Reimagining CS Pathways
High School and Beyond

Interim Report #2

Defining Pathways Beyond
a Foundational High School CS Course

Authors & Leadership

The project is primarily planned, facilitated, and coordinated by the project team:

CSTA
Bryan Twarek (PI)
Jake Koressel (Project Manager)

ACE
INSTITUTE for ADVANCING COMPUTING EDUCATION
Dr. Monica McGill (Co-PI)
Dr. Julie Smith (Researcher)

Steering Committee

The steering committee has ultimate responsibility for all research, deliverables, and the overall project success. The steering committee members are:

CSTA
Bryan Twarek
VP of Education & Research

acm
Dr. Tom Cortina
Education Board & Advisory Committee

CODE
Dr. Jamila Cocchiola
6-12 Curriculum Product Manager

CollegeBoard
Crystal Furman
Director, AP Curriculum, Instruction, & Assessment

CSforALL
Dr. Leigh Ann DeLyser
Executive Director

ECEP Expanding Computing Education Pathways
Sarah Dunton
Director

Advisory Board

Advisors ensure that the project is making adequate progress toward its goals. Advisors are:

- Dr. Adrienne Decker, Associate Professor, University at Buffalo
- Deborah Seehorn, Past Chair, CSTA Board of Directors & Standards Revision Task Force
- Delmar Wilson, Teacher, Miami Springs Senior High School

Funding Support

This project is supported by the National Science Foundation (NSF) under Grant No. 2311746. 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 NSF.

Suggested Citation

CSTA, IACE, ACM, Code.org, College Board, CSforALL, & ECEP Alliance. (2024). *Reimagining CS Pathways Interim Report #2: Defining pathways beyond a foundational high school CS course.* New York, NY: Computer Science Teachers Association. Retrieved from https://ReimaginingCS.org.

Convening Participants

- **Julie Alano**, High School CS Teacher and Department Chair, Hamilton Southeastern High School, *Fisher, IN*
- **Cathy Ammirati**, STEM Outreach Manager, Micron, *Boise, ID*
- **Quiana Bannerman**, Instructional Supervisor - CTE, Prince George's County Public Schools, *White Plains, MD*
- **Kris Beck**, Director of CS, Chicago Public Schools, Chicago, IL
- **Darlene Bowman**, Founder and CS Teacher, AusomeTech Industries, *Staten Island, NY*
- **Justin Cannady**, Northern Lights Collaborative for Computing Education, *Birmingham, AL*
- **Cindi Chang**, Director of Teaching and Learning, Nevada DOE, *Las Vegas, NV*
- **Dr. Terry Coatta**, Practitioners Board Chair, Association of Computing Machinery, *Langley, BC*
- **Jackie Corricelli**, PreK-12 CS Curriculum Specialist and Teacher, West Hartford Public Schools, *West Hartford, CT*
- **Becca Dovi**, Chief CS Advocate, CodeVA, *Richmond, VA*
- **Charlotte Dungan**, Director of Content Development, TeachAI, *Durham, NC*
- **Rachel Fenichel**, Engineering Manager, Google, *San Francisco, CA*
- **Sara Frey**, State Lead for K-12 CS Education, Pennsylvania DOE, *Harrisburg, PA*
- **Crystal Furman,** Director of Java in Education, Oracle, *Snellville, GA*
- **Laura Gray**, AI and CS Instructional Specialist, Gwinnett County Public Schools, *Athens, GA*
- **Dr. Christopher Harris**, School Library System Director, Genesee Valley BOCES, *LeRoy, NY*
- **Dr. Sean Jackson**, K-12 CS Lead, Kentucky DOE, *Frankfort, KY*
- **David Lockett,** K-12 Outreach and Federal Programs Manager, Meharry School of Applied Computational Sciences, *Nashville, TN*
- **Dr. Michelle Magallanez**, Head of Strategic Partnerships and Innovation, AVID, *San Diego, CA*
- **Dr. Janice Mak**, Clinical Assistant Professor, Arizona State University, *Tempe, AZ*
- **Dr. Laura Malavé**, College of CS and IT Faculty, St. Petersburg College, *St. Petersburg, FL*
- **Dr. Amanda Mason-Singh**, Principal Data Scientist, The MITRE Corporation, *McLean, VA*
- **Sofia Mohammed,** Executive Director, Raspberry Pi Foundation - North America, *Madison, MS*
- **Angela Oechslie**, Director of Project Login, Educate Maine, *Bangor, ME*
- **Yolanda Payne**, Research Associate, Georgia Tech, *Northeast Georgia, GA*
- **Carla Strickland**, Digital Curriculum Development Manager, UChicago, *Chicago, IL*
- **Cat Tabor**, CS Teacher, Canutillo Independent School District, *El Paso, TX*
- **Brett Tanaka,** CS Education Specialist, Hawaii DOE, *Honolulu, HI*
- **Amy Traylor**, CS Program Development Specialist, Albuquerque Public Schools, *Albuquerque, NM*
- **Dr. John Underwood,** STEM Specialist, Louisiana Department of Education, *Baton Rouge, LA*
- **Thomas Wang,** High School CS Teacher, LGSUHSD, *Bay Area, CA*
- **Perla Weaver**, Associate Professor, Johnson County Community College, *Overland Park, KS*
- **Lavita Williams**, CS Specialist, Georgia DOE, *Atlanta, GA*

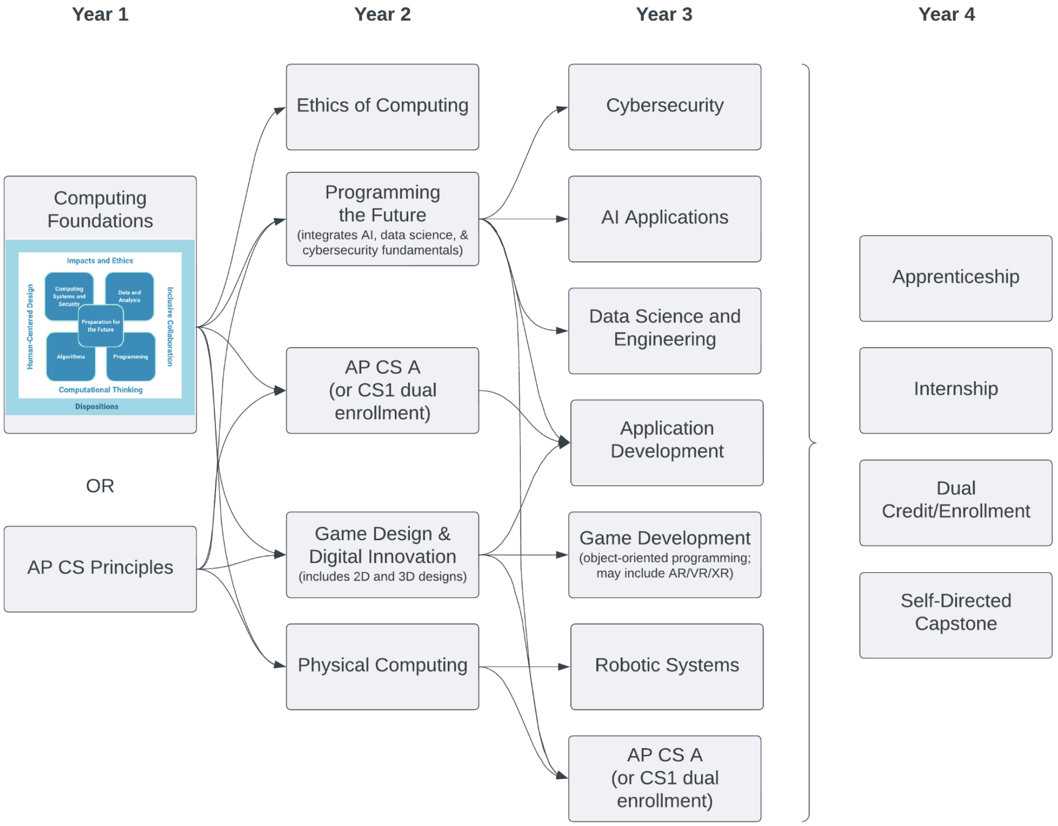
Table of Contents

View the final *Reimagining CS Pathways* report at **ReimaginingCS.org**

Executive Summary

Anticipated changes in computing such as innovations in artificial intelligence, a shifting policy landscape (e.g., an increase in state-adopted CS graduation requirements), and perhaps other circumstances that cannot even be predicted will continue to necessitate changes in the structure and approach to K-12 CS education. To best prepare students for these changes, this project – *Reimagining CS Pathways: High School and Beyond* – has the goals of developing a community definition of essential content for high school students and exploring what CS pathways schools might offer stemming from that foundational content.

This project has several aspects and phases, and this interim report provides a summary of the project's second convening, which focused on articulating what the pathways following the foundational high school CS content might look like. Convening participants considered the following speciality areas when developing content progressions beyond the foundation: computer science, cybersecurity, artificial intelligence, physical computing, data science, game design and development, and X + CS. Draft content progressions can be found in Section 4 of this report. The diagram below shows possibilities for implementing the content progressions in the form of pathways (course sequences), acknowledging that only portions of the diagram may be implementable depending on the size and characteristics of a given school. This and other implementation pathways are described in detail in Section 5.



Participants had a variety of suggestions – as well as concerns – surrounding high school CS pathways, particularly around equity and the ability of under-resourced schools to offer robust CS programming. Specifically, participants grappled with the level of granularity necessary to capture the intent of progressions - recognizing the need to be clear and precise while also allowing for flexibility in pedagogical approach and implementation strategy. Participants also recognized the need for content progressions and their ultimate pathway implementation to be customizable to suit the needs of varying local (e.g., district) contexts.

This report describes the process participants followed and synthesis of ideas to develop suggested content progressions, implementation pathways, and other related considerations.

1. Introduction

We aspire to reimagine computer science (CS) education in high school and beyond. Critical to this effort is defining a new set of foundational knowledge, skills, and dispositions that *every* student will learn in the coming years. Additionally, we expect a growing – and more diverse – group of high school students to choose to continue learning CS beyond this foundation.

In this second interim report, we present a variety of content progressions for continued CS learning beyond the foundation, including specialized areas such as artificial intelligence (AI), cybersecurity, game design, and data science. Additionally, we present models of learning pathways and considerations for implementation.

2. Project Background and Overview

The purpose of the Reimagining CS Pathways: High School and Beyond project is to develop community definitions of 1) what CS content is essential for all high school graduates and 2) what content and pathways for continued CS learning should exist for high school students beyond a foundational course. We aim to not only develop recommendations to inform the future of the CSTA K-12 Standards and AP CS courses, but also to clarify the alignment of and develop model pathways for CS learning from high school through introductory computing experiences at the post-secondary level.

Across the 2023-24 school year, we will hold three convenings with representatives from across the K-16 CS education landscape (including teachers, administrators, 2- and 4-year college instructors, curriculum developers, and industry). Project leadership will synthesize data from these convenings and other sources, in order to produce a written report with final recommendations, which we expect to release in summer 2024.

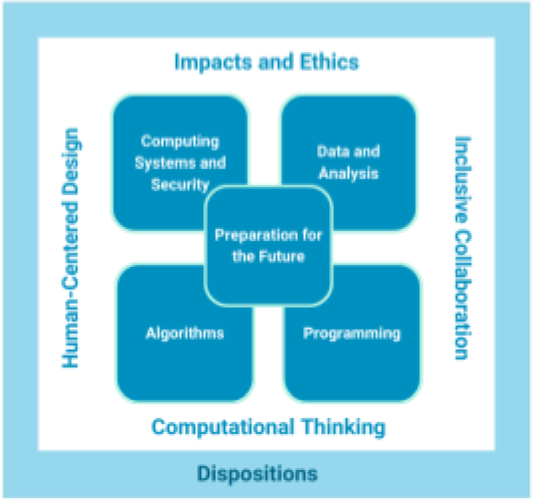


Figure 1. Foundational high school CS content established from the first convening.

The first phase of the project focused on what CS content is essential for all high school graduates and is summarized in Figure 1. More detail about this convening and its conclusions can be found in the Interim Report #1.

Feedback to inform recommendations and next steps are being gathered from a diverse cross section of the CS education community and includes both synchronous and asynchronous opportunities for interactive feedback.

In support of the aims of this project, the following project values have been identified and will be leveraged for continuous reflection on progress and refinement of deliverables.

Equity-centered. Promotes broad and equitable access, participation, and experiences in computer science education among all high school students.

Community-generated. Meets the needs of the community, including K-12 educators, post-secondary institutions, students, parents, and industry.

Future-oriented. Anticipates future needs of current high school learners, and prepares them for a future that is increasingly reliant on computing.

Grounded in research. Reflects the evolving body of knowledge of how students learn computer science.

Flexible in implementation. Considers multiple pathways for meeting individual needs of learners, including regional, cultural, ability, social, and economic factors.

2.1 Convening Participants

The steering committee and project team selected 40 convening participants from 26 states via a process that prioritized deep experience and diversity across a variety of factors. Factors included geographic locale (i.e., U.S. region as well as urban/suburban/rural), expertise, role, demographic, and institution type. Figure 2 shows that 73% of participants identified as women; 53% identified as Asian, Black, Latinx, and/or Native; and 14% have a disability or chronic condition. A breakdown of convening participants by primary professional role and relevant experience can be found below. More detailed demographics are presented in Appendix B.

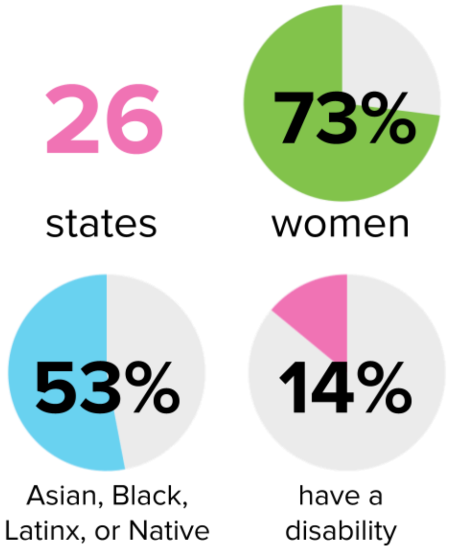


Figure 2. Summary of convening participant demographics.

Thirty-three participants (see list on p. Ⅱ) joined Convening #2. This report summarizes data from these 33 participants, plus five members of the steering committee who were present.

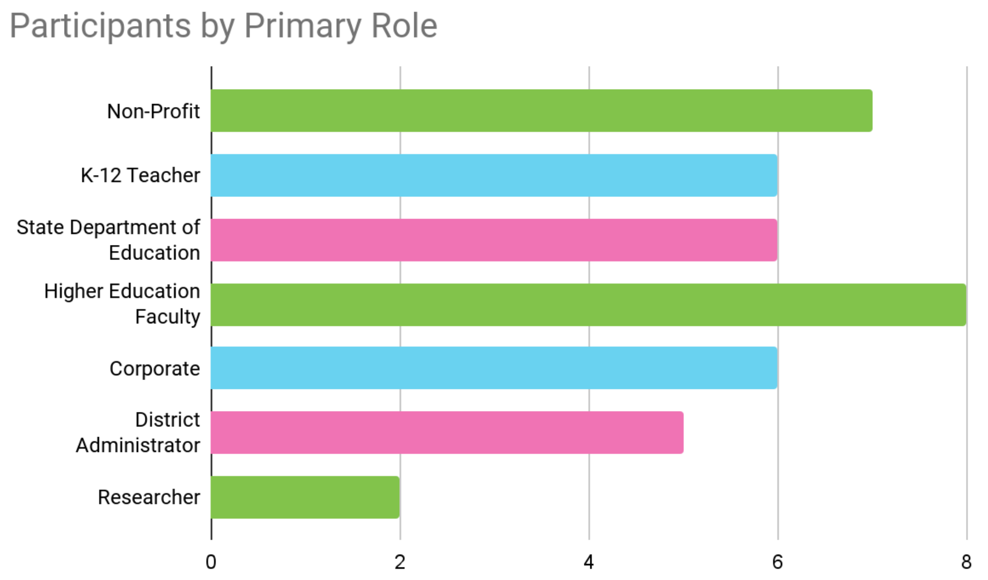


Figure 3. Convening participants by primary professional role.

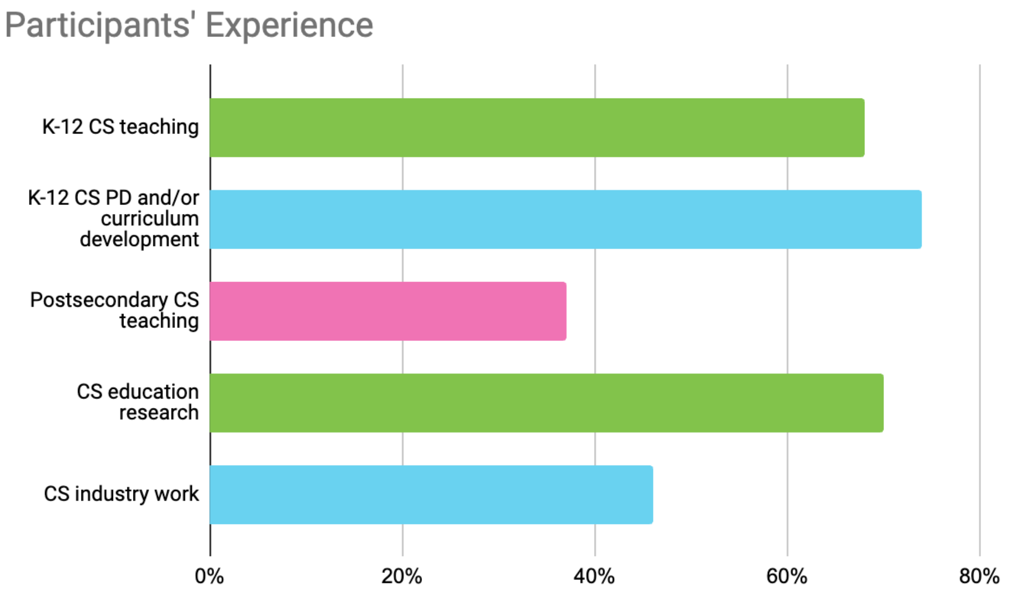


Figure 4. Convening participants' experience related to CS education and industry.

2.2 Student Perspectives

Before the second convening, the project team conducted interviews with college students studying (or very recent graduates who had studied) computer science **to ensure that student voice was included** when making decisions about changes to CS content in high schools. Interview subjects were selected to represent a diverse cross-section of students from those who indicated interest in response to a call for participants shared via attendees at the first convening. Students described some challenges to studying CS in high school, primarily a lack of teachers available to teach CS and limited course offerings. It is perhaps not surprising that these students – all of whom chose to study CS in college – frequently expressed that they wished that their high school had offered more CS classes, particularly more programming instruction. While the students had a variety of high school experiences – from a traditional suburban school to a very small early college high school program – none had a wide selection of CS offerings, with most schools offering just one or two CS courses. Several mentioned positive experiences with extensive student-directed projects.

