HANOVER RESEARCH

RESEARCH PRIORITY BRIEF

ELEMENTARY SCHOOL COMPUTER SCIENCE PROGRAMS

Introduction

In our increasingly connected, technology-driven, modernized world, educators are tasked with imparting both hardware and software skills to students in Grades K-12. Today's students must be taught "not just how to start [a computer] up and surf the web, but how they actually run, how – at the simplest level – a series of inputs leads to a series of particular outputs."[1]

However, very few schools formally structure computer science instruction at the high school and middle school levels – and even fewer incorporate computer science into the elementary school level.[2] Specifically, a survey conducted by Google and Gallup in 2016 across 16,000 students, parents, teachers, principals, and superintendent shows that just over half students in Grades 7-12 "attend a school that offers a dedicated computer science class."[3] Further, "black students are less likely than their white peers to have access to such courses, and teachers and parents are more likely to tell boys that they would be good at computer science than they are girls."[4]

In this report, Hanover Research (Hanover) provides an overview of effective computer science models for elementary-level students. Additionally, Hanover provides a series of computer science implementation considerations, as well as several district profiles.

Key Findings

- **The literature is inconclusive as to whether elementary schools should integrate computer science instruction into traditional coursework, or create a stand-alone computer science class.** Some academic journal publications posit that deep, meaningful computer science instruction occurs only when students attend a separate computer science class, with a highly-qualified, computer science-certified instructor. Other journal publications claim that embedding computer science into math, science, or other coursework satisfactorily imparts the computer science discipline to elementary school students.
- **Districts should consider resource availability when deciding how to incorporate computer science into elementary curricula.** Integrating computer science into existing courses requires scheduling accommodations and teacher training. Standalone computer science coursework necessitates hiring additional staff, ensuring appropriate instructional certifications, purchasing additional hardware and software, and reserving appropriate classrooms, among other implementation considerations.
- **School districts with exemplary elementary computer science programs dedicate 50 to 150 minutes of classroom time per week.** Loudoun County Public Schools (VA) recently launched a computer science initiative at three elementary schools, in which students participate in 30 minutes per day of dedicated computer science instruction plus integration of computer science into core subject areas. Charles County Public Schools (MD) has been recognized for its K-12 computer science program, in which elementary school students participate in 50 minutes of computer class per week.
- **District leaders should be intentional to ensure that computer science instruction is inclusive of traditionally underrepresented students.** Women and racial minorities are underrepresented in computer science fields – but survey data suggests these students have equal interest in computer science when compared their Caucasian, male peers. Incorporating computer science into the curriculum for all students equalizes students' exposure to the discipline.

Relevant Research

Education policy researchers, economists, and other workforce and labor experts collectively assert that computer science skills are a critical part of 21st century learning.[5] Specifically, "computer science develops problem solvers and innovators" by teaching computational thinking across the following domains:[6]

Logical data analysis and organization
The ability to create concise problem statements
The ability to identify and implement algorithm-based solutions
The ability to generalize and extrapolate solutions that can be applied to other problems

Source: Forbes[7]

Relatedly, **experts from Stanford University's Computer Science department explicitly advocate for the adoption of computer science curricula in elementary school.** The authors claim that the "topics and skills examined by [elementary] students when they are actively engaged in a computer science curriculum" include:

Stepwise Refinement

- Significant computer programs are not written in one step. Instead, computer science is typified by an iterated process of making small, stepwise changes and then ironing out errors and inefficiencies. This process generally is not covered in primary school curriculum until much later, since most of the problems approached by primary school students can be solved in a single step. For this reason, the process of "debugging" actually can be a valuable addition to the curriculum.

Curriculum Support

- Programming projects in general can be tailored to suit a particular curricular need. For instance, after an in-class experiment with computerized equipment collecting data, computers could write code to process or display data. For a probability unit, students could use random number generators to test their hypotheses about various probabilities.

Algorithmic Thought Process

- Computer science is one of few fields that formalizes the concept of an "algorithm." Once students understand the detail and number of steps necessary to describe an algorithm to a computer, they are more likely to understand and accept algorithms for other processes. For instance, while teachers who have not covered algorithms in any significant way may consider algorithms like the long division algorithm to be "stifling creativity", when taught as part of a larger "algorithms" curriculum long division can be studied as an example of a clever, efficient method for solving a problem that is otherwise intractible.

