.

Strategies	Start/End	Responsible Party/Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	

5. Outreach

How will people know about the plan to offer computer science to all students and provide input/feedback?

Effective implementation of the statewide computer science initiative requires proactive communication using a variety of methods within the LEA. Students, educators, administrators, community members, and industry leaders must have open channels of communication to ask questions and provide feedback. In this section, you will determine your LEA's outreach plan.

Considerations and Recommendations

- What pre-existing meetings and events can be used to share the LEA plan and get buy-in?
    - Consider a variety of local events. For example, Vail School District holds annual computer science celebration evenings in conjunction with CS Education Week with hands-on demonstrations and informative content for the entire community.
    - How are you incorporating the goals and message for expanding computer science for all students and broadening participation in computing?

NCWIT has hundreds of resources available for all of the audiences with whom you might engage. Paradise Valley Unified School District has a PVWIT initiative with a goal of encouraging and increasing representation of girls in tech through innovative internships, outreach and partnerships.
    - Consider your main messaging or talking points around the initiative. How does the messaging differ for different audiences? Here is a presentation with a variety of slides that can be used to advocate for K-12 computer science.
- What events can the public (and media) attend for input or to showcase progress?
    - Somes states have used the Hour of Code and Computer Science Education Week to highlight the state's initiatives and results.
- What national resources and their local affiliates are you connected to?
    - Organizations such as the CS for All Consortium, CSTA (CSTA AZ), and CS for All Teachers are a few of the organizations that provide communication forums and networking opportunities.

Outreach					
Goals 1.					
Strategies	**Start/End**	**Responsible Party/Partners**	**Progress**		**Specific Evidence of Success or Completion**
			Planning	**Acting**	

6. Funding

How will we pay for this effort?

Funding is required to achieve many of the goals in a plan. Approaches and uses of funding can differ over the short term and long term of implementing K-12 computer science. In the short term, dedicated funding for computer science should be allocated and the funding should emphasize the professional development of existing teachers for the purpose of building an initial teacher force. Funding priority should be given to districts in which a demonstrable effort will be made to engage underrepresented groups. In the long term, states should secure sustained funding streams to support a system of high-quality computer science education. In this section, you will plan how to secure funding to support the execution of the state plan.

Considerations and Recommendations

- What needs to be funded?
  - Consider funding multiple areas, such as professional development, computers, internet,and a dedicated computer science education leadership position.
- How can federal funding be leveraged?
  - Computer Science and ESSA builds off ESSA's inclusion of computer science as part of a "well-rounded education" and provides suggestions for incorporating computer science into a state's ESSA plan, including the use of funds as designated under ESSA. These suggestions include:
    - Using Title I Schoolwide programs funds to support the recruitment and retention of computer science teachers, activities designed to prepare students for success in high-quality advanced coursework, and CTE programs to prepare students for postsecondary education and the workforce.
    - Using Title II funds to support professional development for computer science educators.
    - Using Title IV, Part A funding to support activities to improve STEM, including computer science.
  - The Arizona Department of Education has AZ Computer Science professional development grants that target computer science. Applications are accepted on a rolling basis.

Funding (see example)

Goals

1.

Strategies	Start/End	Responsible Party/Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	

Example Plan: LEA X

Landscape and Goals *(the strategies below are examples for illustrative purposes)*

Landscape Report (back to the Landscape Report section)

Goals

1. Measure the current state of computer science education in the LEA across a variety of areas to inform the LEA's goals.

Strategies	Start/End	Responsible Party/Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	
Build collaborative team to define data to collect, develop survey, collect data and write landscape report.	Summer 2019	LEA leadership, teacher/parent/curriculum/PD representatives		X	Team of 5 people identified as key leaders on landscape report development

Strategic Goals (back to the Strategic Goals section)

Vision

By 2022, High School X in the LEA will offer a computer science pathway and have a qualified computer science teacher. The LEA envisions a future where all students:

- critically engage in public discussion on computer science topics;
- develop as learners, users, and creators of computer science knowledge and artifacts;
- better understand the role of computing in the world around them; and
- learn, perform, and express themselves in other subjects and interests.