The second convening included a panel of students from the Atlanta area who are in or who recently graduated from high school and learned CS. Since our convening was held in Atlanta, students were recruited by one of the convening participants who was an AI and CS instructional specialist for the local school district. The student panelists shared their perspectives and stories describing how they became interested in CS and started on their path. Their beginnings differ, with some being exposed to CS classes in middle school, playing programing games in their own free time, attending a tech summer camp, going to a high school that provided CS classes, or discovering job opportunities that sparked their interest, but **one commonality was that teachers were influential in guidance and mentoring** during their CS journey.

The panelists were very thoughtful about the content that they felt was valuable or less so in their high school coursework. For example, one panelist was pleased to have learned **linear algebra** and suggested this would be a great addition to high school content. He noted, "You have to know linear algebra as your fundamental building block because everything else in computer science is based on linear algebra." More than one student suggested that emphasis on **traditional algorithms**, such as sorting algorithms, may be receiving **too much attention** in high school computer science. One student described that, despite their emphasis, they have not "run into sorting algorithms ever again." These panelists, along with some of our attendees, are implying that there may need to be a **shift in topics away from how to create traditional algorithms and toward understanding their relevance and utility in creative problem solving** (J. Corricelli, personal communication, April 2024).

Another student stressed that sometimes “it may be difficult to see the real-life application of the CS concepts.” They added it will be helpful if students were able to have experiences in the “real-world applications of CS.” Others also highlighted that students should be aware of different CS pathways: “CS is not just about coding.” Another student said a course should be added that touches on the different pathways. One panelist noted that her teacher required them to attend networking events and do job shadowing, and it was beneficial for her. This type of work-based learning requirement aligns neatly with preparing students for the future, a topic area that is part of the foundational CS content.

2.3 Focus Groups

In fall of 2023, we held three focus groups, one each for those whose primary role is teaching high school CS, higher education CS, or working in industry. We chose participants via a process that prioritized diversity across a variety of factors, including geography, expertise, experience, demographics, and institution type. A portion of the discussions focused on what CS pathways should exist for high school students. Participants articulated different schema for pathways, including:

- A description of a graduate as one with experience in AI, game design, robotics, digital design, and intellectual property
- A sequence of CS courses: exploring CS, intro to CS, AP CS P, and AP CS A
- A sequence of CS courses (intro to CS, AP CS P, AP CS A) followed by the student's choice of a course in AI, cybersecurity, or data science

These pathways were generally in alignment with current pathways, with the addition of some specialized options such as AI and data science that are less common in schools today.

2.4 The Second Convening

Figure 5. Participants develop content progressions at the second convening.

The second convening was held in Atlanta, Georgia on January 25-26, 2024 (see the complete agenda in Appendix A). The focus of this convening was the development of possible pathways for computing that extend beyond the foundational CS content defined in the first convening. During the first day, participants heard from a guest speaker, Dr. Matt Welsh, the Chief Architect and Co-founder of Fixie.ai, who discussed possible trajectories for the impact of generative AI on the future of computing and computing education. Then participants formed small groups to map what CS content would be needed to bridge the foundational content and particular post-secondary student outcomes: CS major, cybersecurity major, AI major, physical computing/robotics major, data science major, game design or development major, CS + Humanities major, alternative pathways to CS (i.e., not involving higher education), and non-CS career paths. A backward design template was used, as shown in Figure 6. Example posters for three specialty areas are shown in Figure 7. At the end of the first day of the convening, participants engaged in a gallery walk where they viewed a variety of pathways from non-CS disciplines (see Figure 8 for samples).

Upon Graduation		
ESTABLISHED GOALS *Students engaged in this content will be prepared to . . .*	***Meaning - Big Ideas to Achieve the Goal***	
	UNDERSTANDINGS ***Beyond the CS Foundation content****, students will understand that…* Refers to the big ideas students will have when they complete the goal. (e.g., students will understand how the Internet works)	
	Acquisition - Specific Content and Skills to Achieve the Goal	
Refers to outcome. (e.g., major in CS)	*Students will know…* Refers to the key knowledge students will acquire from the goal. (e.g., students will know the relationships between the seven layers of the OSI model)	*Students will be skilled at…* Refers to the key skills students will acquire from the goal.

Figure 6. Backwards-design template for mapping CS content.

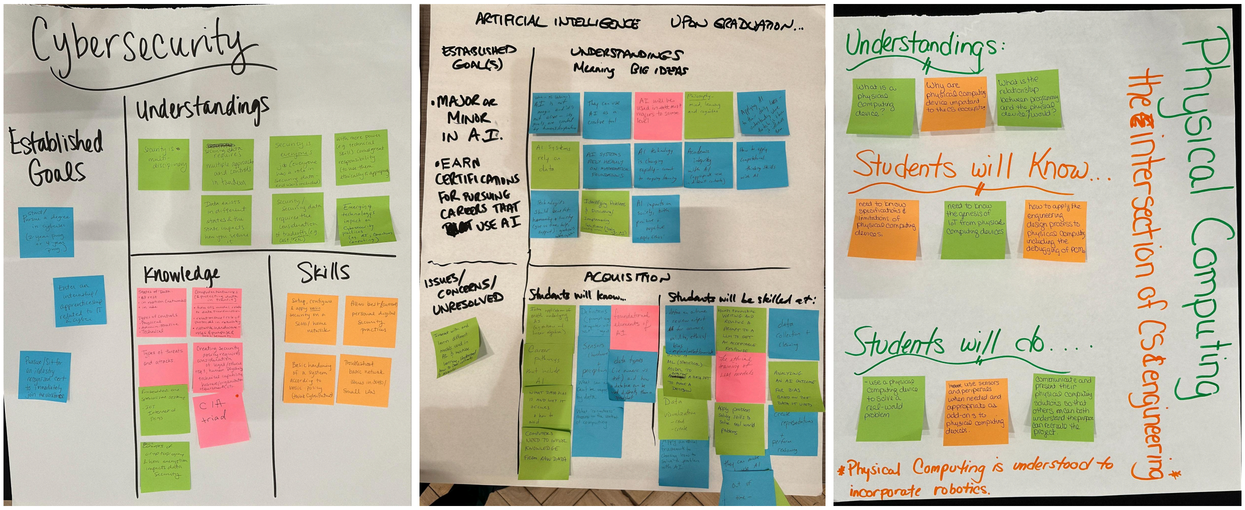


Figure 7. Example posters from working groups to define CS content beyond the foundation.

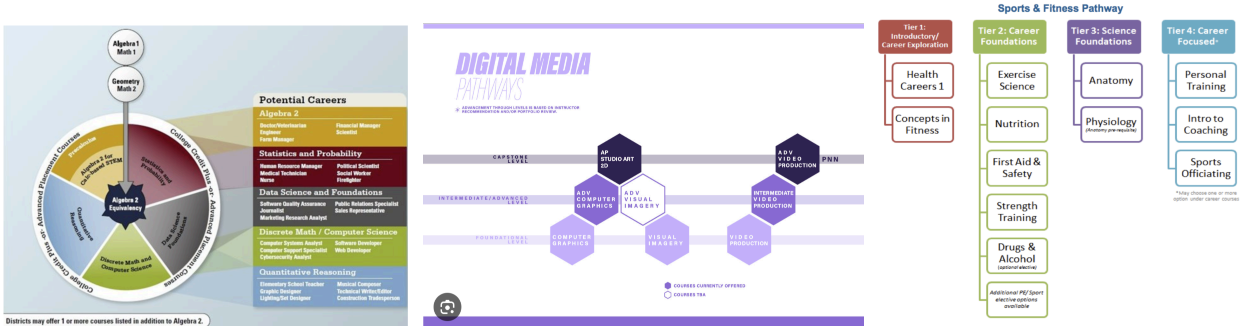


Figure 8. Sample pathways from the gallery walk (sources for the first, second, and third image).

Participants were asked to consider which features of these pathways were or were not useful to include in the pathways that they would soon develop (see Section 4 below). As a result of this gallery walk, participants identified which features they did or did not find helpful in the sample pathways. Features of strong pathways included a simple design: avoiding too many lines, crossing lines, or curving lines, and focusing on the use of simple shapes, with clear sequence or progressions. Participants also felt that the best pathways connected the 'why' to the pathway content by identifying community needs, student interests, careers, and outcomes that align to or extend from each pathway. They noted language should be brief, clear, descriptive, and understandable. While less commonly mentioned, participants also flagged that pathways should be clearly organized, flexible, and use color strategically. Appendix C shows the sample pathways in greater detail.

On the second day of the convening, participants built pathways for each of the postsecondary trajectories listed above (e.g., AI, cybersecurity). These pathways were refined (see Section 4). Then, participants considered how the pathways could be selected and implemented at a hypothetical high school (see Section 5).

Figure 9. Convening participants refining CS content progressions.

3. Challenges to Convening 2 Goals

Several challenges became apparent in the effort to achieve this convening's goals of articulating CS pathways beyond the foundation. As noted by participants, the current lack of aligned CS curriculum and advanced teacher professional development (PD) to support new CS pathways are key challenges. Furthermore, participants may not feel adequately equipped to contribute fully, reflecting concerns about their expertise in navigating the various forms of potential pathways.

Implementation. During this convening, participants were asked to design pathways that implemented the content described for each area. Participants noted real-world constraints related to teacher capacity, school schedules, and other resources made it extremely difficult to include all or even most of the content, particularly for smaller schools. Further, implementation challenges generally held equity implications, with schools serving the most marginalized students facing the largest challenges.

Curriculum and PD. A key challenge identified by participants is the absence of well-defined CS outcomes aligned with detailed pathways (using either a single curriculum or a combination of curricula) that are relevant to students and meet the needs of our changing world, as well as PD for educators that also meets those needs. Participants, including higher education CS faculty and high school CS instructors, expressed reservations about their ability to provide meaningful insights. For instance, higher education CS faculty members may lack understanding of considerations specific to high school learning, while high school CS instructors may struggle to discern the knowledge and skills required for success in a major like AI.

Semantics. How to name various content groupings presented a challenge. For example, we called one grouping “Game Design and Development.” There was support for renaming this group in a way that reflected that the skills necessary for game design and development are applicable more broadly to other 2D/3D simulations, such as Digital Simulation Innovation. However, other participants felt that students, especially younger students, are more likely to understand and be attracted to “Game Design” than to “Digital Simulations,” while others felt that girls and women may be less attracted to “Game Design” than to other titles.

The Shifting Landscape. The dynamic nature of educational requirements adds another layer of complexity to the implementation process. Changes at the K-8 and the higher ed levels will necessarily impact what is, should be, and can be taught in high schools. The rapid changes in K-12, higher education, and industry are likely to continue, and will continue impacting each other - raising questions about the adequacy of existing preparation for future educational requirements.

Policy Landscape and Local Implications. Anticipating an uneven policy landscape, the report explores the possibility of varying adoption rates of graduation requirements across states. While some states have adopted CS graduation requirements and more likely will, others may not, introducing potential disparities in the educational experience. Consequently, understanding the local implications of such disparities becomes crucial for successful implementation.

K-8 Preparation. The report highlights the uncertainty surrounding the K-8 preparation that students can expect prior to entering high school. This lack of clarity poses a challenge for educators and administrators aiming to design effective pathways that seamlessly bridge the gap between K-12 and postsecondary education.

Modularity and Granularity. A modular design (e.g., a six-week unit that introduces students to cybersecurity principles) has the advantage of being easier to incorporate into various contexts than a year-long class, but modular design can make the process of aligning instructional materials very challenging. Similarly, describing content at the right level of granularity (that is, neither too specifically nor too generally) was a key and persistent challenge.

The Appropriate Level of Reimagining. Participants confirmed the need to innovate within these conversations, but also pointed out that we should be mindful of what schools might consider unattainable. The balance between aspirational and realistic is important for ensuring that outputs from the convening are useful and implementable.

4. Content Progressions

The purpose of content progressions is to delineate CS content that students can learn across multiple years and/or experiences in high school. The participants were urged (1) to explain succinctly but with sufficient detail to convey critical knowledge and skills and (2) to package content into model courses or other experiences (e.g., after school clubs, integrated courses). The audience for the content progressions was assumed to include CS teachers, curriculum designers, and administrators who are developing or selecting pathways. (Note that this presents something of a contrast to the community-facing pathways diagrams shown in Figure 8.)

As an important consideration, we separated content from implementation and how the content might appear in formats that students may engage in. Below is a series of tables that detail a progression of content (i.e., from foundational to fundamental to specialty) within a given specialty area (e.g., cybersecurity, physical computing). A given cell within each table could be offered as a standalone course, as an out of school time (extracurricular) learning experience, at a boot camp, through a summer camp, or via another learning experience. How content is implemented is at the sole discretion of the districts/schools and their particular needs, goals, constraints, and resources, including availability of teachers to teach this content.

Note that the project team engaged in some editing and realignment of material for consistency. For example, the computer science content progression initially included cybersecurity and data science content, which was moved to those respective sections. The content progressions that follow were each reformatted as tables with three columns intended to progress sequentially (left to right; see Figure 10), to represent what students would learn across multiple high school experiences, for students choosing to pursue a particular pathway. Modified versions of participants' original designs for content progressions are shown in Appendix D.

Foundational high school CS content, as drafted in the first interim report	**Fundamental** knowledge and skills that is core to the concentration area and extends beyond the foundation	**Specialty** knowledge and skills that deepens students' understanding of the concentration area

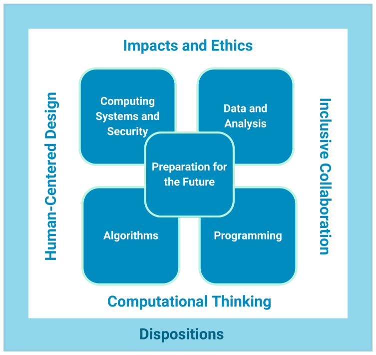
Figure 10. Consistent structure of the content progressions.

4.1 Computer Science Content Progression

Table 1 shows the model content progression developed for students who are interested in continuing to learn computer science broadly, such as those intending to major in CS and potentially become a computer scientist or software engineer. **We note again that this content does not include further cross-cutting concepts, such as additional impacts on society.** The absence of this does not reflect its importance, as cross-cutting concepts should also be embedded in fundamental and specialty learning.

Table 1

Computer Science Content Progression

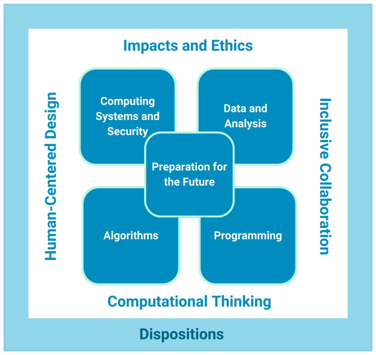
Foundational CS Content	Fundamentals	Specialty
	• Decomposition • Problem solving • Conditions, iterations, selection, functions • Arrays and data structures • Unit testing • Debugging • Usage of IDEs	• Programming skill development • Software development processes (e.g., Agile/SCRUM) • Application development (e.g., mobile apps, virtual reality apps) • Team project skills • Collaborative source control
Possible careers: Computer Scientist, Software Engineer, Data Scientist, Artificial Intelligence, Cybersecurity Specialist, Network Specialist, Roboticist		

4.2 Cybersecurity Content Progression

While at some level security is a shared responsibility, Table 2 shows the content progression for acquiring more knowledge specifically related to cybersecurity. This content might lead to a major in cybersecurity or to earning industry certifications, followed by a career as a network technician, security analyst, or network systems administrator.

Table 2

Cybersecurity Content Progression

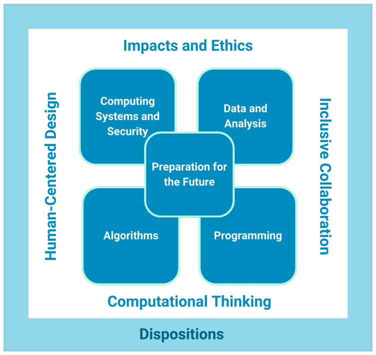
<table>
<tr><th>Foundational CS Content</th><th>Fundamentals</th><th>Specialty</th></tr>
<tr><td>
</td><td>● Basic computing systems
● Hardware/software roles and components
● Basic understanding of file systems
● Impact of cybersecurity on society/like and critical infrastructure
● Personal security habits
● ‘Consumer’ of networks
● SOHO/home networks
● Types of attacks, threats, vulnerabilities, and basic remediation strategies
● Wifi versus Internet
● Public networks
● network addressing (IP addressing, MAC addressing)</td><td>● CIA triad, states of data, and types of controls
● High-level understanding of policies and why they matter
● Network hardware and their roles (servers, switches, routers, endpoints, firewalls)
● Basics of digital communication (OSI model, protocols, ports, etc.)
● Basic application security
● Basic hosting security
● scripting
● Incident response
● Ethical hacking and penetration testing basics
● Risk management
● Business continuity
● More on org policies (impact of regulations, law, etc.)
● Emerging technologies’ impact on cybersecurity</td></tr>
<tr><td colspan="3">**Possible careers:** Network Technician, Computer Support Specialist, Security Analyst, Network Systems Admin, Penetration Tester, SOC analyst, Security Operations and Testing, Risk Management, Security Management, Security Architect, Security Research and Development</td></tr>
</table>

4.3 Artificial Intelligence Content Progression

Table 3 shows the content progression for artificial intelligence. The AI content may require more prior mathematical knowledge than other pathways. This progression might lead to an AI major and to careers as a machine learning engineer, computer vision engineer, or AI ethics and policy analyst, among others.

Table 3

Artificial Intelligence Content Progression

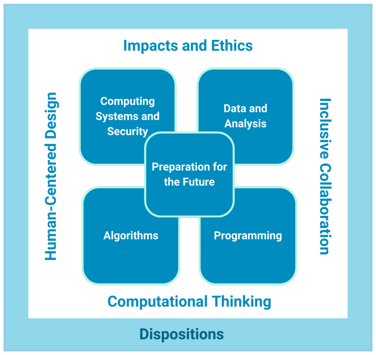
Foundational CS Content	Fundamentals	Specialty
	• What is AI: history, levels of AI, future careers • Intro to AI programming and intro to prompt engineering • Natural interaction, semantics, chatbots • Representation and reasoning, KNN, vectors • AI programming (project), using AI tools to solve problems • Ethical frameworks, philosophy, psychology, bias • Sensors, perception, classification • Using datasets, regression, probabilistic thinking • CNN, decision trees, bias • Return to ethical design and empathy interviews	• Fundamentals of electronics, mechanisms, circuits, gears, sensors • Computer vision, sensor applications, models, perceptions • Robot hardware manipulation (or software simulators) • Using data: collection, cleaning, data types, validity, bias • Machine learning (ML) models: optimization, accuracy, decision making, ethical considerations • Linear algebra, matrices, vectors, probability, statistics • Programming applications with math • Biases in data collection, analysis, and reporting • Data visualization
Possible careers: Machine Learning Engineer, Data Scientist, AI Research Scientist, Computer Vision Engineer, Natural Language Processing Engineer, Robotics Engineer, AI Ethics and Policy Analyst, Autonomous Vehicle Engineer, AI Cybersecurity Engineer		

4.4 Physical Computing Content Progression

Table 4 shows the content progression for physical computing, which includes robotics. This content may lead to a physical computing or a robotics major and ultimately to careers as a robotics engineer, industrial automation specialist, control systems engineer, or human-robot interaction specialist, among others.

