B | Center for Universal Education at BROOKINGS

BUILDING SKILLS FOR LIFE

How to expand and improve computer science education around the world

Emiliana Vegas, Michael Hansen, and Brian Fowler

BROOKINGS

About the authors

Emiliana Vegas
is a senior fellow and co-director at the Center for Universal Education at Brookings

Michael Hansen
is a senior fellow at the Brown Center on Education Policy at Brookings

Brian Fowler
is a former research analyst at the Center for Universal Education at Brookings

Acknowledgments
The authors thank Kareen Fares, Benson Neethipudi, Yuri Nesen, and Nicolás Zerbino for their contributions to the background research for this report. We are also grateful to Tobias V. Abelmann, Shuchi Grover, Mark Guzdial, Niel McLean, Sue Sentance, and Pat Yongpradit for helpful guidance throughout the research and in earlier drafts and to Gabriel Sanchez Zinny, Jake Baskin, David Miyashiro, and Richard Culatta for providing guidance.

The Brookings Institution is a nonprofit organization devoted to independent research and policy solutions. Its mission is to conduct high-quality, independent research and, based on that research, to provide innovative, practical recommendations for policymakers and the public. The conclusions and recommendations of any Brookings publication are solely those of its author(s), and do not reflect the views of the Institution, its management, or its other scholars.

Brookings gratefully acknowledges the support provided by Amazon, Atlassian Foundation International, Google, and Microsoft.

Brookings recognizes that the value it provides is in its commitment to quality, independence, and impact. Activities supported by its donors reflect this commitment.

Contents

Introduction

Access to education has expanded around the world since the late 1990s through the combined efforts of governments, bilateral and multilateral agencies, donors, civil society, and the private sector, yet education quality has not kept pace. Even before the COVID-19 pandemic led to school closures around the world, all young people were not developing the broad suite of skills they need to thrive in work, life, and citizenship (Filmer, Langthaler, Stehrer, & Vogel, 2018).

The impact of the pandemic on education investment, student learning, and longer-term economic outcomes threatens not only to dial back progress to date in addressing this learning crisis in skills development but also to further widen learning gaps within and between countries. Beyond the immediate and disparate impacts of COVID-19 on students' access to quality learning, the global economic crisis it has precipitated will shrink government budgets, potentially resulting in lower education investment and impacting the ability to provide quality education (Vegas, 2020). There is also a concern that as governments struggle to reopen schools and/or provide sufficient distance-learning opportunities, many education systems will focus on foundational skills, such as literacy and numeracy, neglecting a broader set of skills needed to thrive in a rapidly changing, technologically-advanced world.

Among these broader skills, knowledge of computer science (CS) is increasingly relevant. CS is defined as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their [implementation], and their impact on society" (Tucker, 2003).[1] CS skills enable individuals to understand how technology works, and how best to harness its potential to improve lives. The goal of CS education is to develop computational thinking skills, which refer to the "thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer" (K-12 Computer Science Framework Steering Committee, 2016). CS education is also distinct from computer or digital literacy, in that it is more

[1] Denning et al. (1989) defined the discipline of computing as "the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application."

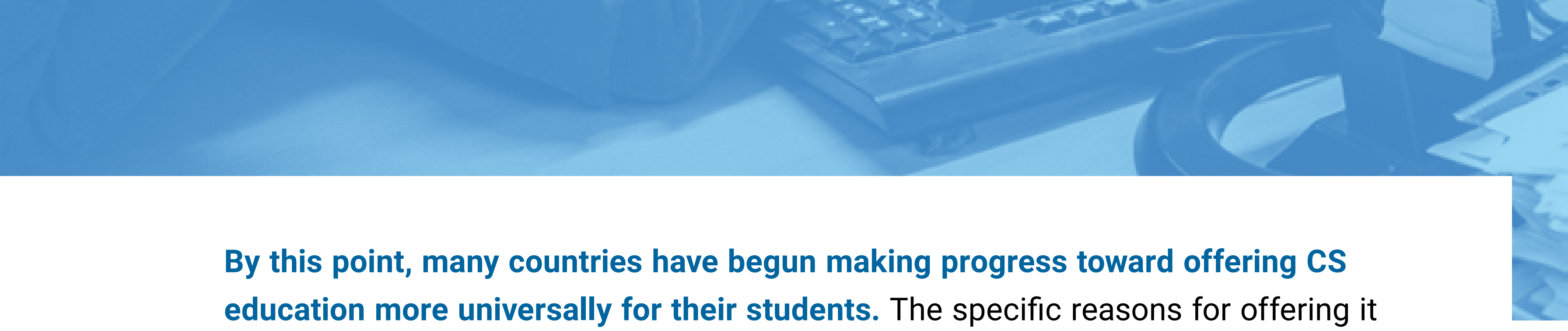
concerned with computer *design* than with computer use. For example, coding is a skill one would learn in a CS course, while creating a document or slideshow presentation using an existing program is a skill one would learn in a computer or digital literacy course.