Problem-Solving Practice

- While usual problem-solving exercises require students to propose solutions which are then verified or disproved by a teacher, programming exercises provide interactive environments in which students have to figure out the problems with their solutions independently. This way, students get more of a response than "yes" or "no" and can receive more immediate feedback than a teacher addressing a class of 20 to 30 students can supply.

Source: Stanford University Computer Science Department[8]

Gaps across Student Subpopulations

However, although economists and education policymakers alike frame computer science as a critical discipline for student success, not all student subpopulations see equal gains in outcomes. **Educators, researchers, and policymakers have consistently noted gaps in computer science access and services across student subpopulations.** Code.org – the preeminent computer science research, advocacy, and technical assistance organization for K-12 schools - shares that the "fields of software, computing, and computer science are plagued by tremendous underrepresentation of women, African Americans, and Hispanics."[9]

Google's education research division cites several structural barriers for this persistent gap in computer science exposure, enrollment, and professional attainment across student subpopulations. These structural barriers include:[10]

- Black students are less likely than White students to have classes dedicated to computer science at the school they attend (47% versus 58%, respectively).
- Black (58%) and Hispanic (50%) students are less likely than White students (68%) to use a computer at home most days of the week.
- Teachers are more likely than parents to say a lack of exposure is a major reason why women and racial and ethnic minorities are underrepresented in computer science fields.

In response to these structural barriers, Google's education research division drafted a series of policy solutions to better engage and attract minority students to computer science within the K-12 public school ecosystem. Key findings and recommendations from the Google research team are further detailed in the figure below. Districts may incorporate this checklist to ensure inclusive computer science practices and programmatic supports for all students.

✓	Black students are less likely than White students to say their school offers a dedicated computer science class. Schools may consider offering a dedicated computer science class to equalize exposure across all student groups, as most students who have learned computer science did so in a class in school.
✓	In general, when students have access to computer science learning in school, they are more likely to say they are very interested in learning it, suggesting that exposure to these opportunities is key to piquing students' interest in the first place. Educators should think about ways to integrate computer science into schools outside of dedicated computer science classes to appeal to more students.
✓	Males are more likely than female students to be aware of groups or clubs at their schools where computer science can be learned. One possibility for this difference is that these opportunities are geared towards activities more likely to attract boys, such as gaming, and the material itself might not resonate as much with some girls. Material should be adapted to appeal to both male and female students.
✓	Educating parents about community-based computer science learning opportunities is a crucial step in supporting and encouraging computer science learning among children.
✓	While there are no differences between male and female students when it comes to computer usage, female students are more likely to use a cellphone or tablet every day. Opportunities to learn computer science through mobile technology (for example, programs that show students how to make their own mobile app) could help build girls' interest in computer science.
✓	Schools could introduce computer science elements into a broader range of online activities – not just "gaming" for boys, but other activities oriented towards all students.
✓	Students' exposure to adults who work with computers and technology can influence their attitudes towards computer science. Schools can employ adults that represent the student body to model working with computers or technology, as students who say they often see people like themselves doing computer science are more likely to say they are 'very interested' in learning computer science.
✓	Female students are less likely than male students to say they are 'very skilled' in math or science – parents and teachers may unconsciously push their daughters away from computer science. Intentional efforts must be made to encourage girls and minorities into computer science.

Source: Google Education Research Division[11]

Curriculum

Although education researchers generally agree that computer science develops critical thinking skills aligned with 21st century learning goals, **experts cannot conclusively determine how to best implement a computer science curriculum in elementary school classrooms.**

Authors from the St. Mary's College educational studies journal *A Rising TIDE* shared in 2015 that "there has been very little research that approached teaching computer science in the early elementary years and even fewer studies that actually integrate computer programming into the curriculum… most programming languages written for students, such as Alice and Scratch, are not intended for young children… making them not developmentally appropriate for students in Grades K-2."[12]

However, even with this limitation of empirical research, experts from the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative collectively developed a series of implementation recommendations for computer science instruction. Specifically, the framework is a product of collaboration across "states, districts, and the computer science education community to develop conceptual guidelines for computer science education."[13]

The authors outline what computer science skills and abilities students should master, segmented by grade level. The "developmentally appropriate" framework is sequential, so that each year's instruction builds upon previous years' computer science work. The figure below outlines two of the suggested concepts and skills for Grades K-2 computer science instruction. The comprehensive computer science pedagogical framework (for students in Grades K-2 and Grades 3-5) may be accessed by clicking here.