Table 4

Physical Computing Content Progression

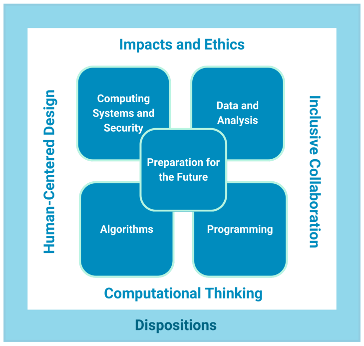
Foundational CS Content	Fundamentals	Specialty
	• Specifications and limitations of physical communication devices • Genesis of IoT from physical computing devices • How to apply the engineering design process to physical computing, including debugging • Use a physical computing device to solve a real-world problem • Use sensors and peripherals appropriately as add-ons to physical computing devices • Communicate and present physical computing solutions so that others can understand the purpose and recreate the project	• Creating solutions to problems using physical computing • Programming for physical devices • Software development processes (e.g., Agile, SCRUM) • Networking for physical devices • Application development (e.g., mobile apps, virtual reality apps) • Team project work • Collaborative source control
Possible careers: Robotics or Embedded Systems Engineer, Robotics Research Scientist, Industrial Automation Specialist, Control Systems Engineer, Automation Engineer, Mechatronics Engineer, Robotics Software Developer, Drone Engineer, Human-Robot Interaction Specialist, Biomechanics Engineer		

4.5 Data Science Content Progression

Table 5 shows the data science content progression. This content may lead to a data science major and a career as, for example, a data scientist, data modeler, statistician, or data ethicist.

Table 5

Data Science Content Progression

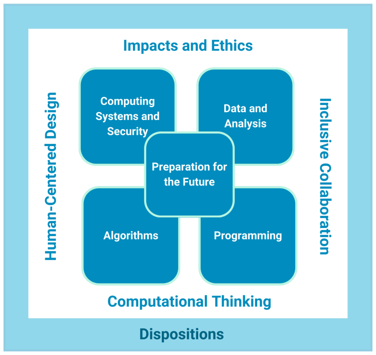
Foundational CS Content	Fundamentals	Specialty
	• Extract meaning from tabular data using a function • Descriptive statistics • Simple visualizations • Data forms and bias (ethics) • Manipulate data • Data validity (clean and accurate) • Data privacy, security, bias, missing data, ethics • Make predictions (using e.g., frequentist and Baysian statistics) • Legal and ethical implications • Data science tools • Structured problem-solving (case studies; case analysis) • Query formation (prompt engineering; SQL; elastic search) • Statistics (normal distribution, descriptive statistics, regression analysis) • Data fairness and bias (mitigating bias)	• Distributed cloud based systems • Data storage locations and data manipulation (e.g., APIs, building .json, data scraping, finding and processing data outside of a traditional data location) • Databases (structured and unstructured data) (e.g., relational, graph, vector databases, other NoSQL) • Data modeling (e.g., how to map tables together) • Machine learning basics • Data validity, credibility, and reliability (data consciousness), • Data visualization • Data privacy and security • Database architecture • Interface development for data analysis (e.g., BI tools, such as PowerBI, Tableau)
Possible careers: Data Scientist, Data Security, Data Privacy, Data Ethicist, Data Modeler, Statistician		

4.6 Game Design and Development Content Progression

Table 6 shows the game design and development content progression, which shares content with other 2D and 3D digital simulations. (See also the challenge related to **Semantics** in section 3 regarding naming and whether the pathway should be broader than games.) This content may lead to a game design or a game development major and a career as a graphic designer, game tester, sound engineer, or game developer.

Table 6

Game Design and Development Content Progression

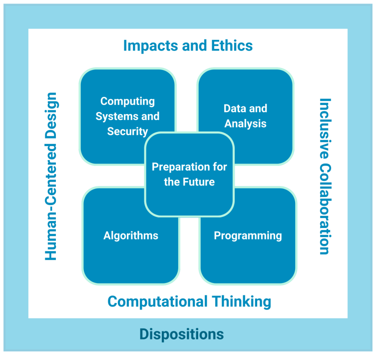
Foundational CS Content	Fundamentals	Specialty
	- Game Design - Game Art - Game Sound - Interactive Design - User Interface - Psychology of Games - Story - Ethics - Accessibility (games should be accessible to all) - Inclusivity (broad cultural, religious, gender, physical, cognitive differences) - Social impact (games have power to influence culture, cultural values and norms) - Physical modeling - Programming (e.g., interaction, navigation, world building, physics) - Debugging - Game/simulation careers	- UI/UX - Character and environment design - Art history and direction - 2D and 3D animation - Motion graphics - Simulations - Sound/music history - Encoding analog info (character state, mood) - AR/VR/XR - Object-oriented programming - Physics and states - Controller design - Integrating art and animation - Integrating sound/music - Encoding analog info - Source Control - Team Collaboration - Game development engines
Possible careers: Game Designer, Game Developer, Graphic Designer, Concept Artist, Producer, Writer, Level Designer, Game Tester, Sound Engineer, Simulation Engineer		

4.7 X + CS Content Progression

Table 7 shows the content progression for X + CS. X can represent any subject area, including humanities. X + CS requires integration between the two or more subject areas. This content may lead to a major in CS, the 'X' subject, or X + CS, followed by a wide variety of careers, including biomedical engineer, educational technologist, digital media specialist, or medical simulation specialist (See also Appendix E: X + CS Implementation Details).

Table 7

X + CS Content Progression

Foundational CS Content	Fundamentals	Specialty
	• Identify and articulate common themes, practices, and terminology between CS and X • Explore historical examples of CS and X considering universal human endeavors as a bridge and identifying gaps and challenges • Evaluate data visualizations in X, manipulate X data via computational models • Reframe problems in X using CS: decompose problem, translate into program, determine whether the program solves the problem • Use programming skills to explore multiple perspectives in X	• Examine how the evolution of X impacts CS and vice versa • Manipulate data models to allow for utilization of source data from X • Evaluate and compare algorithms that address problems in X • Contribute to the evolution of X in CS by creating an artifact • Manipulate, create models in order to use source data from X (aligned to student's project choice) • Develop a plan that uses algorithms in programming to address problems in X (student is selecting)
Possible careers: Medical Simulation Specialist, Biomedical Engineer, Business Data Analyst, Computing ethicist, Neuroscientist, Education Technologist, Digital Media Specialist, Digital Linguist, Human Language Technologist		

5. Implementation Pathways

Whereas the content progressions (Section 4) delineate CS learning to occur across multiple years of learning, implementation pathways describe ways that this content could be packaged together into courses, or other equivalent experiences such as integrated instruction. These pathways are designed to provide additional, elective learning opportunities for students that extend the foundational high school CS content.

5.1 Recommendations

At the second convening, participants were presented with details for six hypothetical high schools, based on real data from the National Center for Education Statistics (NCES) and designed to represent various realistic scenarios (see Appendix F for details). They were asked to determine ways various CS content progressions could be implemented in each context, given the school's context and available resources.

A common challenge that surfaced with participants' responses across all school contexts is an inability to offer the full range of CS pathways. However, combining several ways to offer content can extend beyond the classroom; for example, small schools may offer extracurricular activities like robotics club or game design clubs. They may also establish partnerships within the community to offer summer enrichment camps and programs, apprenticeships programs, and other resources to support students in their learning. Participants generated many other ideas for implementing the pathways in the face of these and other challenges, and we present a summary of recommendations.

5.1.1 Equity Considerations

When designing implementation pathways, educators and leaders should consider many implications for promoting educational equity, including:

1. Flexibility in implementation better supports students who, for example, move into a school district in the middle of high school to participate in the pathway. Flexibility is also useful for students who choose to change pathways, have differing prior experience, extend learning outside of school, or complete self-guided learning.

2. Pathways should accommodate a variety of postsecondary plans, including not just higher education but also industry certifications, direct entry into the workforce, and military service, etc.
3. Only the most highly resourced schools will be able to implement a wide variety of CS pathways and options; in all other instances, students will have fewer opportunities to exercise choice in what CS content they choose to study.
4. Pathways should accommodate students with a range of prior experience – including no prior experience – in CS, as well as a range of prior math knowledge and English language fluency.
5. There are barriers specific to rural and urban contexts, and innovative solutions may need to be implemented to overcome them (e.g., teacher sharing programs, transportation for afterschool programs).
6. Course offerings are often connected to teacher certification/credentialing requirements, which may limit a school's ability to offer specific courses. For example, high school CS courses are often classified as either Career Technical Education (CTE) or traditional academic courses. In some states, dual coding is permitted and in others, it is not. Offering CTE courses may qualify schools for Perkins V funding, to support software, hardware/equipment, curricular materials, teacher professional development, and hiring of new teachers and administrators for up to three years.
7. Limited funding and/or infrastructure (e.g., start-up costs for robotics equipment, unreliable wifi) are likely to affect some schools more than others.
8. Opportunities for postsecondary credit (e.g., dual enrollment) and placement in advanced coursework (e.g., after passing AP exams) may be limited by students' ability to pay for college credits, exams, and certifications.
9. Different communities will have differing priorities for higher education versus certifications.
10. Ensuring that administrators and counselors are well-informed to encourage participation can replace 'gate-keeping' with 'gate-opening.'
11. One potential disadvantage of outside-of-school programs is that they may be less accessible for some students. For example, students may not have transportation options from an after school robotics club, and thus may be unable to participate, or under-resourced schools may not be able to fund the staff for a coding club.

*Note that the above list is numbered for reference purposes only and does not represent a particular rank or hierarchy.

5.1.2 Teaching the Foundation and Beyond

Schools could teach the foundational high school CS content and pathways for continued learning in several ways. The following ways are listed in no particular order:

1. **Offer a discrete course(s).**
   - A discrete foundational course could focus on just CS content and satisfy the school's CS graduation requirement, if one exists.
   - If there is computing offered in middle school and/or if some foundational content is supplemented in other high school courses and/or experiences, then this might be a semester-long course.
   - An extended version of this course would allow for exposure to different pathways so that students can make informed decisions on subsequent learning experiences.
   - A series of discrete courses could create a pathway.

2. **Integrate foundational content into other subject areas.**
   - Schools may distribute foundational CS content to teach in other classes, based on strategic alignment. There are opportunities for interesting collaborations, such as with data analysis integrated into social studies or ethnic studies and new media integrated into art courses.
   - This may be challenging: there is the potential need for pairing integration with a discrete course or intensive integration planning.
   - Supplement with informal learning opportunities, like out-of-school time.
   - Creating integrated pathways may require an analysis of what content beyond the foundation does not require prerequisite knowledge to identify potential points for integration, or intensive integration planning may need to occur.

3. **Offer as part of one or more of the following programs:**
   - Dual enrollment, based on course offerings at a local college or university
   - Advanced Placement (AP)
   - Career and Technical Education (CTE)
   - In partnership with an institution of higher education for early college credit (i.e., dual enrollment or dual credit).
   - In tandem with relevant afterschool programming.

4. **Provide flexible options:**
   - Allow students to take the foundational high school CS course in middle school, while satisfying a CS graduation requirement, if one exists.
   - Provide access to a virtual or online course.
   - Create work-based, service-based, and/or project-based learning integration.
   - Teach specific fundamental and specialty content on a rotating basis, to maximize teaching capacity.

5.1.3 Pathway Endpoints

High school pathways may lead to postsecondary studies and eventual careers related to the concentration area. However, there are many potential endpoints for a pathway, including:

- Certifications
- Internships
- Apprenticeships
- Student-directed capstone courses
- Certificate or specialty at a 2-year institution
- Minor or major at a 4-year institution
- Bootcamp
- Direct entry career

Given these varied endpoints, CS pathways should support students in career exploration and industry awareness in the relevant areas. Pathways should also develop professional skills necessary for college, career, and civic engagement (see Appendix G).

5.2 Model Pathways

Figure 11 illustrates a model set of implementation pathways. Actual pathways can and should differ widely based on local needs and resources. However, this sample is meant to suggest what a relatively full implementation might look like for a large school positioned to deliver a comprehensive set of CS pathways. It is not expected that schools would have the capability of offering all (or even multiple) pathways represented. Note that content of the courses shown in the diagram are intended to align with content from the content progressions in Section 4.

In this model, students may choose to take either a computing foundations course or AP CS Principles. Fundamental content of different concentration areas are packaged both discreetly (e.g., Physical Computing and Game Design & Digital Innovation) and in an integrated fashion (e.g., the Programming the Future course integrates fundamentals from CS, AI, data science, and cybersecurity). Since students may take courses in different grades, the columns represent years of instruction, rather than specific grade levels; some students may only complete one year of instruction, whereas others may complete two, three, or even four years. While most pathways progress linearly (left to right), there are generally multiple options for selecting specialization courses. For example, after taking Game Design & Digital Innovation, a student may choose to take Game Development, Application Development, or AP CS A. Fourth year options include apprenticeships, internships, dual enrollment courses, and self-directed capstones.

Most schools will not be able to offer as many options for specialized CS learning, so they may select a relevant subset of these pathways or substitute other areas of concentration.

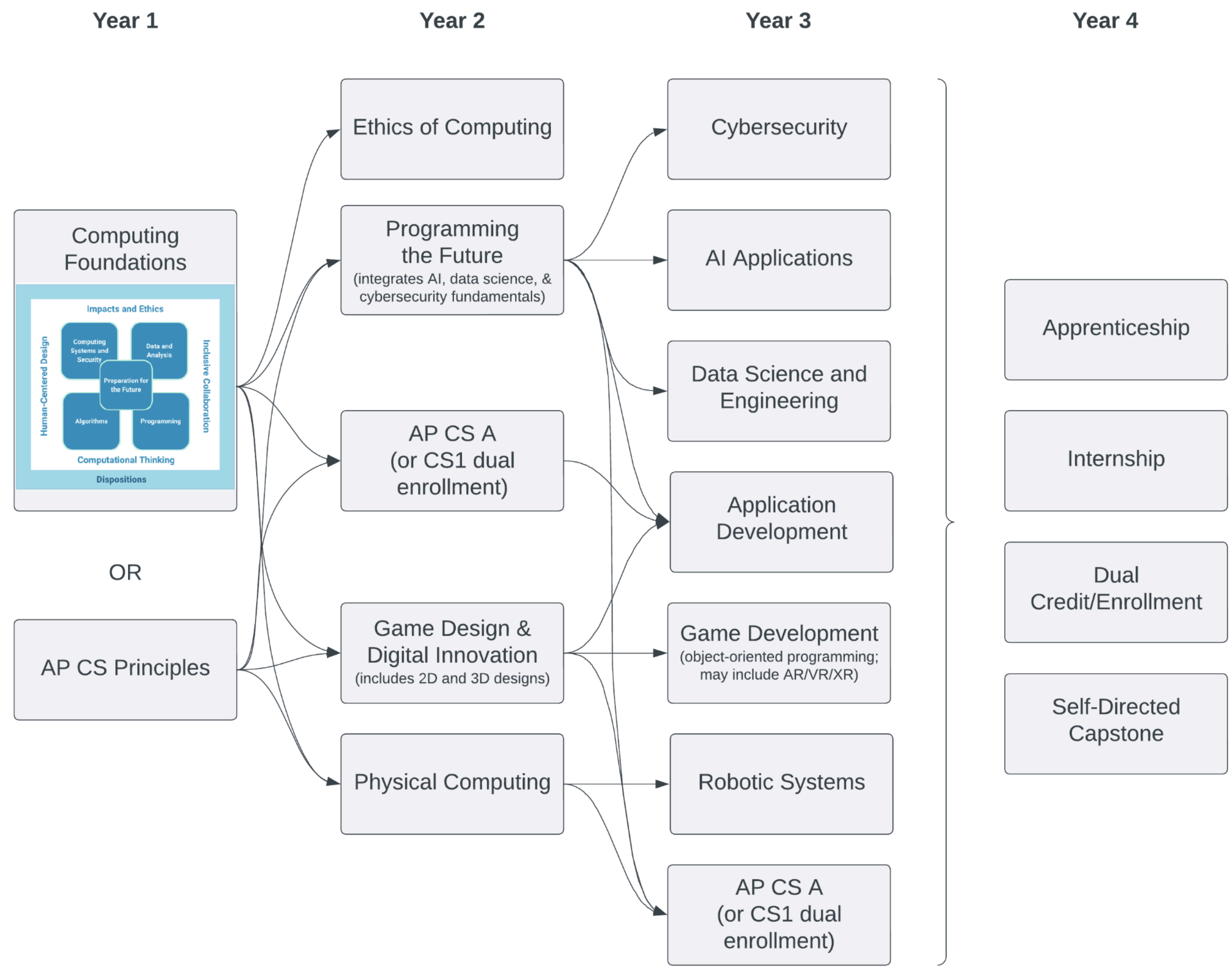


Figure 11. Model implementation pathways.

5.3 Model Integration

When all students develop a consistent foundation in computer science, teachers are able to leverage and extend their CS knowledge and skills in other courses. Whereas the previous model illustrated how pathways could be implemented in discrete courses (Figure 11), another option is to integrate advanced CS content in other subject areas. For example, Figure 12 shows how fundamental and specialized AI content (see section 4.3: AI Content Progression) could be integrated into other courses that students take after the foundational CS course.

Subject Area	**Example Integration of AI Content**
Social Studies (including Civics & Ethnic Studies)	● What is AI: history, levels of AI, future career ● Ethical frameworks, philosophy, psychology, bias ● Return to ethical design and empathy interviews ● Biases in data collection, analysis, and reporting ● Using AI tools to solve problems
Mathematics	● Representation and reasoning, KNN, vectors ● Using datasets, regression, probabilistic thinking ● Using AI tools to solve problems ● Linear algebra, matrices, vectors, probability, statistics ● Programming applications with math
Language Arts	● Natural interaction, semantics, chatbots ● Intro to prompt engineering ● Using AI tools to solve problems
Science	● Sensors, perception, and classification ● Developing simulations to model phenomena ● Using AI tools to solve problems ● Using data: collection, cleaning, data types, validity, bias
Computing	● Intro to AI programming ● CNN and decision trees ● AI programming project
Arts	● Biases in data collection, analysis, and reporting ● Data visualization ● AI programming (project)

Figure 12. Example integration of fundamental and specialized AI content into other subject areas.

5.4 Implementation Guidance

Before implementing a new CS pathway, it is important to define current CS offerings at a school or district, then identify areas for development based on relevance to the community and desired outcomes for students. Ramping up a robust CS program takes time, and it may require establishing a multi-year plan that involves assessing teacher interest, evaluating possible professional development opportunities, training teachers (including non-CS teachers), and recruiting additional teachers.

One convening working group articulated how a school may design and implement a set of relevant CS pathways over the course of five years (see Figure 13).