Research has shown that students benefit from CS education by increasing college enrollment rates and developing problem-solving abilities (Brown & Brown, 2020; Salehi et al., 2020). Research has also shown that lessons in computational thinking improve student response inhibition, planning, and coding skills (Arfé et al., 2020). Importantly, CS skills pay off in the labor market through higher likelihood of employment and better wages (Hanson & Slaughter, 2016; Nager & Atkinson, 2016). As these skills take preeminence in the rapidly changing 21st century, CS education promises to significantly enhance student preparedness for the future of work and active citizenship.

The benefits of CS education extend beyond economic motivations. Given the increasing integration of technology into many aspects of daily life in the 21st century, a functional knowledge of how computers work—beyond the simple use of applications—will help all students.

In this report, we first discuss in section one the evidence that argues for the expansion of CS education in primary and secondary schools across the world. Section two analyzes global efforts to expand CS education and four key challenges to its expansion: (1) providing educators and students with adequate access to devices and connectivity; (2) ensuring qualified teachers through teacher preparation and professional development; (3) fostering student engagement and interest in CS education; and (4) developing and using evidence on curriculum and core competencies, effective instructional methods, and assessment of CS skills. Section three synthesizes the lessons learned from regional and national governments at various levels of economic development and progress in expanding CS education and in multiple regions of the world. In Section four, we distill the lessons learned into recommendations for other governments looking to implement or expand CS education.