Computer Science Concepts for Grades K-2 Students

Devices

- *People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally. Computing devices can be used to do a number of things, such as play music, create documents, and send pictures. Computing devices are also very precise.*
- *For example, computers can perform multiple complex calculations much faster and with greater accuracy than people. While people may diverge from instructions given to them, computers will follow instructions exactly as they are given, even if they do not achieve the intended result.*

Hardware and Software

- *A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand.*
- *Examples of hardware include screens to display information and buttons, keys, or dials to enter information. Software applications are programs with specific purposes, such as a web browser or game. A person may use a mouse (hardware) to click on a button displayed in a web browser (software) to navigate to a new web page. Computing systems convert instructions, such as "print," "save," or "crop," into a special language that the computer can understand.*
- *At this level, understanding that computer information is encoded is appropriate, but the explicit understanding of "bits" is reserved for later grade levels.*

Source: K12CS.org[14]

Scheduling and Logistics

The Outlier Research and Evaluation Project – an NSF-funded study conducted by the University of Chicago's STEM Education division – outlines 11 steps for school leaders to bring computer science to their school. These recommendations are based on 25 interviews with school leaders across three school districts. These recommendations are further detailed in the figure on the following page, and can be adapted to elementary, middle, or high school contexts.[15]

1. Commit to bringing computer science to your school
2. Assess knowledge of and interest in computer science for stakeholders within and outside school walls
3. Visit schools with computer science programs, sit in on classes, and talk to teachers and school leaders
4. Communicate the importance of computer science to stakeholders within and outside your school
5. Determine the right size and scope for your own computer science program
6. Identify desired student learning outcomes and create an assessment plan
7. Review available instructional materials and identify equipment needs
8. Prepare a realistic budget and allocate resources
9. Consider who will teach your computer science classes and what credentials they need
10. Identify a range of support for the teachers who will teach computer science
11. Be flexible and patient

Source: University of Chicago STEM Outlier Research and Evaluation Project[16]

Relatedly, the University of Chicago STEM Outlier Research Project also surveyed 101 computer science teachers across five U.S. cities to identify the "most important supports for teaching computer science."[17] Based on their responses, **the University of Chicago researchers posit that the most important supports for teaching computer science include**:[18]

- Professional Development
- Other Computer Science Teachers
- Effective Curriculum
- Peer Teachers/Colleagues in General
- Online Teaching Resources for Computer Science

Schools and districts across the country have attempted to integrate computer science into the elementary school curriculum. Specifically, "many elementary and middle schools are working to integrate 'computational thinking' – broadly defined as using the concepts and practices of computer science to solve problems – into existing subjects, such as math and science." Others offer computer science as a separate discipline, apart from reading, writing, or mathematics.

Notably, **the research is inconclusive as to whether elementary schools should infuse computer science concepts and skills into existing coursework, or develop stand-alone computer science classes.** The figure below summarizes several academic journal publications that advocate for and against computer science as a separate discipline at the elementary school level.

+	−
• *Educational Technology and Society* authors suggest "a reconsideration of computer science as a separate subject in both primary and secondary education."[19] • The K-12CS.org framework authors write that "broad and deep exposure" to computer science curriculum should entail "independent special [coursework], similar to Science, Music, Art," in Grades K-5. Integration of computer science into the general classroom equates to only moderate exposure.[20]	• Authors from the journal *Educational Information Technology* write that, "differences remain over whether all [elementary] students should learn the discipline of computer science."[21] • *Learning and Leading with Technology* journal authors write that "Level I [computer science instruction], recommended for Grades K-8…can be best accomplished by adding short modules to existing science, mathematics, and social studies units."[22]