Year 1	Year 2	Year 3	Year 4	Year 5
• Survey teachers to learn about their interest and past experience with CS • Research sources for quality professional development • Plan training to prepare one or more teachers to teach foundational courses	• Courses are in place for one of the pathways • Provide training for non-CS teachers and CS teachers to continue to grow course offerings • Make decisions about curriculum • Clarify credentialing requirements	• Courses are in place for both pathways • Provide training for non-CS teachers and CS teachers to continue to grow course offerings • Students can complete at least one pathway • Recruit additional teachers to attain required credentials	• At least two pathways are available • Provide training for non-CS teachers and CS teachers to continue to grow course offerings • Pathways are chosen based on district needs • Authentic CS is integrated into non-CS subjects • Capstone or dual credit option	• At least three pathways are available • Provide training for non-CS teachers and CS teachers to continue to grow course offerings • Pathways are chosen based on district needs • Authentic CS is integrated into non-CS subjects • Five year strategic plan exists • CS teachers are credentialed

Figure 13. Model five-year implementation plan for a high school implementing new CS pathways.

6. Conclusion and Ongoing Considerations

As we continue to navigate this project, the following represent challenges that have not been resolved and remain considerations in future work:

- The first convening articulated CS content in terms of topic areas (e.g., programming) as well as cross-cutting concepts (e.g., inclusive collaboration). Ensuring that the cross-cutting concepts – including ethics and social impacts, the most strongly emphasized material in the first convening – are actually incorporated into topic areas in the pathways is an ongoing concern.

- The work on content progressions and implementation pathways constitutes a first pass by participants and will be further refined during the third convening, which will revisit the foundational content as well as the possible pathways to 'stress test' them in a variety of ways (e.g., whether equity concerns are adequately addressed, whether any content in the foundational CS experience needs to be modified based on the pathways that were developed).
- All decisions about pathway implementations have equity considerations, and there is sometimes a tension between the desire to offer as many opportunities as possible and the reality that some students may not be able to choose these opportunities due to a variety of potential barriers (e.g., lack of transportation or other time commitments limiting the opportunity to participate in out of school learning experiences).

Grappling with these tensions and considerations will be crucial as we work toward finalizing recommendations for future standards writers, course developers, and more in an effort to increase the number of students that participate in high quality foundational CS learning experiences and continue their learning beyond the foundation.

7. References

Moll, L., Amanti, C., Neff, D., & Gonzalez, N. (2006). Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. In *Funds of knowledge* (pp. 71-87). Routledge.

8. Appendices

Appendix A: Convening #2 Agenda

Reimagining CS Pathways: High School and Beyond

Second Convening

January 25-26, 2024
Atlanta, GA, USA

Convening Goal

- **Define what content and pathways for continued CS learning should exist for high school students and their post-high school lives.**

Thursday, January 25, 2024

12:00 – 12:30 pm	Check-in/Lunch
12:30 – 1:00 pm	Project Overview/Recap, Norms, Framing
1:00 – 1:45 pm	Guest Presentation on the Future of CS
1:45 – 3:00 pm	Define CS Content Beyond a Foundational Course
3:00 – 4:00 pm	Refine Content Beyond a Foundational Course
4:00 – 5:00 pm	Student Panel
5:00 – 5:45 pm	Pathways Gallery Walk
5:45 – 6:00 pm	Debrief and Close-out
7:00 – 9:00 pm	Working Dinner

Friday, January 26, 2024

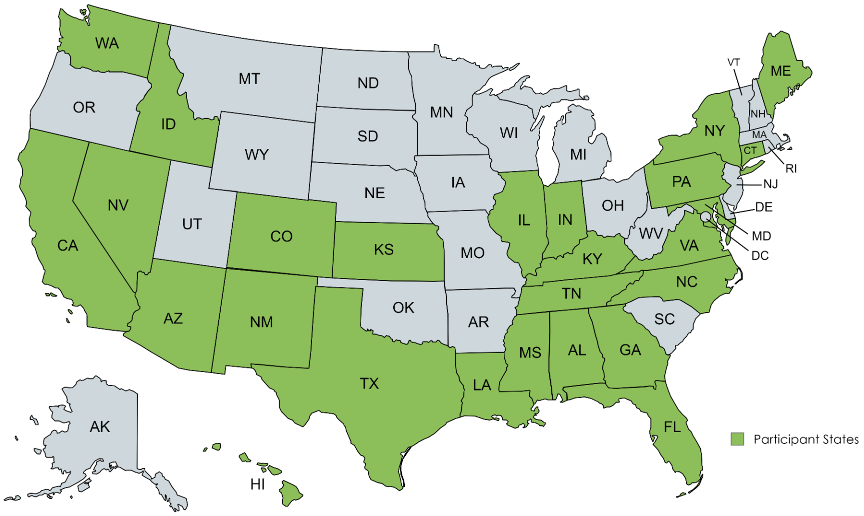
8:00 – 8:30 am	Breakfast
8:30 – 9:00 am	Reflection and Framing for Day 2
9:00 – 11:00 am	Design/Draft Model CS Pathways
11:00 – 11:45 am	Refine Model CS Pathways
11:45 am – 12:30 pm	Lunch
12:30 – 2:00 pm	Continue to Refine Model CS Pathways
2:00 – 3:45 pm	Develop Recommended Implementation Strategies for Various School Contexts
3:45 – 4:00 pm	Debrief and Close-out
4:00 pm +	Travel Home

Appendix B: Participant Demographics & Experience

The steering committee and project team selected 40 convening participants via a process that prioritized deep experience and diversity across a variety of factors, including geographic (i.e., U.S. region as well as urban/suburban/rural), expertise, role, demographic, and institution type).

States

Participants represent 26 states: AL, AZ, CA, CO, CT, FL, GA, HI, ID, IL, IN, KS, KY, LA, MD, ME, MS, NC, NM, NV, NY, PA, TN, TX, VA, and WA.



Created with mapchart.net

Race/Ethnicity

The table below shows the distribution of participants' racial and ethnic identities. Several participants identify with multiple races or ethnicities, so the numbers and percentages do not sum to 40 and 100%, respectively.

Race/Ethnicity	Number	Percent
White or Caucasian	21	53%
Black or African American	8	20%
Hispanic or Latinx	7	18%
Asian or Asian American	5	13%
Prefer not to answer	2	5%
American Indian or Alaska Native	1	3%
Native Hawaiian or Pacific Islander	0	0%
Another race or ethnicity	0	0%

Gender Identity

The majority of participants identify as women (*n* = 29, 72.5%), and the remainder identify as men (*n* = 11, 27.5%). No participants identify as non-binary or another gender.

Gender Identity	Number	Percent
Woman	29	73%
Man	11	28%
Non-Binary	0	0%
Another gender	0	0%
Prefer not to answer	0	0%

Disability Status

Approximately 14% of participants identify as having a disability or chronic condition. We did not collect data about specific types of disability or condition, though we did ask about and provide disability-related accommodations at convenings.

Identify as having a disability	Number	Percent
No	27	75%
Yes	5	14%
Prefer not to answer	4	11%

Primary Professional Role

Participants' current and primary professional roles were relatively balanced across K-12 teachers, higher education faculty, district administrators, state departments of education, corporations, and K-12 CS education non-profit organizations. While there are only two participants whose primary role is researcher, 70% of participants have experience with CS education research (as shown in the next table: Professional Experience).

Primary Professional Role	Number	Percent
Higher Education Faculty	8	20%
Non-Profit	7	18%
Corporate	6	15%
K-12 Teacher	6	15%
State Department of Education	6	15%
District Administrator	5	13%
Researcher	2	5%

Professional Experience

Participants have wide-ranging experience across K-12 CS education, postsecondary CS education, and industry, with an average of 9 experience types listed in the table below.

Experience	Number	Percent
K-12 CS professional development	28	76%
CS education research	26	70%
K-12 CS curriculum development	23	62%
9-12 CS teaching	21	57%
Teaching introductory high school CS courses	20	54%
K-12 CS standards development	17	46%
CS industry work	17	46%
Teaching AP CS Principles and/or AP CS A courses	16	43%
6-8 CS teaching	14	38%
K-12 school leadership	12	32%
K-12 district or local education agency leadership	12	32%
K-5 CS teaching	9	24%
K-12 state education agency leadership	9	24%
Postsecondary CS teaching at 4-year primarily undergraduate institution	8	22%
Postsecondary CS teaching at 4-year PhD-granting institution	8	22%
Teaching dual enrollment CS courses	5	14%
Postsecondary CS teaching at HSI	4	11%
Postsecondary CS teaching at 2-year institution	3	8%
Postsecondary CS teaching at HBCU	1	3%
K-12 guidance counselor	0	0%
Postsecondary CS teaching at Tribal College/University	0	0%

Expertise Supporting Marginalized Groups

Participants have significant expertise serving student populations that are marginalized and underrepresented in CS education, as indicated in the following table.

Expertise Supporting Marginalized Groups	Number	Percent
Girls and non-binary students	28	76%
Economically disadvantaged students (or Title I schools)	22	59%
Latinx or Hispanic students	22	59%
Black or African American students	19	51%

Students with disabilities	19	51%
Bi-/multi-lingual learners (English learners)	16	43%
Rural communities	15	41%
Native or Indigenous students	9	24%
Students who identify as LGTBQ+	8	22%
Students who are experiencing homelessness	7	19%
Migrant students	7	19%

CS Content Teaching Experience

Participants have taught the following CS content in their classrooms. The most common topics were computational thinking, algorithms, programming, and impacts of computing.

CS Content Coverage	Number	Percent
Computational thinking	26	70%
Algorithms and programming	25	68%
Impacts of computing	24	65%
Digital citizenship	23	62%
Computing systems (e.g., hardware/software)	21	57%
Data and analysis	21	57%
Networks and the Internet	21	57%
Ethics	20	54%
Accessibility	19	51%
Web development	19	51%
Physical computing	18	49%
App development	15	41%
Artificial intelligence (AI)	15	41%
Cybersecurity	15	41%
Robotics	14	38%
Data science	13	35%
Game design / development	13	35%
Internet of things	13	35%
Quantum computing	3	8%

Appendix C: Sample Pathway Implementation Models

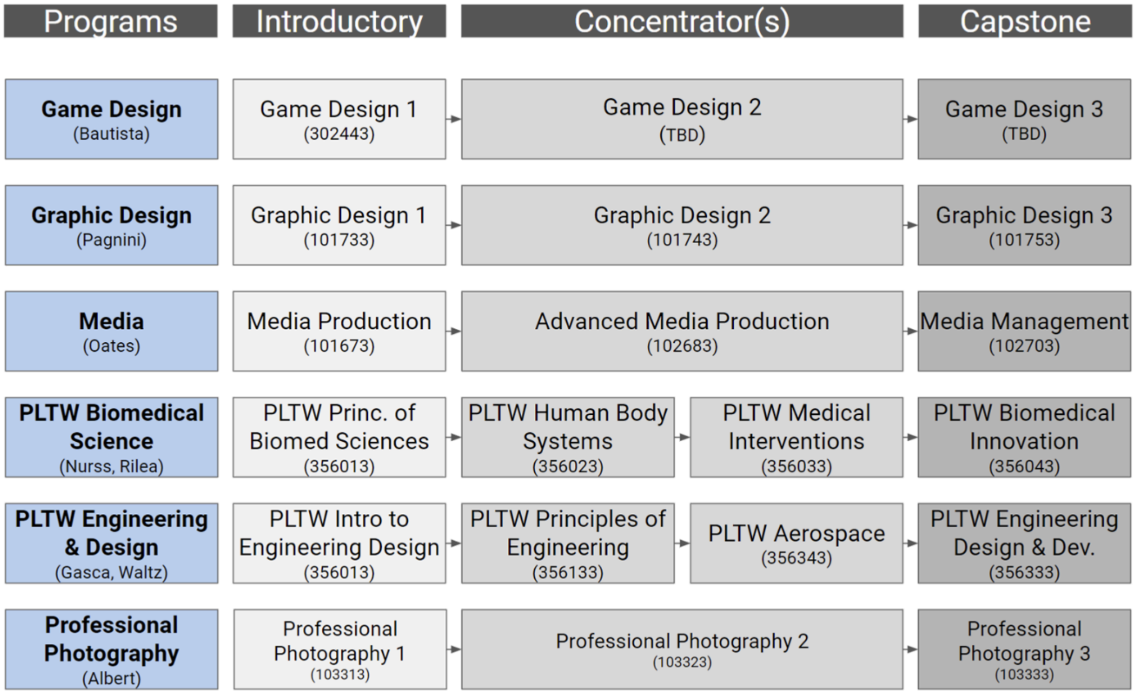
During the convening, participants were invited to evaluate the features of 12 extant pathways (from a variety of disciplines) in preparation for working on CS implementation pathways.

The most commonly identified strong features of pathways were:

- **Simple design:** Avoid too many, crossing, or curvy arrows and simple shapes.
- **Connected to purpose**: Identify interests, careers, and outcomes (e.g., college majors) that align to or extend from each pathway. One participant noted, “Having students’ interest be an asset and a factor in their course selection is something students have been asking for (at least my students have been asking me!)”
- **Clear sequence:** Label progressive content and indicate the direction of the progression – e.g., what is foundational vs. advanced.
- **Brief, clear, descriptive, understandable language**: Use sufficient but not overly detailed information. Define/explain terms when necessary for the audience.
- **Flexible**: Avoid rigid, strictly linear pathways (e.g., only one entry point, no opportunity to adjust/switch).
- **Strategic color coding**.

Most Popular Models

We asked participants to select their 1 - 3 favorite example implementation pathways by placing sticker dots on posters. The three most popular examples are shown below in Figures 14, 15, & 16.



Programs	Introductory	Concentrator(s)		Capstone
Game Design (Bautista)	Game Design 1 (302443)	Game Design 2 (TBD)		Game Design 3 (TBD)
Graphic Design (Pagnini)	Graphic Design 1 (101733)	Graphic Design 2 (101743)		Graphic Design 3 (101753)
Media (Oates)	Media Production (101673)	Advanced Media Production (102683)		Media Management (102703)
PLTW Biomedical Science (Nurss, Rilea)	PLTW Princ. of Biomed Sciences (356013)	PLTW Human Body Systems (356023)	PLTW Medical Interventions (356033)	PLTW Biomedical Innovation (356043)
PLTW Engineering & Design (Gasca, Waltz)	PLTW Intro to Engineering Design (356013)	PLTW Principles of Engineering (356133)	PLTW Aerospace (356343)	PLTW Engineering Design & Dev. (356333)
Professional Photography (Albert)	Professional Photography 1 (103313)	Professional Photography 2 (103323)		Professional Photography 3 (103333)

Figure 14. *West Park High School CTE Program of Study*.

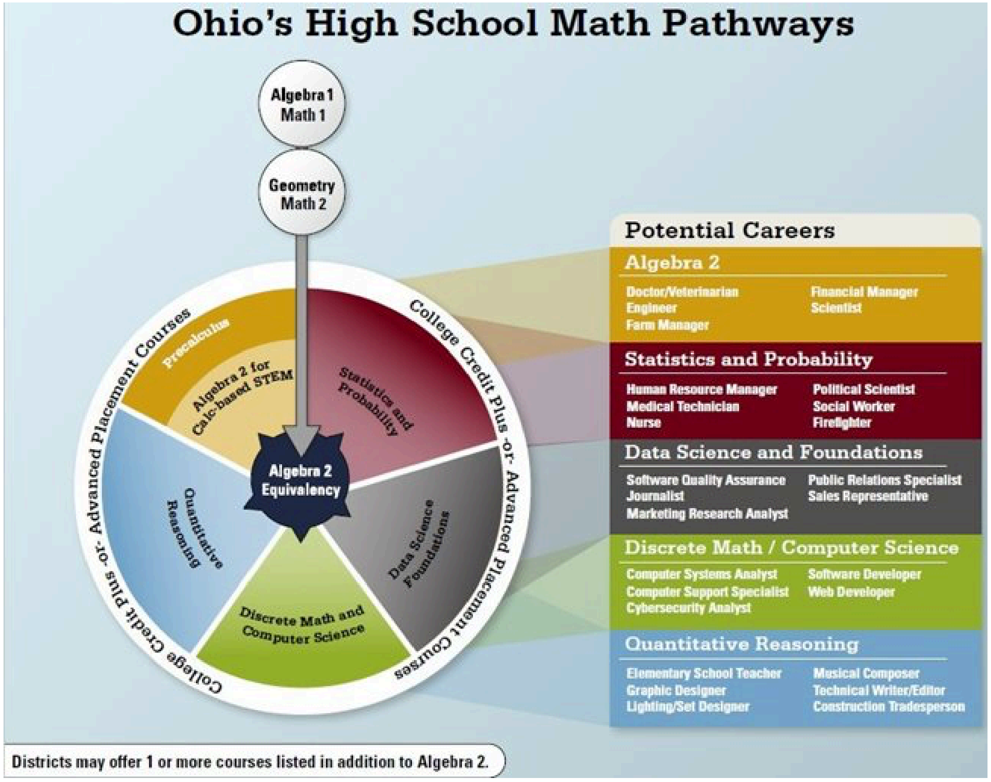


Figure 15. [*Ohio's High School Math Pathways*]().

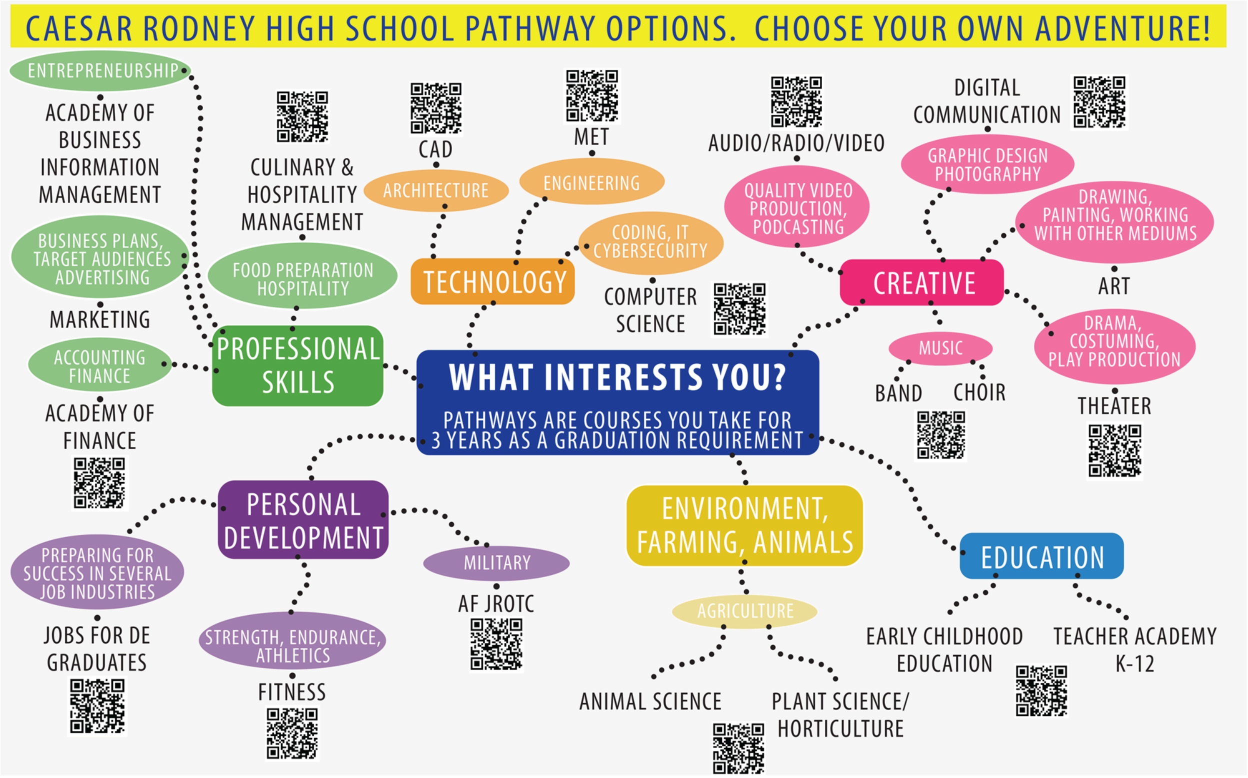


Figure 16. [*Caesar Rodney School District Career and Technical Education*]().