1. Why expand CS education?

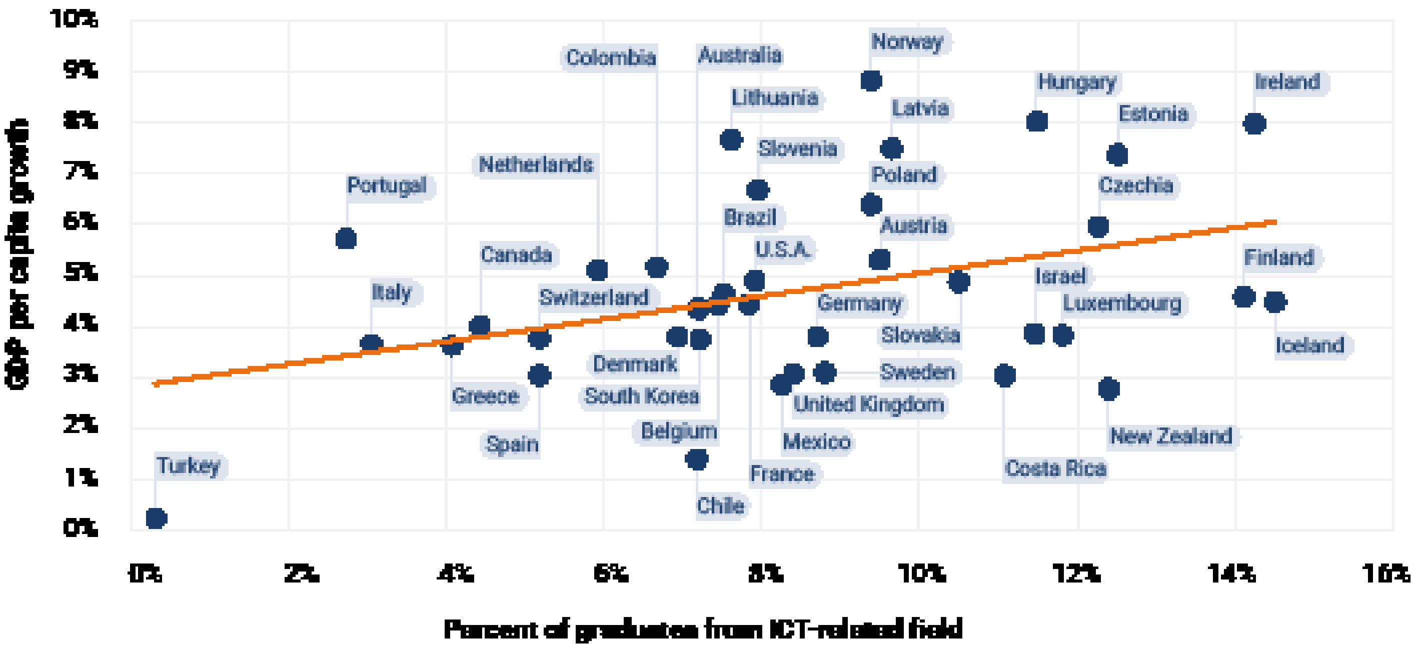


By this point, many countries have begun making progress toward offering CS education more universally for their students. The specific reasons for offering it will be as varied as the countries themselves, though economic arguments often top the list of motivations. Other considerations beyond economics, however, are also relevant, and we account for the most common of these varied motives here.

The economic argument

At the macroeconomic level, previous research has suggested that countries with more workers with ICT (information, communications, and technology) skills will have higher economic growth through increases in productivity (Maryska, Doucek, & Kunstova, 2012; Jorgenson & Vu, 2016). Recent global data indicate that there is a positive relationship between the share of a country's workforce with ICT skills and its economic growth. For example, using data from the Organisation for Economic Cooperation and Development (OECD), we find that countries with a higher share of graduates from an ICT field tend to have higher rates of per capita GDP (Figure 1). The strength of the estimated relationship here is noteworthy: A one percentage point increase in the share of ICT graduates correlates with nearly a quarter percentage point increase in recent economic growth, though we cannot determine the causal nature of this relationship (if any). Nonetheless, this figure supports the common view that economic growth follows from greater levels of investment in technological education.

Figure 1. Countries with more graduates from ICT fields tend to have more economic growth