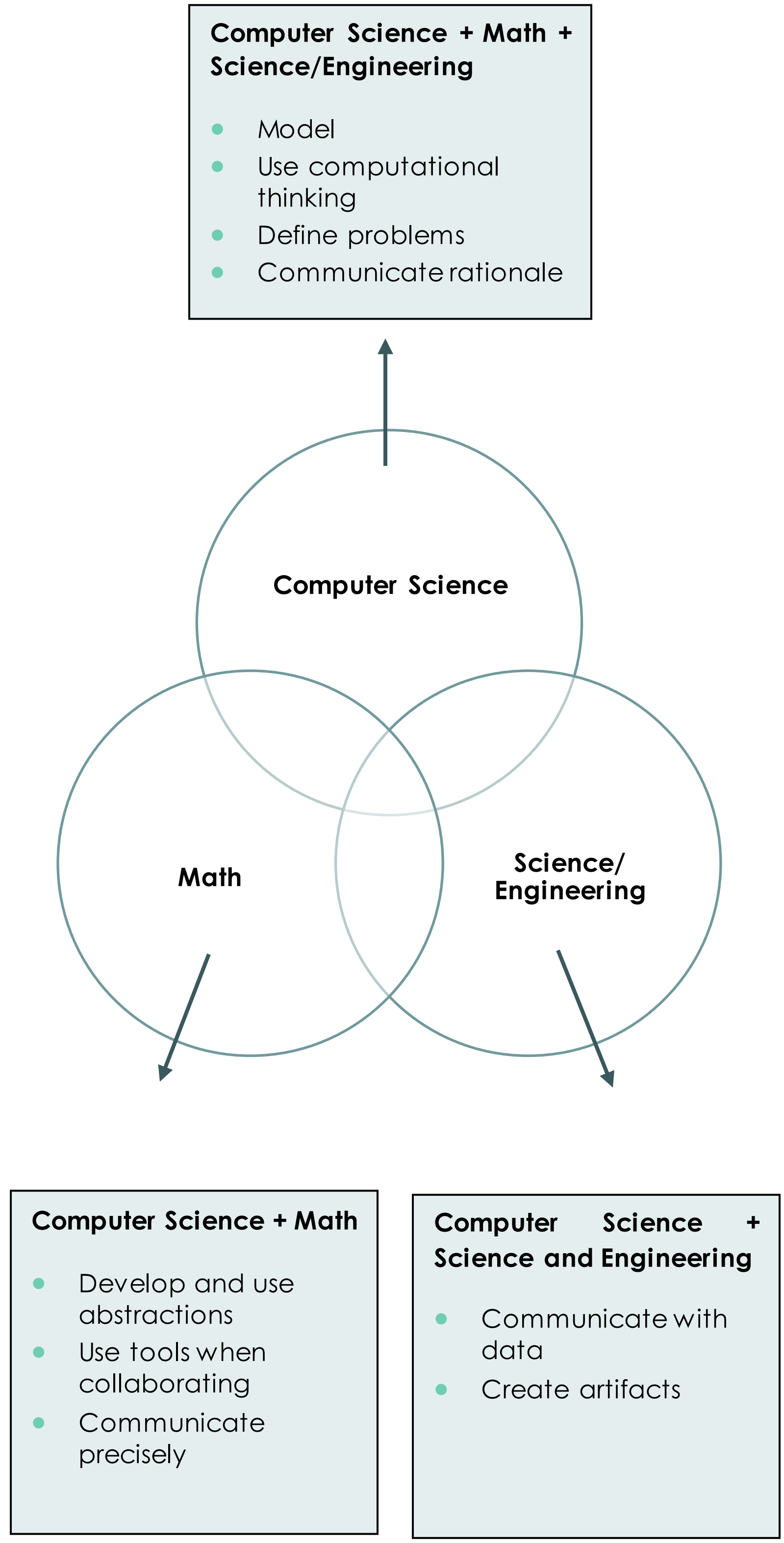
However, although the research is inconclusive as to whether elementary schools should structure computer science as an independent discipline during the school day, **there exist a series of computer science implementation considerations for district leaders.** Several key differences between standalone computer science instruction and the integration of computer science with core coursework are further detailed in the figure on the following page.

Standalone Computer Science	Integrated with Core Subjects
✓ Districts must hire additional staff to teach computer science coursework ✓ Districts must ensure additional staff are sufficiently trained and certified ✓ Districts must secure classroom space for instruction and ensure students have access to necessary hardware and software ✓ Districts must purchase computer science curriculum and/or developmentally appropriate programs ✓ Computer science teachers require professional development and opportunities to network and share best practices	✓ Teachers must explicitly draw connections between core subjects and computer science ✓ Time must be built into lesson plans to cover computer science key concepts – or school day must be extended to allow time for computer science instruction ✓ Core subject teachers must be able to instruct students about "computing software, such as hardware, software and use of devices, networks and the Internet, handling of data and analysis (which is done differently than in science and math), algorithms and coding, and the societal impacts of coding."[23]

Relationship across Disciplines

Computer science instruction heavily overlaps with critical concepts and thinking skills across several disciplines. The K12CS.org authors posit that their computer science instructional framework is "grounded in the belief that computer science offers unique opportunities for developing computational thinking and that the framework's practices can be applied to other subjects beyond computer science."[24]

The figure on the right illustrates the overlap between the computer science, science and engineering, and math disciplines. Notably, "explicit instruction is required to create the connections illustrated in the diagram."[25]



Source: K12CS.org[26]

District Profiles

In the following section, Hanover reviews how school districts structure and implement computer science programming and instruction for elementary school students.