(K–12 Computer Science Framework, 2016)

Goals	Related Subsection of Strategic Plan	Start/End	Responsible Party/ Partners	Progress		
				Planning	Acting	Done
Establish a computer science course pathway	Professional Development	Summer 2018/ Summer 2021	LEAs, school leaders, teachers, professional development providers		X	
Establish at least one teacher teaching either computer science or integrated computer science courses within science and/or mathematics in every middle school.	Professional Development	Summer 2017/ Summer 2020	LEA, school leaders, teachers, professional development providers	X		
Establish at least one teacher teaching either computer science or integrated computer science courses within media arts or computer lab time in every elementary school.	Professional Development	Summer 2017/ Summer 2021	LEA, school leaders, teachers, professional development providers	X		

Diversity *(the strategies below are examples for illustrative purposes)*

Diversity (back to the Diversity section)
Goals 1. Increase the percentage of female students in secondary computer science courses from 20% to 40%.

Strategies	Start/End	Responsible Party/Potential Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	
Identify the difference between statewide student demographics and current representation in computer science classes. Create district chart.	Fall 2019-Spring 2020	LEA stakeholder group	X		Strategic plan to increase access to computer science in K-12, ensuring all students equal access created.
Create a district guide focused on recruiting underrepresented groups and train administrators and counselors at monthly meetings.	Fall 2017/Winter 2018	LEA stakeholder group	X		Guide created, shared, and administrators and counselors trained.

Teacher Pipeline *(the strategies below are examples for illustrative purposes)*

Professional Development (back to the Professional Development section)

Goals

1. Establish at least one teacher teaching high-quality computer science courses in every high school.
2. Establish at least one teacher teaching either computer science or integrated computer science courses within science and/or mathematics in every middle school.
3. Establish at least one teacher teaching either computer science or integrated computer science courses within media arts or computer lab time in every elementary school.

Strategies	Start/End	Responsible Party/Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	

Secure professional development funding through AZ CS Professional Development grant	Spring 2020	LEA leadership	X		Funding is accessible to districts for professional development and stipends.

Curriculum and Courses *(the strategies below are examples for illustrative purposes)*

Curriculum (back to the Curriculum section)

Goals

1. Recommend courses and curriculum aligned to the state standards.

Strategies	Start/End	Responsible Party/Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	
Publish assortment of open-source, LEA-selected curriculum resources on the LEA's computer science web page.	Spring 2019/ Summer 20120	Curriculum and Instruction/ LEAs		X	The LEA computer science webpage includes curriculum resources and includes integration ideas for K-8.

Outreach *(the strategies below are examples for illustrative purposes)*

Outreach (back to the Outreach section)

Goals

1. Increase awareness of the current computer science work in the LEA.

Strategies	Start/End	Responsible Party/Potential Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	
Get feedback on draft plan from stakeholders (teachers, district leaders, parents, etc.)	Spring 2019	LEA stakeholder team	X		Arrange and hold X# of local or regional meetings
Create computer science education portal/website/social media presence to keep stakeholders informed	Fall 2020/ Spring 2021	LEA stakeholder team and marketing team	X		LEA website page created to house all computer science effort materials.

Funding *(the strategies below are examples for illustrative purposes)*

Funding (back to the Funding section)

Goal

Secure funding dedicated to computer science professional development for existing teachers.

Strategies	Start/End	Responsible Party/Potential Partners	Progress		Specific Evidence of Success or Completion
			Planning	Acting	
Work with LEA leadership to apply to AZ Computer Science Professional Development grant	Summer 2019/ Spring 2020	LEA stakeholders and leaders	X		Completed and submitted grant application to Arizona Department of Education; finalized approval by Arizona State Board of Education