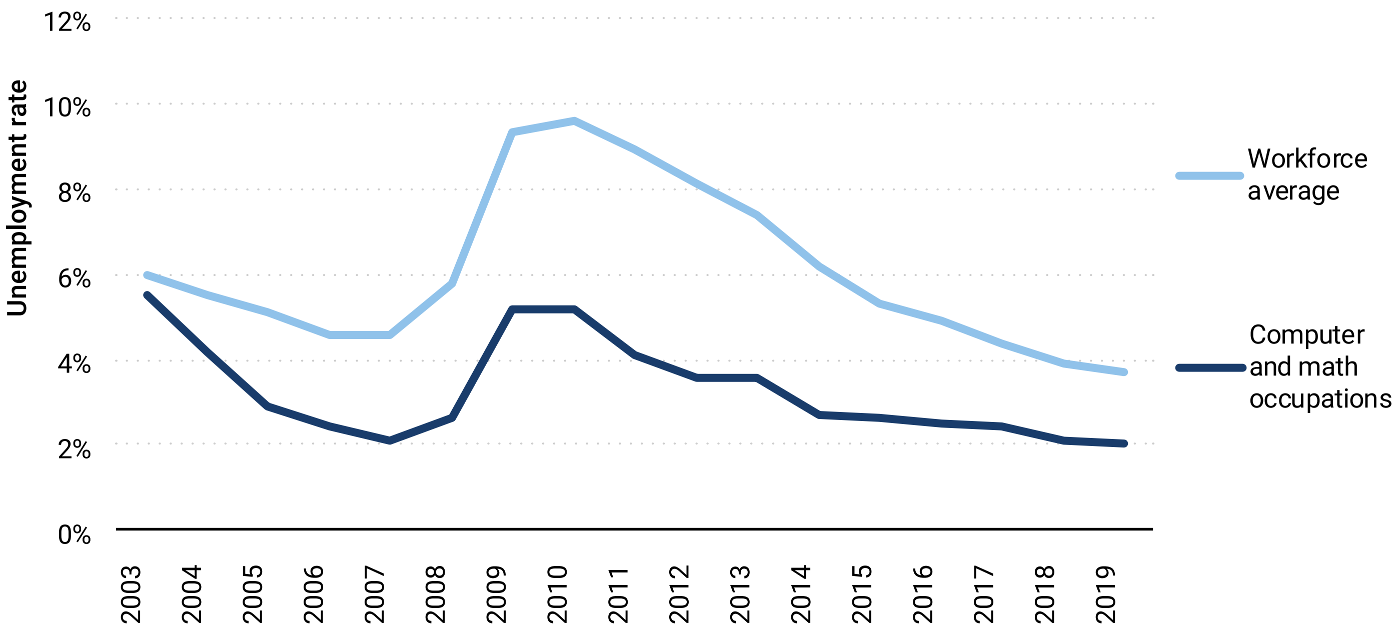


Note: Data comes from 2018 and consists of only males with a bachelor's degree or similar. Thirty-nine countries are included, and Japan was excluded because of missing values. The GDP per capita is adjusted by PPP.
Source: OECD (2021), GDP and distribution of graduates by field.

At the microeconomic level, CS skills pay off for individuals—both for those who later choose to specialize in CS and those who do not. Focusing first on the majority of students who pursue careers outside of CS, foundational training in CS is still beneficial. Technology is becoming more heavily integrated across many industrial endeavors and academic disciplines—not just those typically included under the umbrella of science, technology, engineering, and mathematics (STEM) occupations. Careers from law to manufacturing to retail to health sciences all use computing and data more intensively now than in decades past (Lemieux, 2014). For example, using data from Germany, researchers showed that higher education programs in CS compared favorably against many other fields of study, producing a relatively high return on investment for lower risk (Glocker and Storck, 2014). Notably, completing advanced training in CS is not necessary to attain these benefits; rather, even short introductions to foundational skills in CS can increase young students' executive functions (Arfe et al., 2020). Further, those with CS training develop better problem-solving abilities compared to those with more general training in math and sciences, suggesting that CS education offers unique skills not readily developed in other more common subjects (Salehi et al., 2020).