Loudoun County Public Schools

Former Virginia Governor Terry McAuliffe heavily emphasized the role of computer science in securing the state of Virginia's future economic prosperity. The former governor shared, "Virginia has 36,000 unfilled jobs in the computer science sector… The starting pay is $88,000. We either fill these jobs or they go to other states."[27]

As a direct result of the former governor's computer science education policy initiative, Loudoun County Public Schools (LCPS) "launched the state [of Virginia's] very first computer science immersion schools" in the 2017-18 school year.[28] A key component of these schools is that **every student engages in "with no fewer than 30 minutes of Computer Science lessons and activities per instructional day**."[29] As the school year goes on, teachers and instructors will be tasked with "weaving" computer science into "other core subject area instructional time."[30] Example instructional activities include:[31]

- Learning programming languages
- Learning how to apply programming to LEGOs and Minecraft

Critical elements of the hiring, professional development, programmatic, and curricular frameworks are further detailed in the following subsections.

Hiring Initiatives and Professional Development

LCPS partnered with the nonprofit educational organization "Code to the Future" to provide employees with computer science professional development and training. Specific elements of LCPS/ "Code to the Future" professional development include:[32]

- Teachers at the three designated elementary schools received a full day of "Code to the Future" training ahead of the first day of school.
- Teachers at these schools are provided weekly support and coaching from a "Code to the Future" "coach," who provides instructional resources and lessons to help students solve relevant and authentic problems.

Programming and Curriculum

LCPS developed a rigorous framework for elementary school computer science instruction. The critical components of LCPS's comprehensive computer science initiative are further detailed in the figure below. Notably, computer science instruction is sequenced – each year, students build upon skills harnessed the previous academic year.

Beginning in Kindergarten, students at the three immersion schools will be introduced to Scratch, a coding language and program. Already this school year, students as young as five are learning to spell their names with characters and then animate the letters.

Grade 3 students in the program will be taught to play chess blind folded ahead of learning coding, as the organization "Code to the Future's founder considers chess a good foundation to computer science.

By the end of their elementary career, Grade 5 students will have learned the programming language Java and use it to create a game that incorporates characters and dialogue to tell a story.

Source: LoudonNow[33]

Charles County Public Schools

In 2013, Charles County Public Schools (CCPS) – located in Maryland – adopted a district-wide framework to teach every student the computer science discipline. The specific district-wide goals were to:[34]

- Broaden participation in computer science through multiple, high-quality experiences
- Increase the number and diversity of students choosing to study advanced STEM programs
- Establish an exemplary systemic model for teaching and learning of computer science.

The Superintendent Kimberly A. Hill, Ed.D., shares that, "Every student in CCPS will be exposed to the skills that computer science has to offer… instead of being intimidated by computer science, CCPS students will learn about how much fun computer science can be."[35] Notably, Code.org shared that "CCPS is a model district and a national leader in the promotion of computer science for students."[36]

CCPS school and district leadership adopted a series of policies and initiatives to ensure all students are exposed to computer science. These policies and district-wide actions are further detailed in the subsections in the following columns.

Hiring Initiatives and Professional Development

CCPS leadership recognizes the importance of placing highly qualified, well-trained instructors in computer science classrooms. To this end, CCPS recently partnered with the National Science Foundation (NSF) to provide professional development training and resources to CCPS instructors. This NSF initiative, titled "CS for All," is intended to provide all students with "rigorous" computer science instruction – with special emphasis on "women, girls minorities, and persons with disabilities."[37] Specific NSF-supported programs used by CCPS at the elementary school level are further detailed in the figure below.