Least Popular Models

Additionally, examples of the *least* popular options are shown in Figures 17, 18, and 19 below.

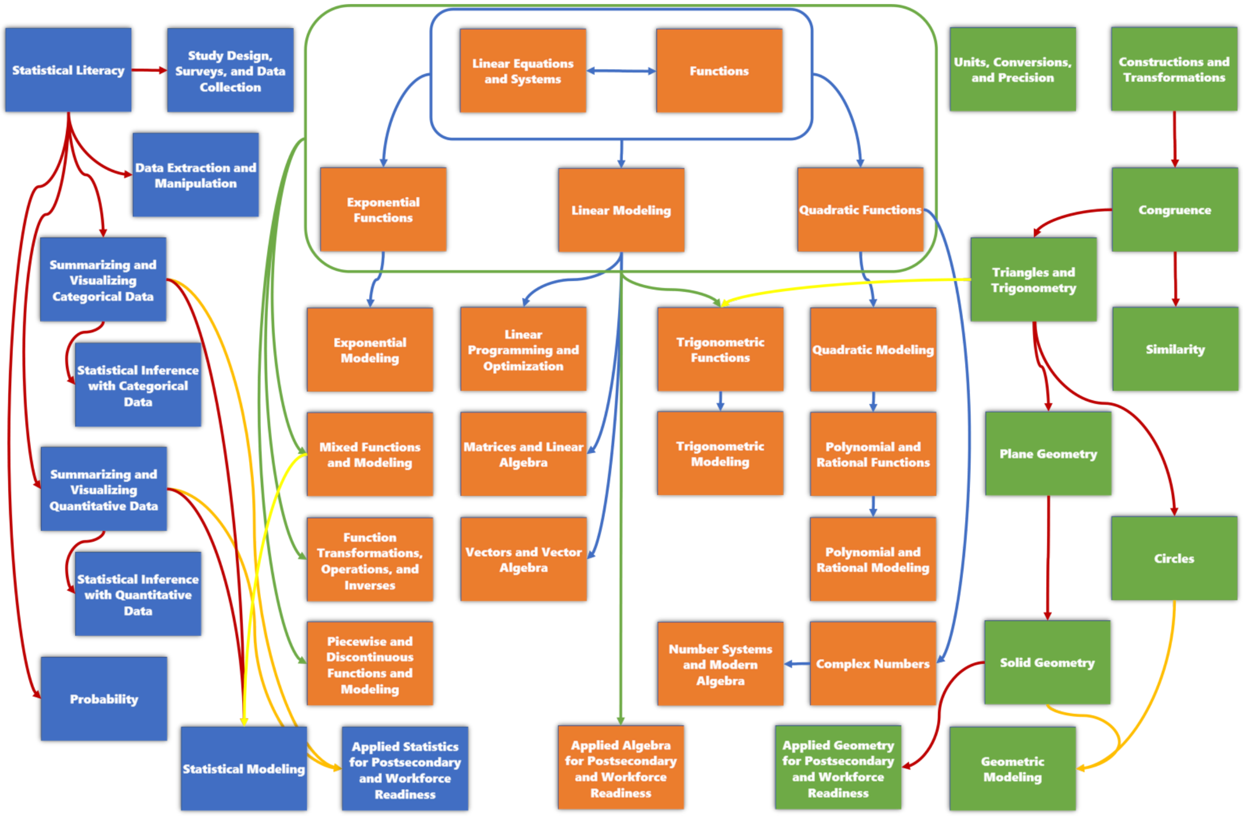
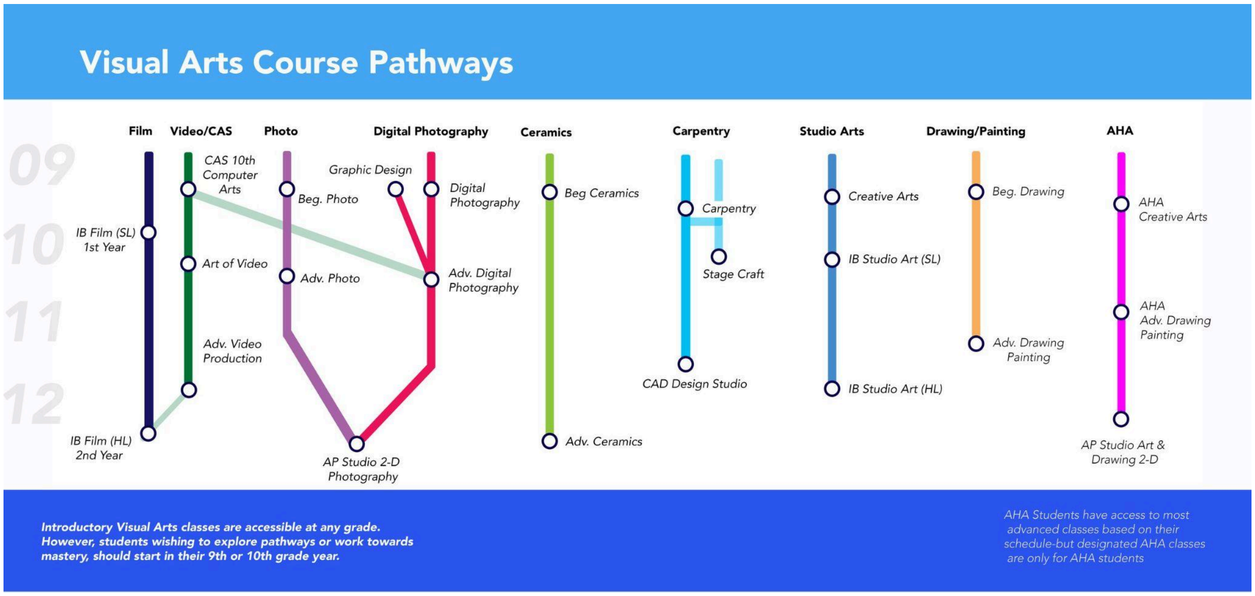


Figure 17. *[Colorado Department of Education Math Topics Pathways](#).*


Visual Arts Course Pathways

Introductory Visual Arts classes are accessible at any grade. However, students wishing to explore pathways or work towards mastery, should start in their 9th or 10th grade year.

AHA Students have access to most advanced classes based on their schedule-but designated AHA classes are only for AHA students

Figure 18. *[Berkeley High School Art Department: Visual Arts Course Pathways](#).*

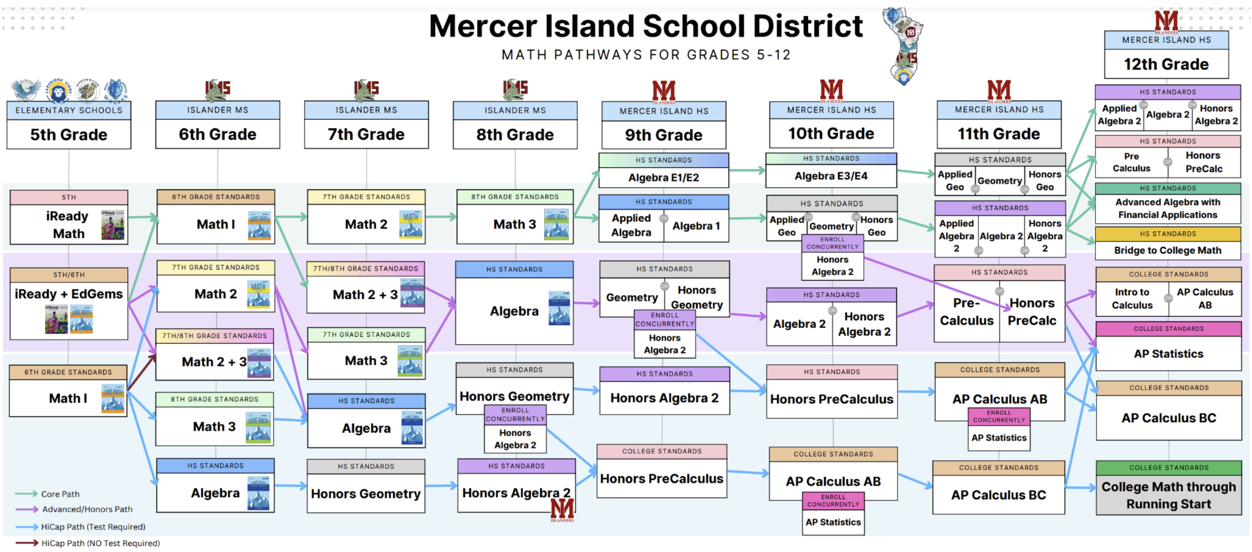


Figure 19. [Mercer Island School District Math Pathways](#).

Additional Models

Six additional pathway models are shown in Figures 20, 21, 22, 23, 24, and 25.

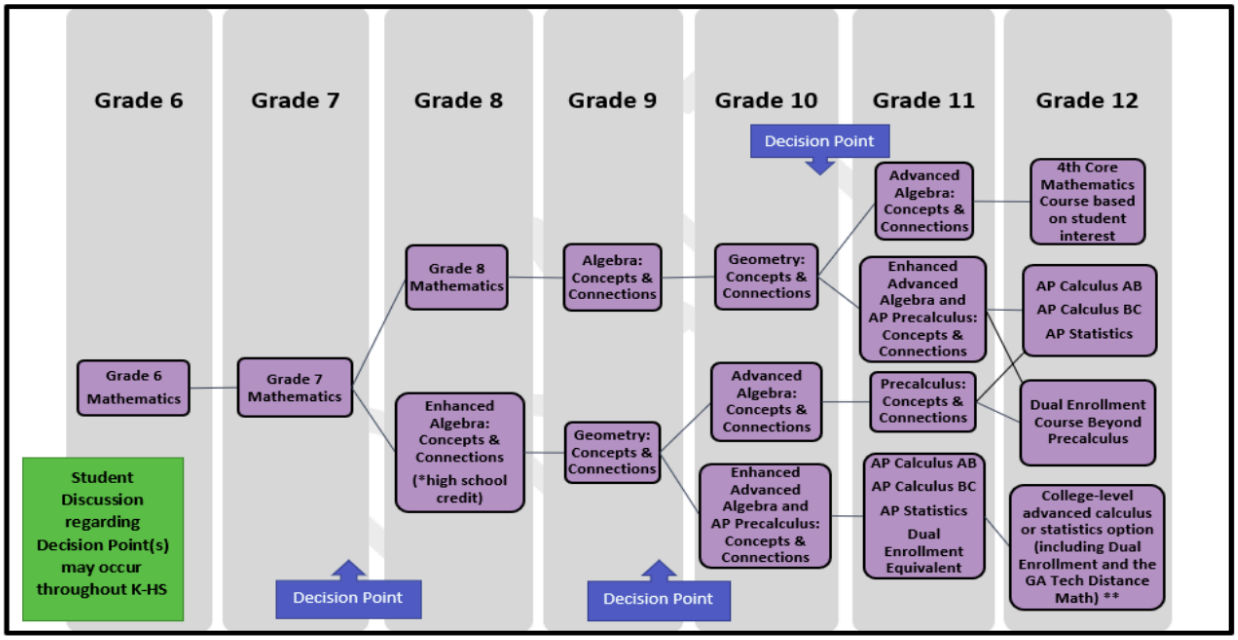


Figure 20. [Georgia's K-12 Mathematics Standards Open Access Pathways for Middle and High School](#).

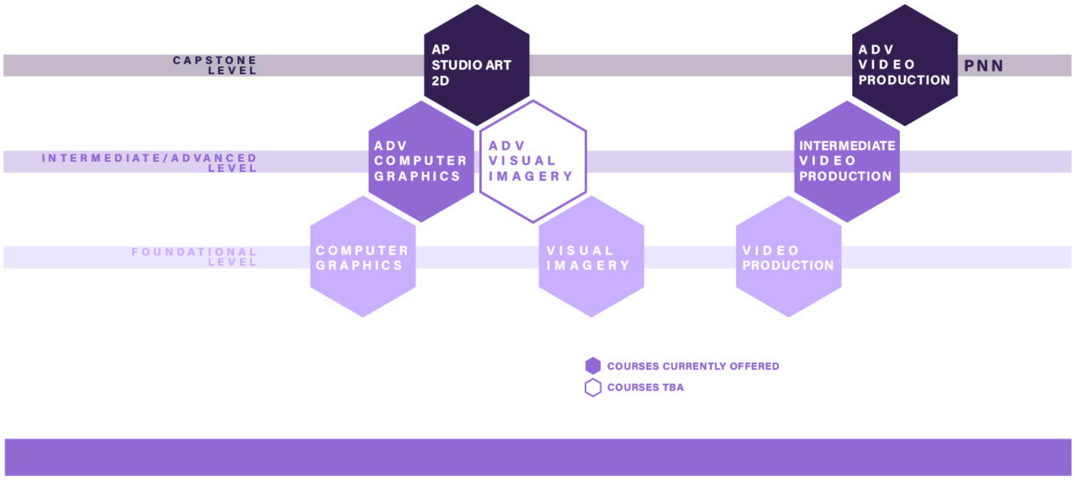


Figure 21. [Portola High School Digital Media Pathways](#).

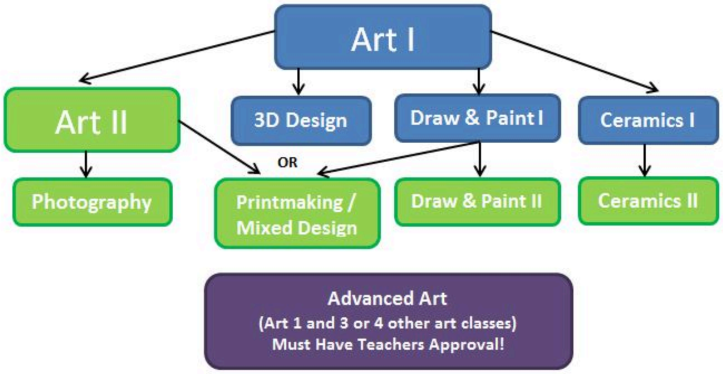


Figure 22. [Central High School Art Pathways](#).

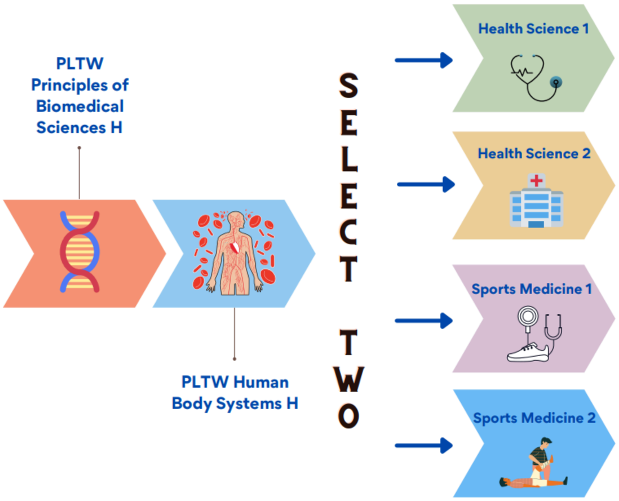


Figure 23. [West Ashley Center for Advanced Studies PLTW Biomedical Sciences Pathway](#).

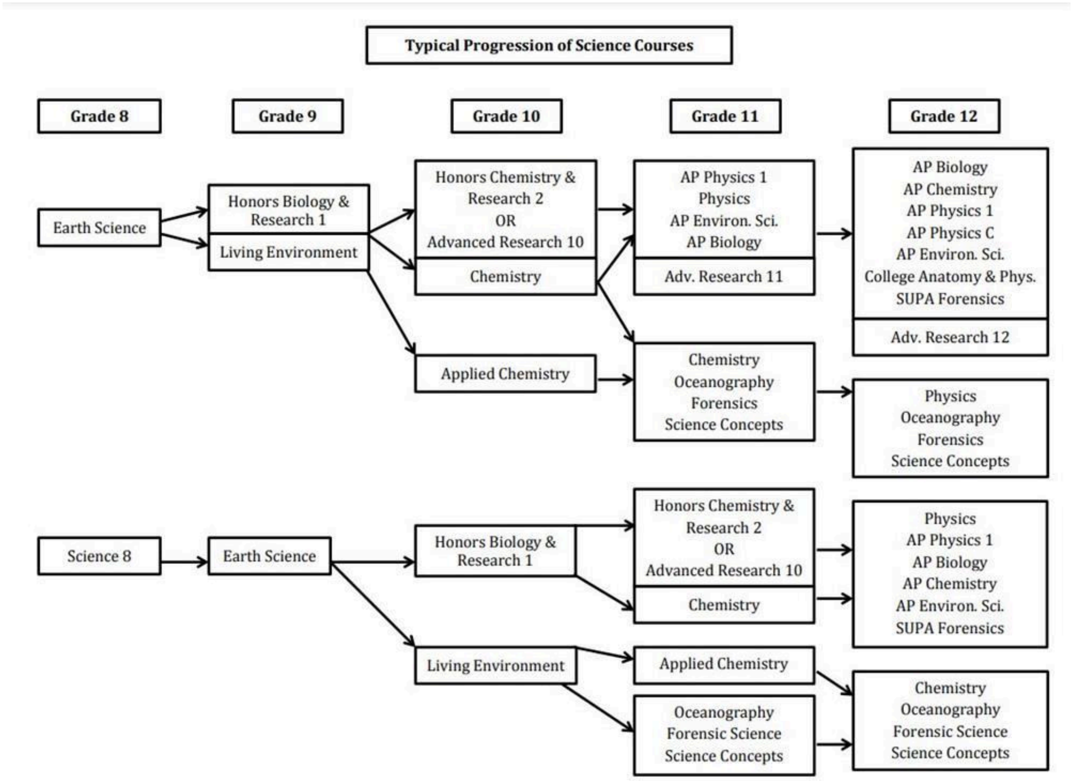


Figure 24. [Syosset Central School District Science Pathways]().