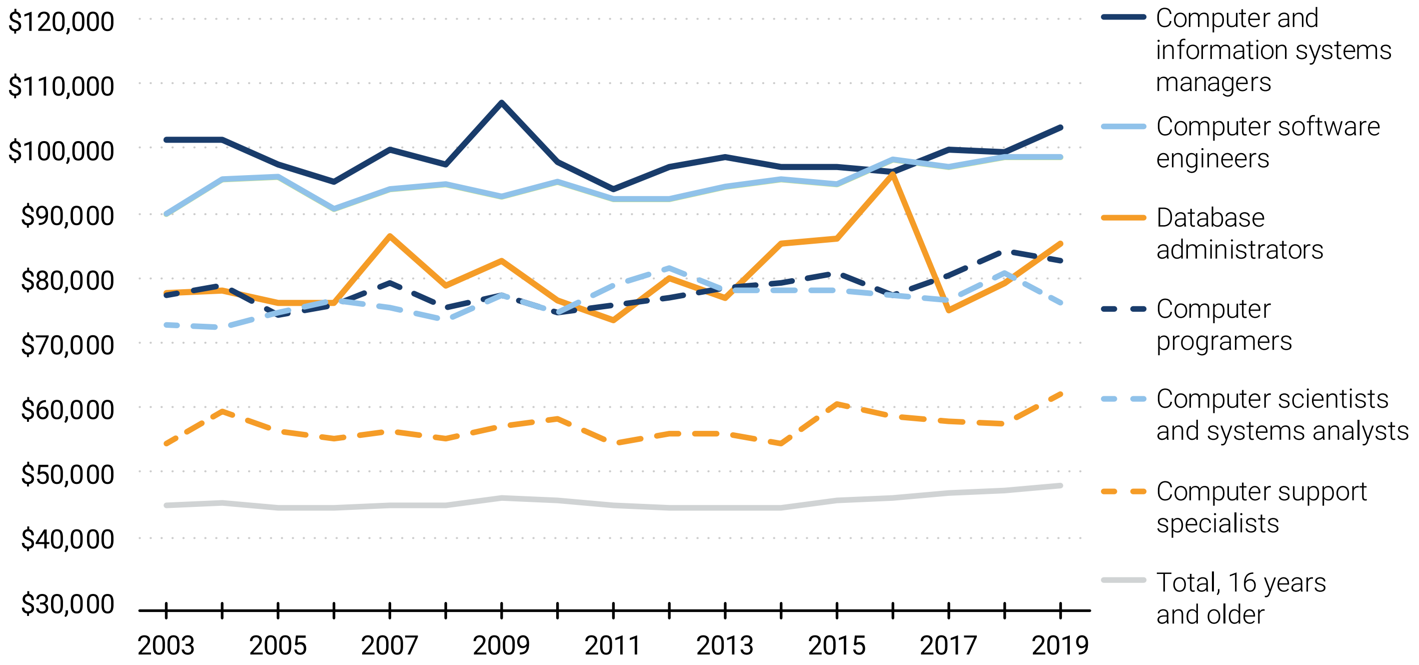
For those who choose to pursue advanced CS studies, specializing in CS pays off both in employment opportunities and earnings. For example, data from the U.S. show workers with CS skills are less likely to be unemployed than workers in other occupations (Figure 2). Moreover, the average earnings for workers with CS skills are higher than for workers in other occupations (Figure 3). These results are consistent across multiple studies using U.S. data (Carnevale et al., 2013; Altonji et al., 2012) and international data (Belfield et al., 2019; Hastings et al., 2013; Kirkeboen et al., 2016). Further, the U.S. Bureau of Labor Statistics has projected that the market for CS professionals will continue to grow at twice the speed of the rest of the labor market between 2014 and 2024 (National Academies of Sciences, 2018).

Figure 2. U.S. workers in computer and math occupations have lower unemployment rates than the rest of the workforce



Source: Bureau of Labor Statistics, Current Population Survey (Unemployed persons by occupation and sex).

Figure 3. U.S. workers with computer science skills have higher average earnings than workers in other occupations



Note: Weekly U.S. earnings data from 2019 were converted to annual data assuming 52 weeks in a year.
Source: Bureau of Labor Statistics, Current Population Survey (Median weekly earnings of full-time wage and salary workers by detailed occupation).

A common, though inaccurate, perception about the CS field is that anybody with a passion for technology can succeed without formal training. There is a nugget of truth in this view, as many leaders of major technology companies including Bill Gates, Elon Musk, Mark Zuckerberg, and many others have famously risen to the top of the field despite not having bachelor's degrees in CS. Yet, it is a fallacy to assume that these outliers are representative of most who are successful in the field. This misconception could lead observers to conclude that investments in universal CS education are, at best, ineffective: providing skills to people who would learn them on their own regardless, and spending resources on developing skills in people who will not use them. However, such conclusions are not supported by empirical evidence. Rather, across STEM disciplines, including CS, higher levels of training and educational attainment lead to stronger employment outcomes, on average, than those with lesser levels of training in the same fields (Altonji et al., 2016; Altonji and Zhong, 2021).

The inequality argument

Technology—and particularly unequal access to its benefits—has been a key driver of social and economic inequality within countries. That is, those with elite social status or higher wealth have historically gotten access to technology first for their private advantages, which tends to reinforce preexisting social and economic inequalities. Conversely, providing universal access to CS education and computing technologies can enable those with lower access to technological resources the opportunity to catch up and, consequently, mitigate these inequalities. Empirical studies have shown how technological skills or occupations, in particular, have reduced inequalities between groups or accelerated the assimilation of immigrants (Hanson and Slaughter, 2017; DeVol, 2016).