ScratchEd	ScratchEd is an important partner in exploring ways to use Scratch as an open-ended tool for storytelling, mathematical reasoning, and the creative design process. CCPS held its first Southern Maryland Scratch "Meetup" in March 2016 with great success, taking another step toward building a teacher community and support system.
Investigating Conceptual Foundations for a Transdisciplinary Model Integrating Computer Science into the Elementary STEM Curriculum	This project offered teachers advice on which projects in their problem-based learning integrated computer science and STEM modules are appropriate for specific grade levels.
Learning Trajectories for Integrating K-5 Computer Science and Mathematics	This project provided integrated mathematics and computational thinking lessons and activities. CCPS is working to develop ways to support implementation of these resources with a focus on broader participation.
Developing Elementary Learning Progressions to Integrate Computational Thinking	This project investigates how computational thinking skills can be developed over time while supporting education in the existing curriculum. CCPS has reviewed and is planning to pilot some digital storytelling materials at the upper elementary level.

Source: National Science Foundation[38]

Relatedly, CCPS partnered with Code.org – a nonprofit organization backed by companies like Google, Amazon, Dropbox, Microsoft, and others – to provide supplemental professional development training for CCPS instructors. Code.org provided training to more than 75 teachers on how to integrate computer coding into lessons about English, mathematics, and science, as well as other content areas. Code.org also offers additional training to CCPS teachers throughout the school year. [39]

Programming and Curriculum

CCPS prioritizes computer science instruction across all grade levels. In elementary school, CCPS has adopted a range of curriculum, programs, and initiatives to ensure all students receive rigorous computer science instruction.

Specifically, the school facilitates **50 minutes of computer class, once per week, for elementary students**. Relatedly, "computer science and computational thinking are integrated across all subject areas, using Bee Bots, robotics, and Code.org activities, amongst others."[40] CCPS also facilitates before and after-school computer coding and robotics clubs, for students across all grade levels.[41]

Project Evaluation Form

Hanover Research is committed to providing a work product that meets or exceeds client expectations. In keeping with that goal, we would like to hear your opinions regarding our reports. Feedback is critically important and serves as the strongest mechanism by which we tailor our research to your organization. When you have had a chance to evaluate this report, please take a moment to fill out the following questionnaire.

http://www.hanoverresearch.com/evaluation/index.php

Caveat

The publisher and authors have used their best efforts in preparing this brief. The publisher and authors make no representations or warranties with respect to the accuracy or completeness of the contents of this brief and specifically disclaim any implied warranties of fitness for a particular purpose. There are no warranties that extend beyond the descriptions contained in this paragraph. No warranty may be created or extended by representatives of Hanover Research or its marketing materials. The accuracy and completeness of the information provided herein and the opinions stated herein are not guaranteed or warranted to produce any particular results, and the advice and strategies contained herein may not be suitable for every client. Neither the publisher nor the authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages. Moreover, Hanover Research is not engaged in rendering legal, accounting, or other professional services. Clients requiring such services are advised to consult an appropriate professional.[42]

Endnotes

1 DeRuy, E. "A Plan To Teach Every Child Computer Science." The Atlantic, October 19, 2016. https://www.theatlantic.com/education/archive/2016/10/a-plan-to-teach-every-child-computer-science/504587/

2 Keilman, J. "Coding Education Rare in K-12 Schools but Starting to Catch On." Chicago Tribune, January 2, 2016. http://www.chicagotribune.com/news/ct-coding-high-school-met-20160101-story.html

3 Wang, J. "Racial and Gender Gaps in Computer Science Learning: New Google-Gallup Research." Google, October 18, 2016. https://blog.google/outreach-initiatives/education/racial-and-gender-gaps-computer-science-learning-new-google-gallup-research/

4 DeRuy, Op. cit.

5 Yongpradit, P. "Computer Science: A Playground for Twenty-First Century Skills." Partnership for Twenty-First Century Learning, February 7, 2017. http://www.p21.org/news-events/p21blog/2128-computer-science-a-playground-for-21st-century-skills

6 Miller, A.D. "Why We Must Have Computer Science In More Schools And Classrooms." Forbes, November 14, 2014. https://www.forbes.com/sites/oracle/2014/11/14/why-we-must-have-computer-science-in-more-schools-and-classrooms/

7 Bulleted text quote verbatim, with minor adaptions, from: Ibid.

8 Figure contents taken verbatim, with minor adaptions, from: Solomon, J. and P. Rusev. "Early Acquisition of Computer Science." Stanford University Computer Science Department, 2008. https://cs.stanford.edu/people/eroberts/cs201/projects/2007-08/early-acquisition-of-cs/advantages.html