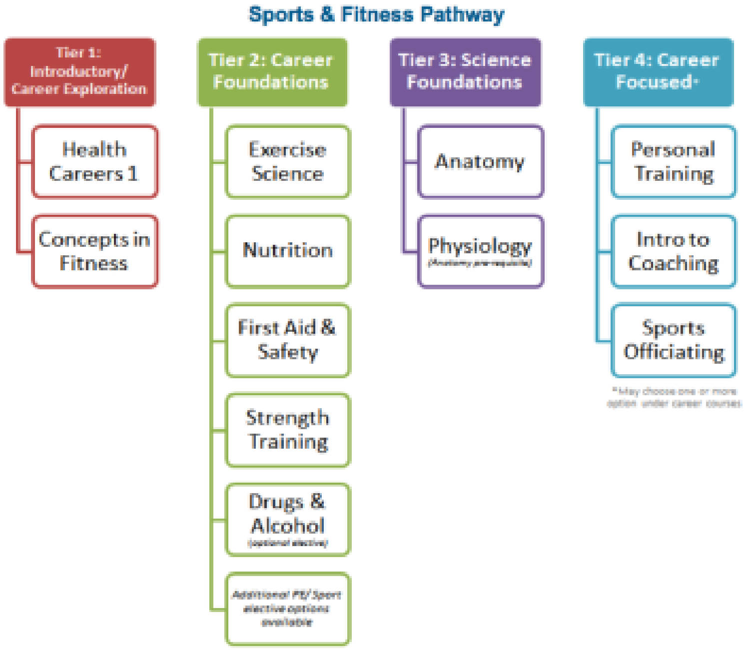
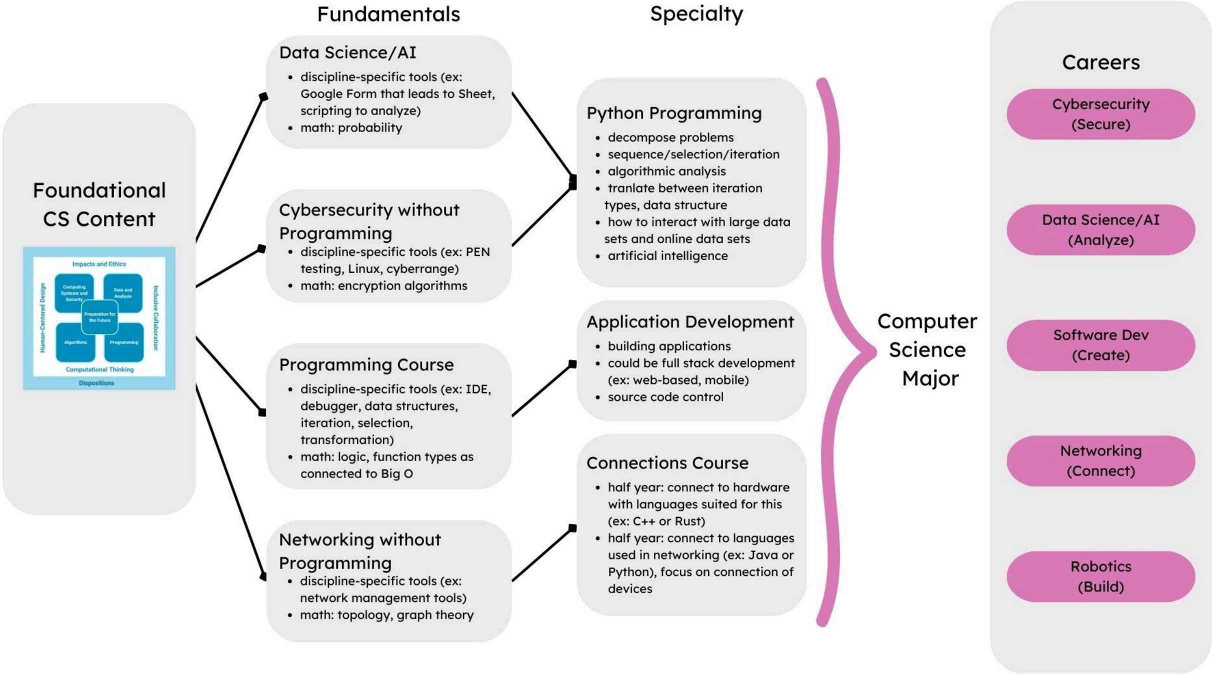


Figure 25. [Carone Learning Sports and Fitness Course Pathway]()

Appendix D: Participants' Content Progression Designs

Computer Science



Data Science/AI

- discipline-specific tools (ex: Google Form that leads to Sheet, scripting to analyze)
- math: probability

Cybersecurity without Programming

- discipline-specific tools (ex: PEN testing, Linux, cyberrange)
- math: encryption algorithms

Programming Course

- discipline-specific tools (ex: IDE, debugger, data structures, iteration, selection, transformation)
- math: logic, function types as connected to Big O

Networking without Programming

- discipline-specific tools (ex: network management tools)
- math: topology, graph theory

Python Programming

- decompose problems
- sequence/selection/iteration
- algorithmic analysis
- tranlate between iteration types, data structure
- how to interact with large data sets and online data sets
- artificial intelligence

Application Development

- building applications
- could be full stack development (ex: web-based, mobile)
- source code control

Connections Course

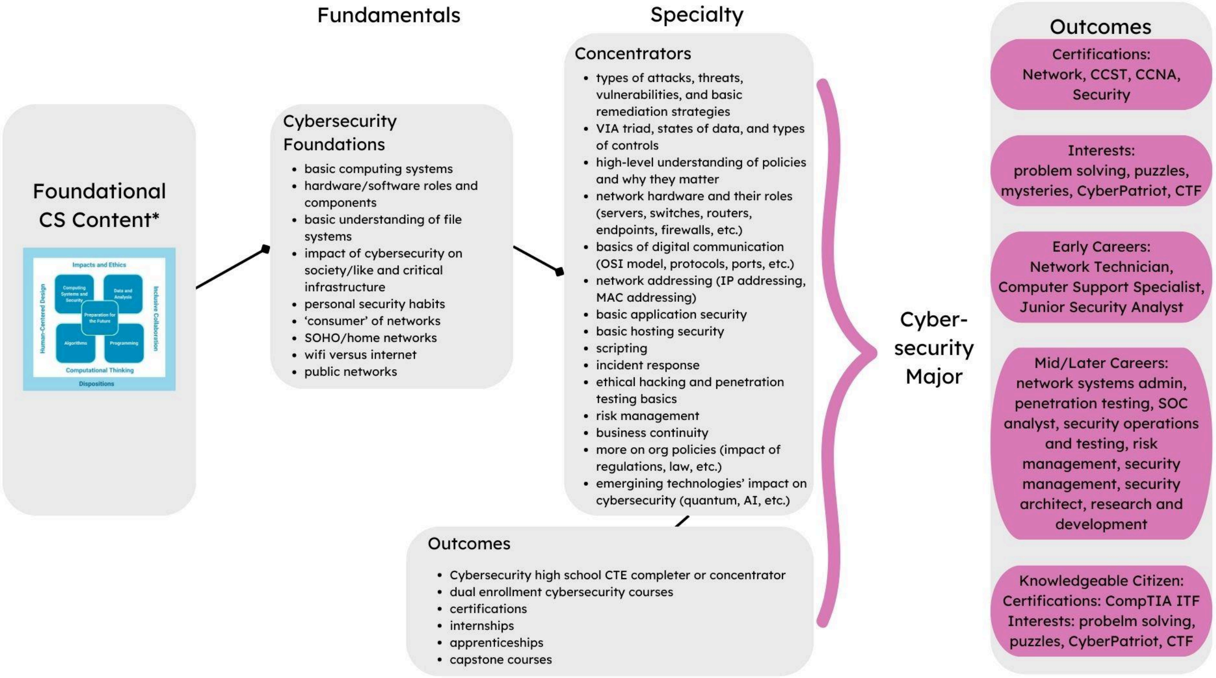
- half year: connect to hardware with languages suited for this (ex: C++ or Rust)
- half year: connect to languages used in networking (ex: Java or Python), focus on connection of devices

Computer Science Major

Careers

- Cybersecurity (Secure)
- Data Science/AI (Analyze)
- Software Dev (Create)
- Networking (Connect)
- Robotics (Build)

Cybersecurity



Cybersecurity Foundations

- basic computing systems
- hardware/software roles and components
- basic understanding of file systems
- impact of cybersecurity on society/like and critical infrastructure
- personal security habits
- 'consumer' of networks
- SOHO/home networks
- wifi versus internet
- public networks

Concentrators

- types of attacks, threats, vulnerabilities, and basic remediation strategies
- VIA triad, states of data, and types of controls
- high-level understanding of policies and why they matter
- network hardware and their roles (servers, switches, routers, endpoints, firewalls, etc.)
- basics of digital communication (OSI model, protocols, ports, etc.)
- network addressing (IP addressing, MAC addressing)
- basic application security
- basic hosting security
- scripting
- incident response
- ethical hacking and penetration testing basics
- risk management
- business continuity
- more on org policies (impact of regulations, law, etc.)
- emergining technologies' impact on cybersecurity (quantum, AI, etc.)

Outcomes

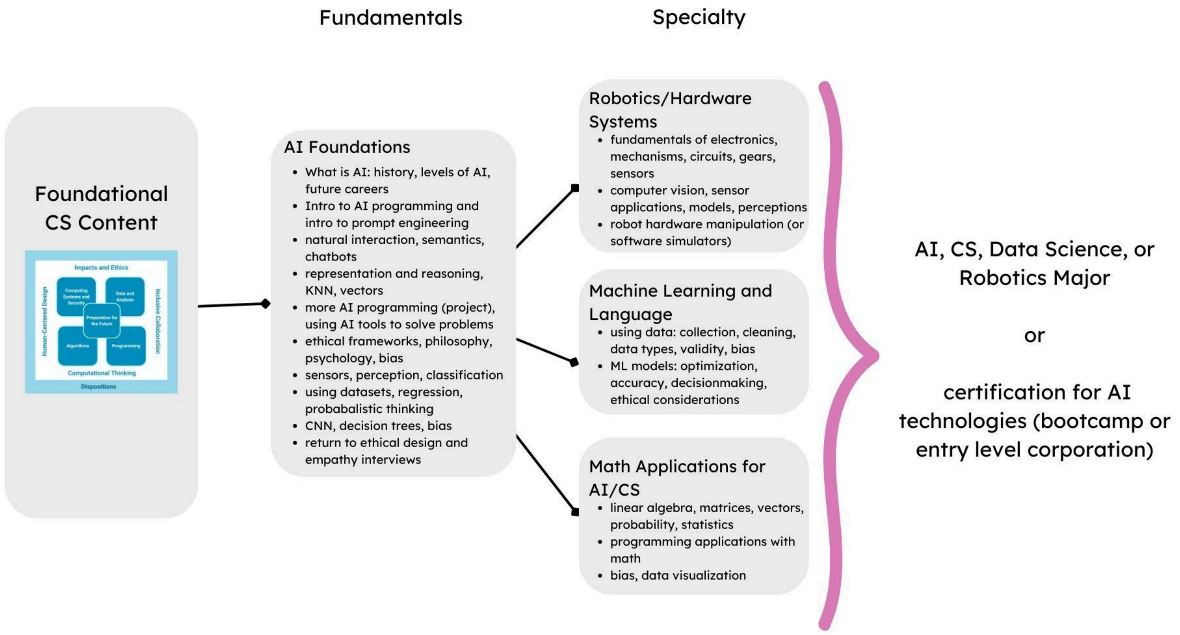
- Cybersecurity high school CTE completer or concentrator
- dual enrollment cybersecurity courses
- certifications
- internships
- apprenticeships
- capstone courses

Cyber-security Major

Outcomes

- Certifications: Network, CCST, CCNA, Security
- Interests: problem solving, puzzles, mysteries, CyberPatriot, CTF
- Early Careers: Network Technician, Computer Support Specialist, Junior Security Analyst
- Mid/Later Careers: network systems admin, penetration testing, SOC analyst, security operations and testing, risk management, security management, security architect, research and development
- Knowledgeable Citizen: Certifications: CompTIA ITF Interests: probelm solving, puzzles, CyberPatriot, CTF

Artificial Intelligence



Fundamentals

Specialty

Foundational CS Content

AI Foundations

- What is AI: history, levels of AI, future careers
- Intro to AI programming and intro to prompt engineering
- natural interaction, semantics, chatbots
- representation and reasoning, KNN, vectors
- more AI programming (project), using AI tools to solve problems
- ethical frameworks, philosophy, psychology, bias
- sensors, perception, classification
- using datasets, regression, probabalistic thinking
- CNN, decision trees, bias
- return to ethical design and empathy interviews

Robotics/Hardware Systems

- fundamentals of electronics, mechanisms, circuits, gears, sensors
- computer vision, sensor applications, models, perceptions
- robot hardware manipulation (or software simulators)

Machine Learning and Language

- using data: collection, cleaning, data types, validity, bias
- ML models: optimization, accuracy, decisionmaking, ethical considerations

Math Applications for AI/CS

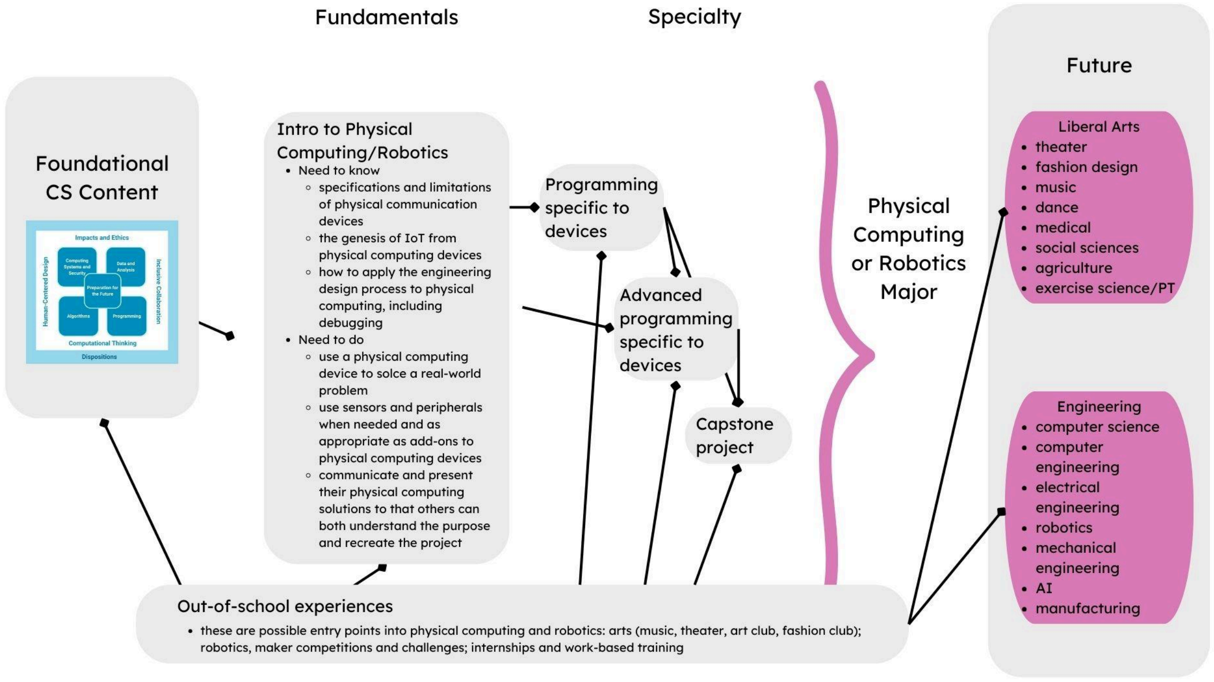
- linear algebra, matrices, vectors, probability, statistics
- programming applications with math
- bias, data visualization

AI, CS, Data Science, or Robotics Major

or

certification for AI technologies (bootcamp or entry level corporation)

Physical Computing or Robotics



Fundamentals

Specialty

Foundational CS Content

Intro to Physical Computing/Robotics

- Need to know
  - specifications and limitations of physical communication devices
  - the genesis of IoT from physical computing devices
  - how to apply the engineering design process to physical computing, including debugging
- Need to do
  - use a physical computing device to solce a real-world problem
  - use sensors and peripherals when needed and as appropriate as add-ons to physical computing devices
  - communicate and present their physical computing solutions to that others can both understand the purpose and recreate the project

Programming specific to devices

Advanced programming specific to devices

Capstone project

Physical Computing or Robotics Major

Out-of-school experiences

- these are possible entry points into physical computing and robotics: arts (music, theater, art club, fashion club); robotics, maker competitions and challenges; internships and work-based training

Future

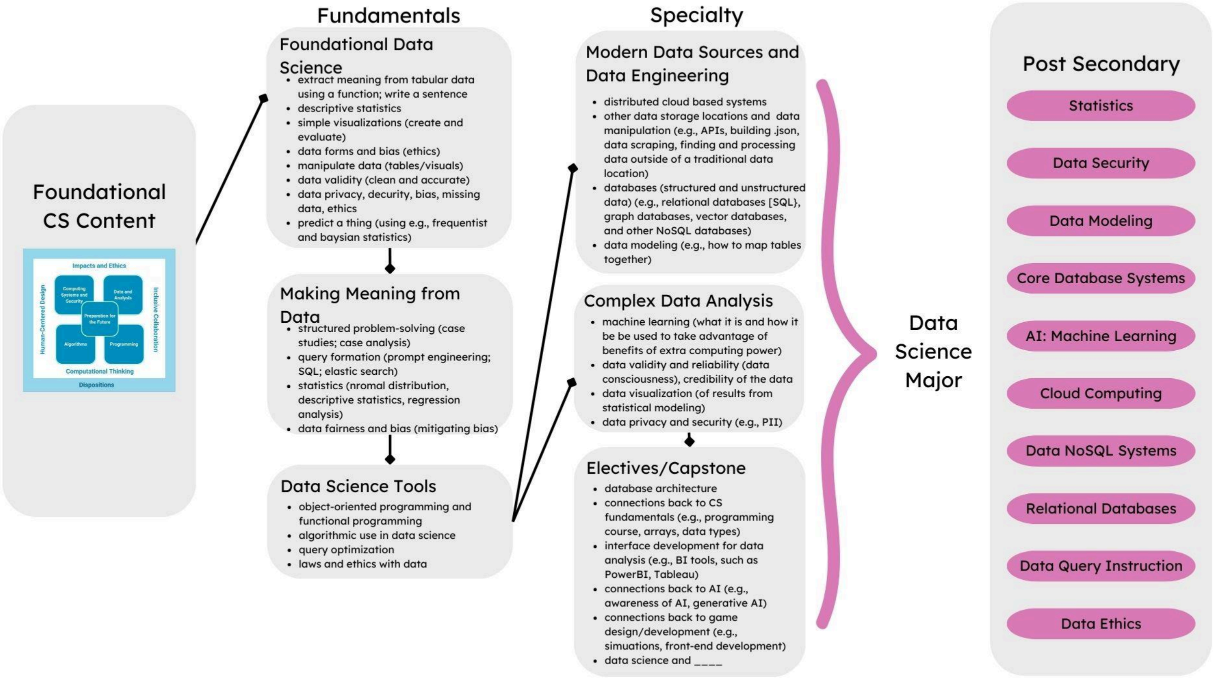
Liberal Arts

- theater
- fashion design
- music
- dance
- medical
- social sciences
- agriculture
- exercise science/PT