9 "Code.Org and Diversity in Computer Science." Code.Org. https://code.org/diversity

10 Bulleted text quoted verbatim from: "Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks, and Hispanics." Google Inc. & Gallup Inc., 2016. http://services.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf

11 Figure adapted from: Ibid.

12 Kind, A. "Computing Attitudes: Will Teaching Second Grade Students Computer Science Improve Their Self-Efficacy and Attitude and Eliminate Gender Gaps?" *A Rising TIDE*, 8, 2015. p. 2. http://www.smcm.edu/mat/wp-content/uploads/sites/73/2015/06/Amanda-Kind-2015.pdf

13 "K–12 Computer Science Framework." K12CS.Org. http://k12cs.org

[14] Figure contents taken verbatim, with minor adaptions, from: Alano, J. et al. "K-12 Computer Science Framework." K12CS.Org, 2016. pp. 91–98. https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf
[15] "Advice from School Leaders to School Leaders: Introducing Computer Science in Your School." Outlier Research and Evaluation, University of Chicago STEM Education. https://s3.amazonaws.com/cemse/basics/files/findings/Advice-fromSchoolLeaders.pdf
[16] Figure contents taken verbatim, with minor adaptions, from: Ibid.
[17] "Barriers and Supports to Implementing Computer Science." Outlier Research and Evaluation, University of Chicago STEM Education. https://s3.amazonaws.com/cemse/basics/files/findings/supports-teaching-cs/Five_Ways_Support_CS_Teachers.pdf
[18] Figure contents taken verbatim, with minor adaptions, from: Ibid.
[19] Fluck, A. et al. "Arguing for Computer Science in the School Curriculum." *Educational Technology and Society*, 19:3, 2016. p. 38.
[20] Alano et al., Op. cit., p. 165.
[21] Webb, M. et al. "Computer Science in K-12 School Curricula of the Twenty-First Century: Why, What, and When?" *Educational Information Technology*, 22, 2017. p. 465.
[22] Tucker, A.B. "A New K-12 Computer Science Curriculum." *Learning and Leading with Technology*, 31:7, 2004. p. 17. https://files.eric.ed.gov/fulltext/EJ695762.pdf
[23] Kirby, A. "Computer Science Dilemma: A Stand-Alone or Part of STEM?" K-12 Daily, January 24, 2017. https://k-12daily.org/curriculum-instruction/computer-science-dilemma-a-stand-alone-or-part-of-stem
[24] "Computational Thinking." K12cs.Org. https://k12cs.org/computational-thinking/
[25] Ibid.
[26] Figure contents taken verbatim from: Ibid.
[27] Nadler, D. "Loudoun Launches Virginia's First Computer Science Immersion Schools." Loudoun Now, September 14, 2017. http://loudounnow.com/2017/09/14/loudoun-launches-virginias-first-computer-science-immersion-schools/
[28] Ibid.
[29] "Code to the Future." Loudoun County Public Schools, 2018. https://www.lcps.org/codetothefuture
[30] Ibid.
[31] Bulleted text quoted verbatim from: Ibid.
[32] Bulleted text taken verbatim from: [1] Nadler, Op. cit. [2] "Code to the Future," Op. cit.
[33] Figure contents taken verbatim, with minor adaptions, from: Nadler, Op. cit.
[34] Bulleted text quoted verbatim from: "Maryland School District Showcases Computer Science Education at All Levels." National Science Foundation, September 15, 2016. https://www.nsf.gov/discoveries/disc_summ.jsp?cntn_id=189772&org=NSF&from=news
[35] "Code.Org." Charles County Public Schools, 2015. https://www.ccboe.com/code.php
[36] Ibid.
[37] "Computer Science Is for All Students!" National Science Foundation. https://www.nsf.gov/news/special_reports/csed/csforall.jsp
[38] Figure contents taken verbatim, with minor adaption, from: "Maryland School District Showcases Computer Science Education at All Levels," Op. cit.
[39] Bulleted text quote verbatim, with minor adaptions, from: "Code.Org," Op. cit.
[40] Anfenson-Comeau, J. "Korean Educators Visit Charles County Public Schools." Maryland Independent, July 26, 2017. http://www.somdnews.com/independent/spotlight/korean-educators-visit-charles-county-public-schools/article_3b56e055-aa66-5c0d-9378-7eb92ada107b.html
[41] Ibid.