Engineering

- computer science
- computer engineering
- electrical engineering
- robotics
- mechanical engineering
- AI
- manufacturing

Data Science



Foundational CS Content

Fundamentals

Foundational Data Science
- extract meaning from tabular data using a function; write a sentence
- descriptive statistics
- simple visualizations (create and evaluate)
- data forms and bias (ethics)
- manipulate data (tables/visuals)
- data validity (clean and accurate)
- data privacy, decurity, bias, missing data, ethics
- predict a thing (using e.g., frequentist and baysian statistics)

Making Meaning from Data
- structured problem-solving (case studies; case analysis)
- query formation (prompt engineering; SQL; elastic search)
- statistics (nromal distribution, descriptive statistics, regression analysis)
- data fairness and bias (mitigating bias)

Data Science Tools
- object-oriented programming and functional programming
- algorithmic use in data science
- query optimization
- laws and ethics with data

Specialty

Modern Data Sources and Data Engineering
- distributed cloud based systems
- other data storage locations and data manipulation (e.g., APIs, building .json, data scraping, finding and processing data outside of a traditional data location)
- databases (structured and unstructured data) (e.g., relational databases [SQL}, graph databases, vector databases, and other NoSQL databases)
- data modeling (e.g., how to map tables together)

Complex Data Analysis
- machine learning (what it is and how it be be used to take advantage of benefits of extra computing power)
- data validity and reliability (data consciousness), credibility of the data
- data visualization (of results from statistical modeling)
- data privacy and security (e.g., PII)

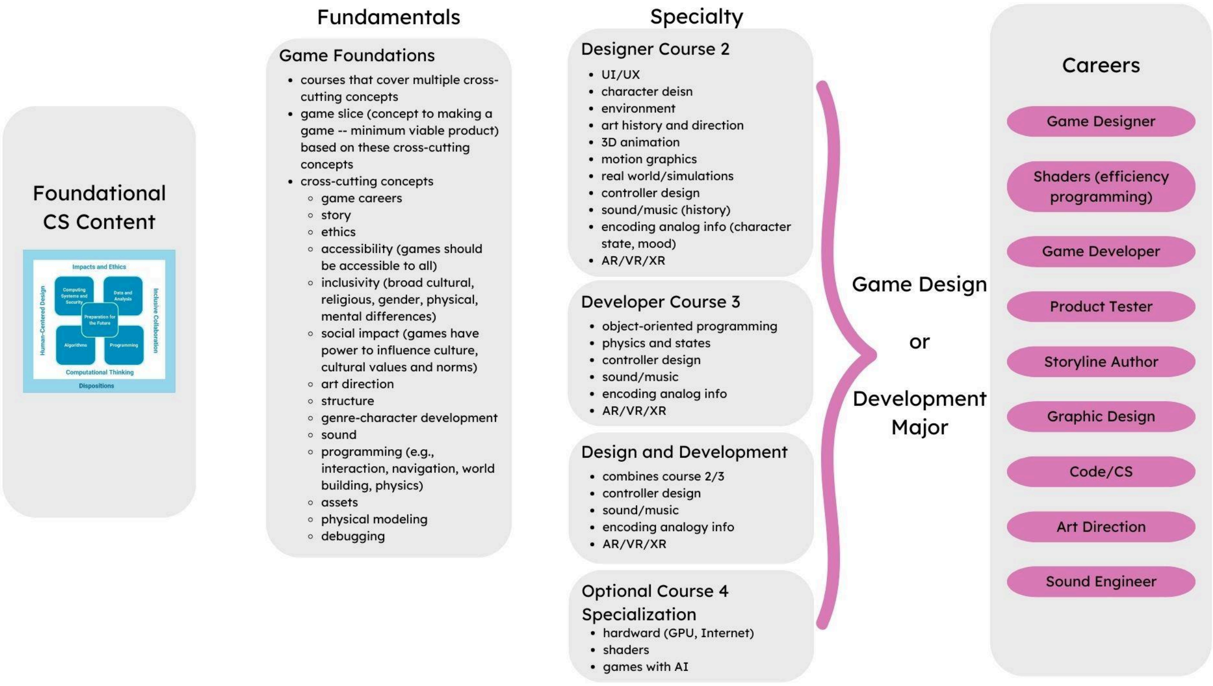
Electives/Capstone
- database architecture
- connections back to CS fundamentals (e.g., programming course, arrays, data types)
- interface development for data analysis (e.g., BI tools, such as PowerBI, Tableau)
- connections back to AI (e.g., awareness of AI, generative AI)
- connections back to game design/development (e.g., simuations, front-end development)
- data science and ____

Data Science Major

Post Secondary

- Statistics
- Data Security
- Data Modeling
- Core Database Systems
- AI: Machine Learning
- Cloud Computing
- Data NoSQL Systems
- Relational Databases
- Data Query Instruction
- Data Ethics

Game Design and Development



Foundational CS Content

Fundamentals

Game Foundations
- courses that cover multiple cross-cutting concepts
- game slice (concept to making a game -- minimum viable product) based on these cross-cutting concepts
- cross-cutting concepts
    - game careers
    - story
    - ethics
    - accessibility (games should be accessible to all)
    - inclusivity (broad cultural, religious, gender, physical, mental differences)
    - social impact (games have power to influence culture, cultural values and norms)
    - art direction
    - structure
    - genre-character development
    - sound
    - programming (e.g., interaction, navigation, world building, physics)
    - assets
    - physical modeling
    - debugging

Specialty

Designer Course 2
- UI/UX
- character deisn
- environment
- art history and direction
- 3D animation
- motion graphics
- real world/simulations
- controller design
- sound/music (history)
- encoding analog info (character state, mood)
- AR/VR/XR

Developer Course 3
- object-oriented programming
- physics and states
- controller design
- sound/music
- encoding analog info
- AR/VR/XR

Design and Development
- combines course 2/3
- controller design
- sound/music
- encoding analogy info
- AR/VR/XR

Optional Course 4 Specialization
- hardward (GPU, Internet)
- shaders
- games with AI

Game Design

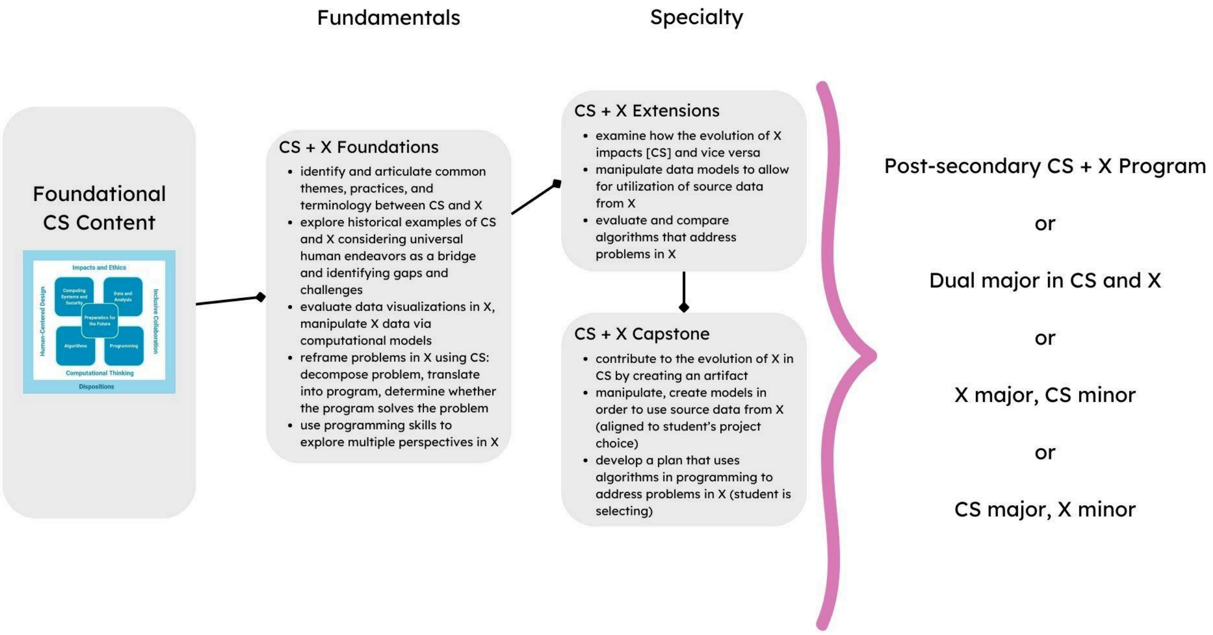
or

Development Major

Careers

- Game Designer
- Shaders (efficiency programming)
- Game Developer
- Product Tester
- Storyline Author
- Graphic Design
- Code/CS
- Art Direction
- Sound Engineer

CS + Humanities



Fundamentals

Specialty

Foundational CS Content

CS + X Foundations

- identify and articulate common themes, practices, and terminology between CS and X
- explore historical examples of CS and X considering universal human endeavors as a bridge and identifying gaps and challenges
- evaluate data visualizations in X, manipulate X data via computational models
- reframe problems in X using CS: decompose problem, translate into program, determine whether the program solves the problem
- use programming skills to explore multiple perspectives in X

CS + X Extensions

- examine how the evolution of X impacts [CS] and vice versa
- manipulate data models to allow for utilization of source data from X
- evaluate and compare algorithms that address problems in X

CS + X Capstone

- contribute to the evolution of X in CS by creating an artifact
- manipulate, create models in order to use source data from X (aligned to student's project choice)
- develop a plan that uses algorithms in programming to address problems in X (student is selecting)

Post-secondary CS + X Program

or

Dual major in CS and X

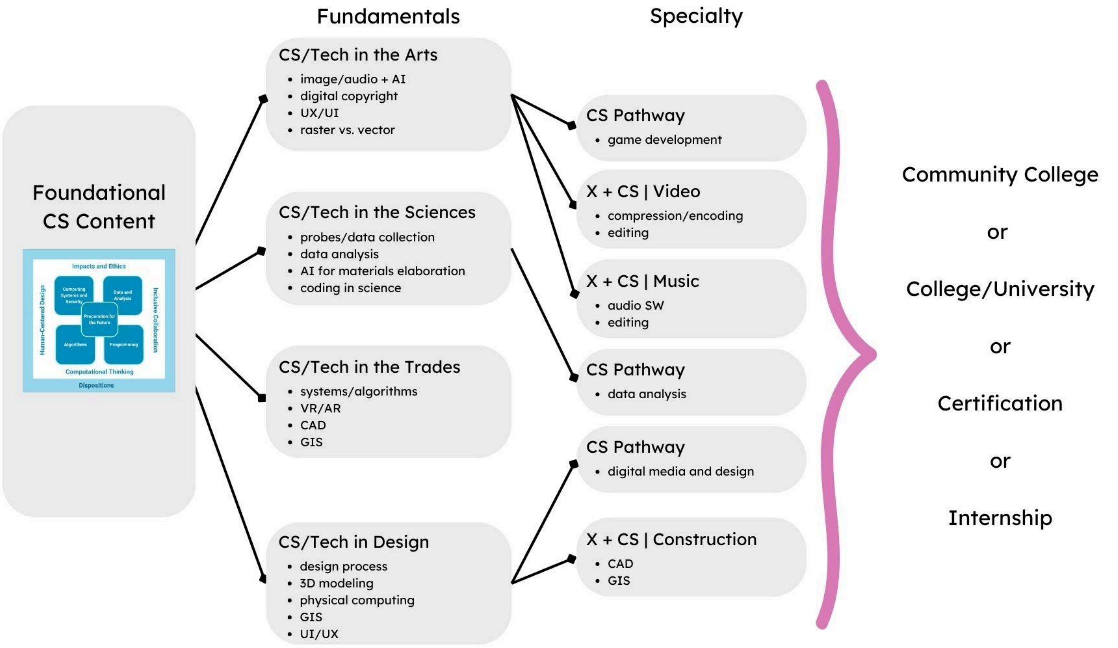
or

X major, CS minor

or

CS major, X minor

X + CS (Integration)



Fundamentals

Specialty

Foundational CS Content

CS/Tech in the Arts

- image/audio + AI
- digital copyright
- UX/UI
- raster vs. vector

CS/Tech in the Sciences

- probes/data collection
- data analysis
- AI for materials elaboration
- coding in science

CS/Tech in the Trades

- systems/algorithms
- VR/AR
- CAD
- GIS

CS/Tech in Design

- design process
- 3D modeling
- physical computing
- GIS
- UI/UX

CS Pathway

- game development

X + CS | Video

- compression/encoding
- editing

X + CS | Music

- audio SW
- editing

CS Pathway

- data analysis

CS Pathway

- digital media and design

X + CS | Construction

- CAD
- GIS

Community College

or

College/University

or

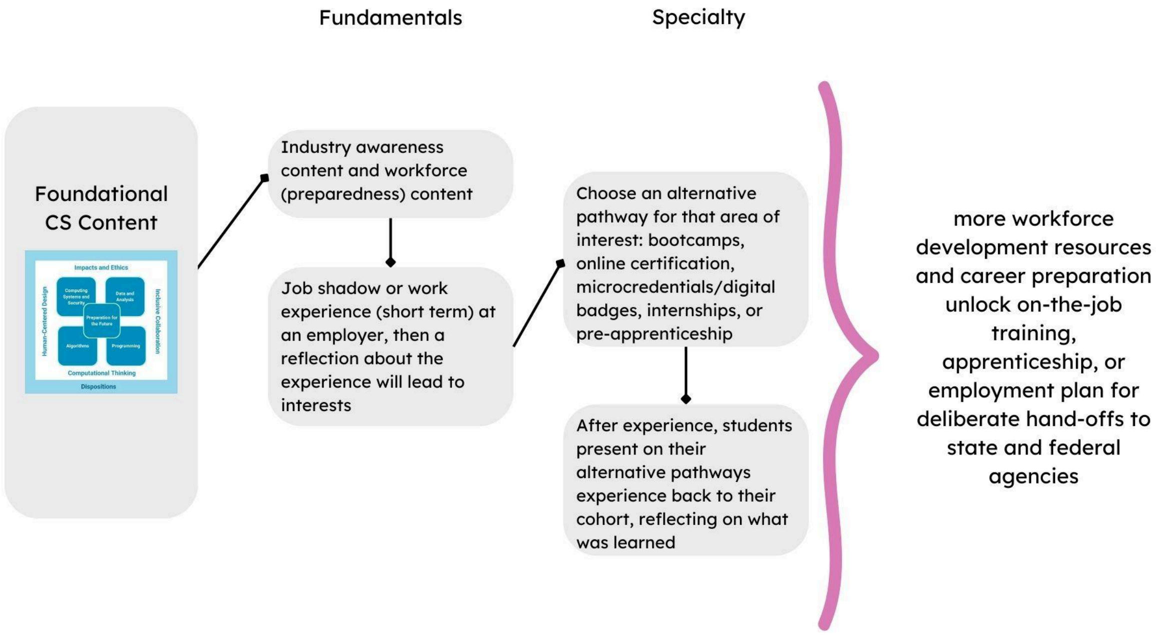
Certification

or

Internship

See also Appendix E: X + CS Implementation Details.

Alternative CS Pathways



Fundamentals

Specialty

Foundational CS Content

Industry awareness content and workforce (preparedness) content

Job shadow or work experience (short term) at an employer, then a reflection about the experience will lead to interests

Choose an alternative pathway for that area of interest: bootcamps, online certification, microcredentials/digital badges, internships, or pre-apprenticeship

After experience, students present on their alternative pathways experience back to their cohort, reflecting on what was learned

more workforce development resources and career preparation unlock on-the-job training, apprenticeship, or employment plan for deliberate hand-offs to state and federal agencies

Appendix E: X + CS Implementation Details

The X + CS content progression (see section 4.7) was organized around several guiding questions. The question, ***How do the ways of knowing and practices in X and CS enhance and transform learning in both?*** spans all coursework and topics. The table shows the relationship between other guiding questions and content for this pathway.

Guiding Question	Course 1 Content	Course 2 Content	Course 3 Content
Prepare for the future: How might the integration of CS transform learning in X?	Identify and articulate common themes, practices, and terminology between CS and X.	Examine how the evolution of X impacts CS and vice versa.	Contribute to the evolution of X in CS by creating an artifact.
Data and analysis: How might exploring data from X using CS's data and analysis methods transform student learning of X?	Evaluate data visualizations in X; manipulate X data via computational models.	Manipulate data models to allow for utilization of source data from X.	Manipulate, create models in order to use source data from X.
Algorithms and programming: How might interpreting, modifying, and composing algorithms transform learning in X?	Reframe problems in X using CS; use programming skills to explore multiple perspectives in X.	Evaluate and compare algorithms that address problems in X.	Develop a plan that uses algorithms in programming to address problems in X.

Appendix F: Examples of Pathway Implementations

Participants were presented with descriptions of hypothetical high schools as shown in the tables describing high schools A, B, C, D, E, and F below. The characteristics of these high schools are slightly adapted from actual schools and their National Center for Education Statistics (NCES) data. Starting with the nine possible CS pathways (see section 4), participants developed an implementation that would be workable for each school. These implementations are included in this section; see Section 5 for a summary of the general principles and ideas for implementation.

High School A

- School Type: STEM-focused Charter School (Grades 9-12)
- Locale: City
- Student Population: 563
- Student Demographics:
    - 38% Black, 32% White, 25% Hispanic, <1% American Indian, Asian, and Multiracial
    - 66% from economically disadvantaged families
    - 66% male, 34% female
- 16% students with disabilities
- 30 teachers
- One full-time CS teacher

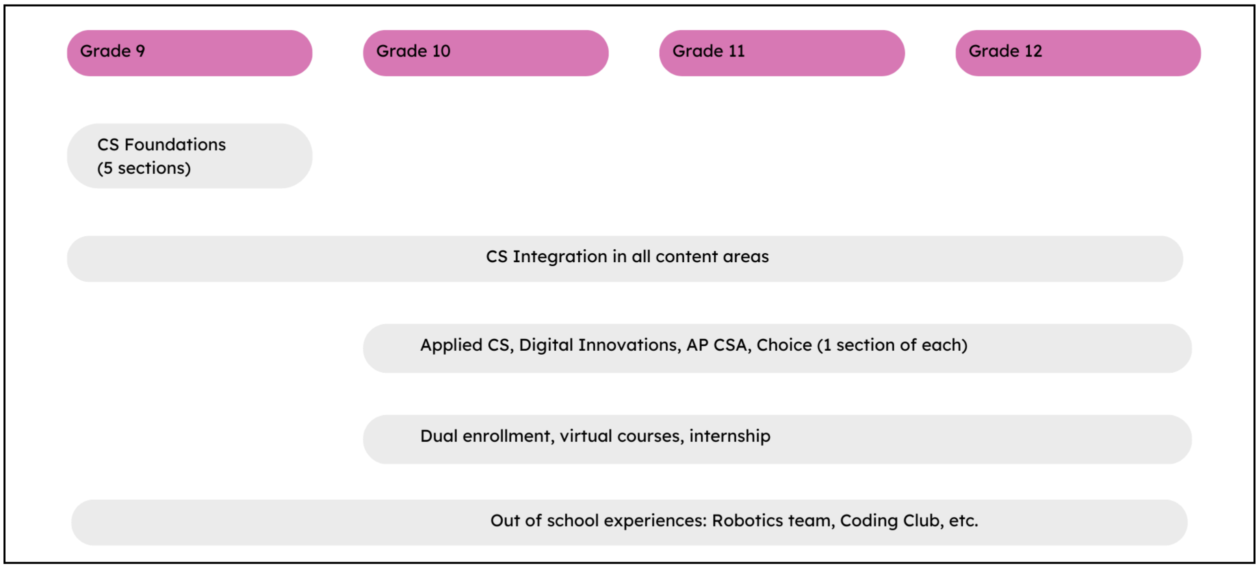
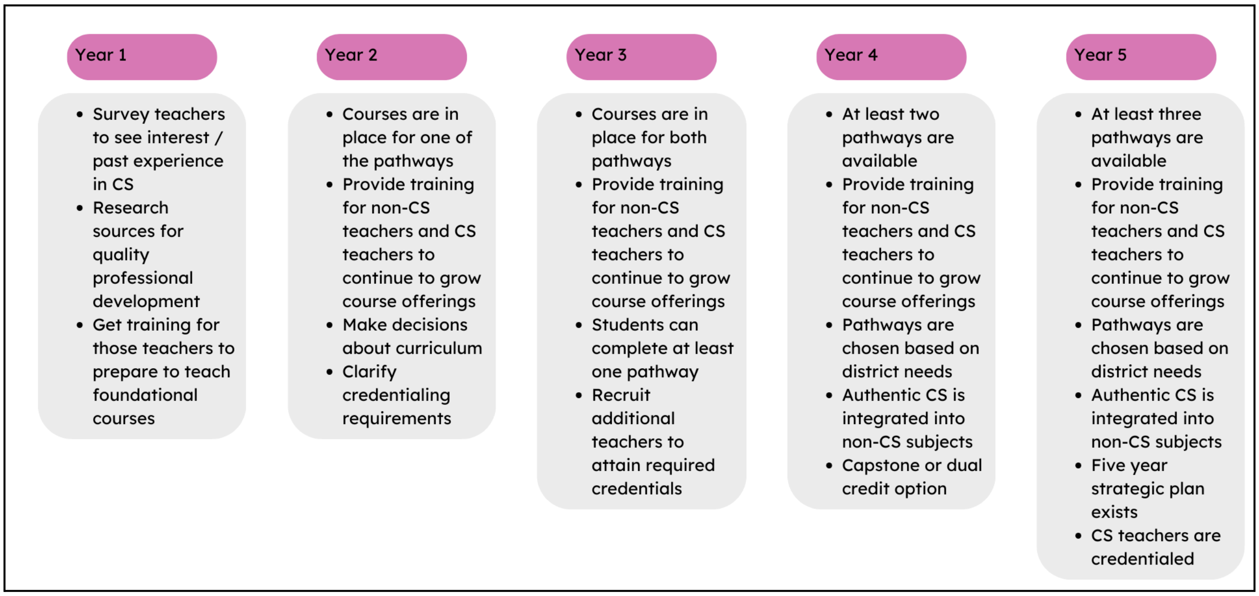


Figure 26. CS offerings at high school A.

High School B

- School Type: Traditional Public (Grades 9-12)
- Locale: Suburb
- Student Population: 2,157
- Student Demographics:
    - 70% Hispanic, 14% White, 8% Black, 4% Asian, 2% Multiracial, 1% American Indian, <1% Native Hawaiian/PI
    - 26% from economically disadvantaged families
    - 51% male, 49% female
- 89 teachers
- Two full-time CS teachers



Year 1

- Survey teachers to see interest / past experience in CS
- Research sources for quality professional development
- Get training for those teachers to prepare to teach foundational courses

Year 2

- Courses are in place for one of the pathways
- Provide training for non-CS teachers and CS teachers to continue to grow course offerings
- Make decisions about curriculum
- Clarify credentialing requirements

Year 3

- Courses are in place for both pathways
- Provide training for non-CS teachers and CS teachers to continue to grow course offerings
- Students can complete at least one pathway
- Recruit additional teachers to attain required credentials

Year 4

- At least two pathways are available
- Provide training for non-CS teachers and CS teachers to continue to grow course offerings
- Pathways are chosen based on district needs
- Authentic CS is integrated into non-CS subjects
- Capstone or dual credit option

Year 5

- At least three pathways are available
- Provide training for non-CS teachers and CS teachers to continue to grow course offerings
- Pathways are chosen based on district needs
- Authentic CS is integrated into non-CS subjects
- Five year strategic plan exists
- CS teachers are credentialed

Figure 27. CS implementation timeline for high school B. Note that this describes the school's implementation, not the student's experiences.

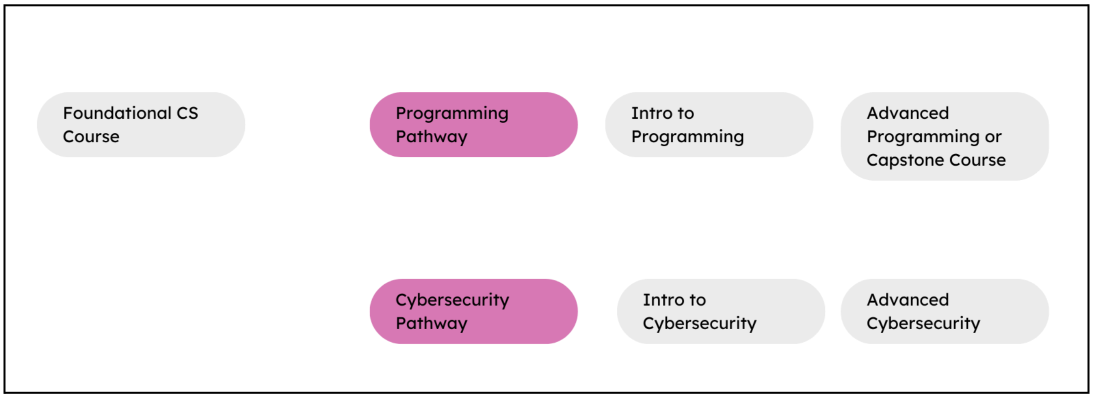
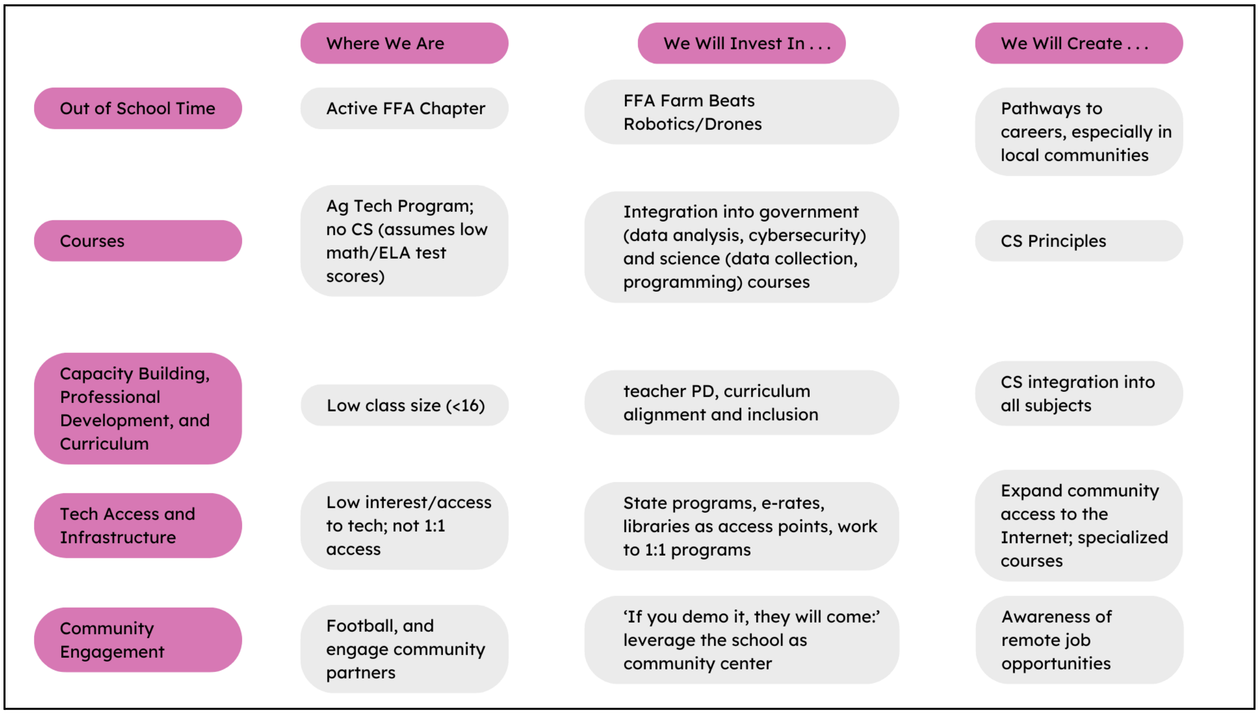


Figure 28. CS pathways for high school B.

High School C

- School Type: Traditional Public (Grades 9-12)
- Locale: Rural
- Student Population: 310
- Student Demographics:
    - 91% White, 5% Hispanic, 2% Multiracial, <1% Asian, Black, American Indiana, Native Hawaiian/PI
    - 18% from economically disadvantaged families
    - 52 male, 48% female
- 40% of students don't have access to high speed internet at home
- 27 teachers
- No current CS teacher



	Where We Are	We Will Invest In . . .	We Will Create . . .
Out of School Time	Active FFA Chapter	FFA Farm Beats Robotics/Drones	Pathways to careers, especially in local communities
Courses	Ag Tech Program; no CS (assumes low math/ELA test scores)	Integration into government (data analysis, cybersecurity) and science (data collection, programming) courses	CS Principles
Capacity Building, Professional Development, and Curriculum	Low class size (<16)	teacher PD, curriculum alignment and inclusion	CS integration into all subjects
Tech Access and Infrastructure	Low interest/access to tech; not 1:1 access	State programs, e-rates, libraries as access points, work to 1:1 programs	Expand community access to the Internet; specialized courses
Community Engagement	Football, and engage community partners	'If you demo it, they will come:' leverage the school as community center	Awareness of remote job opportunities

Figure 29. Map of high school C's current state, investment plan, and desired outcomes.

High School D

- School Type: Traditional Public (Grades 9-12)
- Locale: City
- Student Population: 4,106
- Student Demographics:
  - 31% White, 29% Black, 25% Hispanic, 9% Asian, 4% Multiracial, <1% American Indian and Native Hawaiian/PI
  - 57% from economically disadvantaged families
  - 47% male, 53% female
- 158 teachers
- 4.5 full-time CS teachers

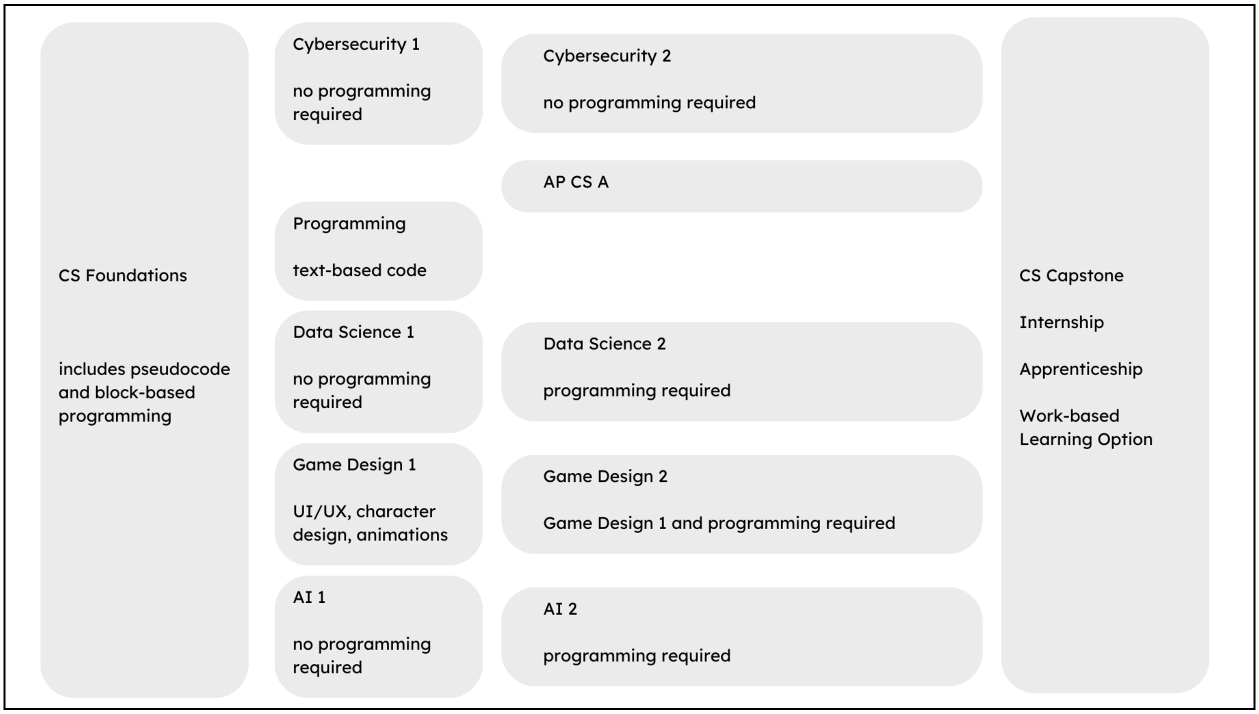


Figure 30. Pathways for high school D.

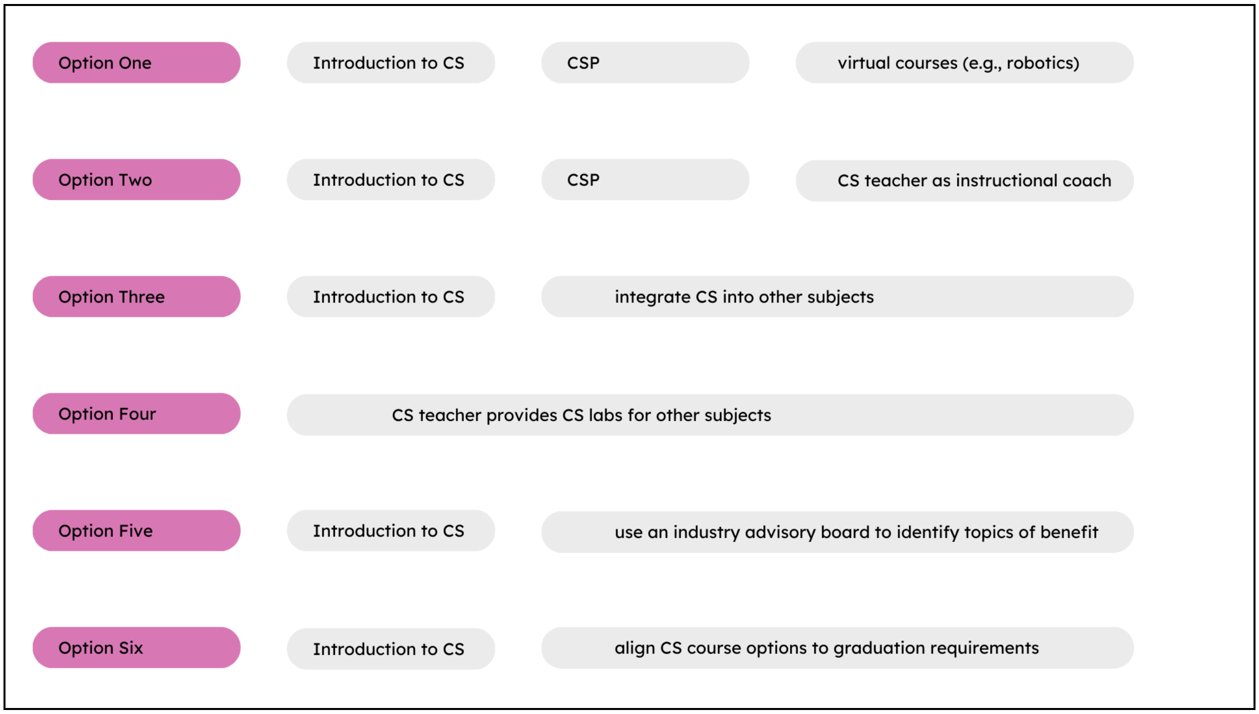
High School E

- School Type: Traditional Public (Grades 6-12)
- Locale: City
- Student Population: 655
- Student Demographics:
  - 50% White, 18% Hispanic, 14% Black, 8% Asian, 8% Multiracial, 1% American Indian, <1% Native Hawaiian/PI
  - 40% from economically disadvantaged families
  - 47% male, 53% female
- Highly transient student population: 45% of students who start 9th grade finish 12th grade at another school
- 47 teachers, with 0.75 full-time CS teacher

Note that there is no graphic for high school E.

High School F

- School Type: Traditional Public (Grades 9-12)
- Locale: Rural
- Student Population: 213
- Student Demographics:
    - 76% White, 13% Hispanic, 8% Multiracial, 2% American Indian, 1% Black, <1% Asian
    - 27% from economically disadvantaged families
    - 50% male, 50% female
- 12 teachers
- 0.5 full-time CS teacher



Option One	Introduction to CS	CSP	virtual courses (e.g., robotics)
Option Two	Introduction to CS	CSP	CS teacher as instructional coach
Option Three	Introduction to CS	integrate CS into other subjects	
Option Four	CS teacher provides CS labs for other subjects		
Option Five	Introduction to CS	use an industry advisory board to identify topics of benefit	
Option Six	Introduction to CS	align CS course options to graduation requirements	

Figure 31. Implementation options for high school F.

Appendix G: Professional Skills

One of the implementation pathways working groups also included discussion about professional skills that are important to develop. They recommended that professional skills include:

- Content designed to prepare students for the attitudinal and disciplinary rigors of employment and to provide a highly structured, simulated work environment that encourages the behavior required for workplace success
- Students take ownership of their decisions and develop concrete skills necessary to get a job, such as resume writing and interviewing
- Students learn the basic qualities that constitute a good employee: reliability, punctuality, a strong work ethic, and a positive demeanor
- Programs like WorkReady, in partnership with local businesses, provide students with real-time feedback through mock interviews and other workshops with human resource professionals; successful participants will receive a credential recognized by employers WorkReady is based on seven standards: personal motivations and challenges, plans for employment, working with others, effective communication, the principles of getting a job, employee rights, and work-related safety information