Technology and CS education are likewise frequently considered critical in narrowing income gaps between developed and developing countries. This argument can be particularly compelling for low-income countries, as global development gaps will only be expected to widen if low-income countries' investments in these domains falter while high-income countries continue to move ahead. Rather, strategic and intensive technological investment is frequently seen as a key strategy for less-developed countries to leapfrog stages of economic development to quickly catch up to more advanced countries (Fong, 2009; Lee, 2019).

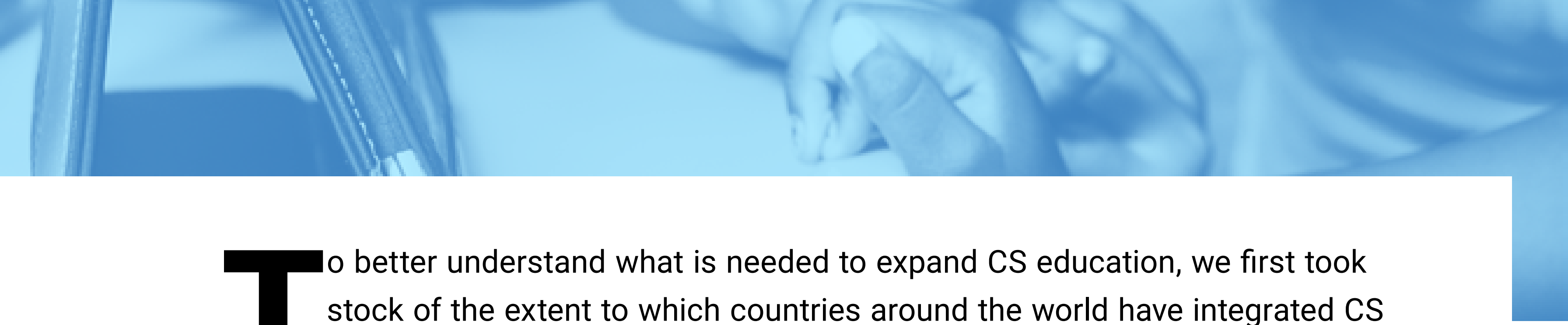
CS skills enable adaptation in a quickly changing world, and adaptability is critical to progress in society and the economy. Perhaps there is no better illustration of the ability to thrive and adapt than from the COVID-19 pandemic. The pandemic has forced closures of many public spaces across the globe, though those closures' impacts have been disproportionately felt across workers and sectors. Workers with the skills and abilities to move their job functions online have generally endured the pandemic more comfortably than those without those skills. And even more importantly, the organizations and private companies that had the human capacity to identify how technology could be utilized and applied to their operations could adapt in the face of the pandemic, while those without the resources to pivot their operations have frequently been forced to close in the wake of pandemic-induced restrictions. Thus, the pandemic bestowed comparative benefits on those with access to technology, the skills to use it, and the vision to recognize and implement novel applications quickly, while often punishing those with the least access and resources (OECD, 2021).

Failing to invest in technology and CS education may result in constrained global competitiveness, leaving governments less able to support its citizens. We recognize that efforts to expand CS education will demand time and money of public officials and school leaders, often in the face of other worthy competing demands. Though the contemporary costs may even seem prohibitive in some cases, the costs of inaction (while less immediately visible) are also real and meaningful in most contexts.

Beyond economics

We expect the benefits of CS education to extend beyond economic motivations, as well. Many household activities that were previously performed in real life are now often performed digitally, ranging from banking, shopping, travel planning, and socializing. A functional knowledge of how computers work—beyond the simple use of applications—should benefit all students as they mature into adults given the increasing integration of technology into many aspects of daily life in the 21st century. For example, whether a person wants to find a job or a romantic partner, these activities frequently occur through the use of technology, and understanding how matching algorithms work make for more sophisticated technology users in the future. Familiarity with CS basic principles can provide users more flexibility in the face of constant innovation and make them less vulnerable to digital security threats or predators (Livingstone et al., 2011). Many school systems now provide lessons in online safety for children, and those lessons will presumably be more effective if children have a foundational understanding of how the internet works.

2. Global advances in expanding CS education

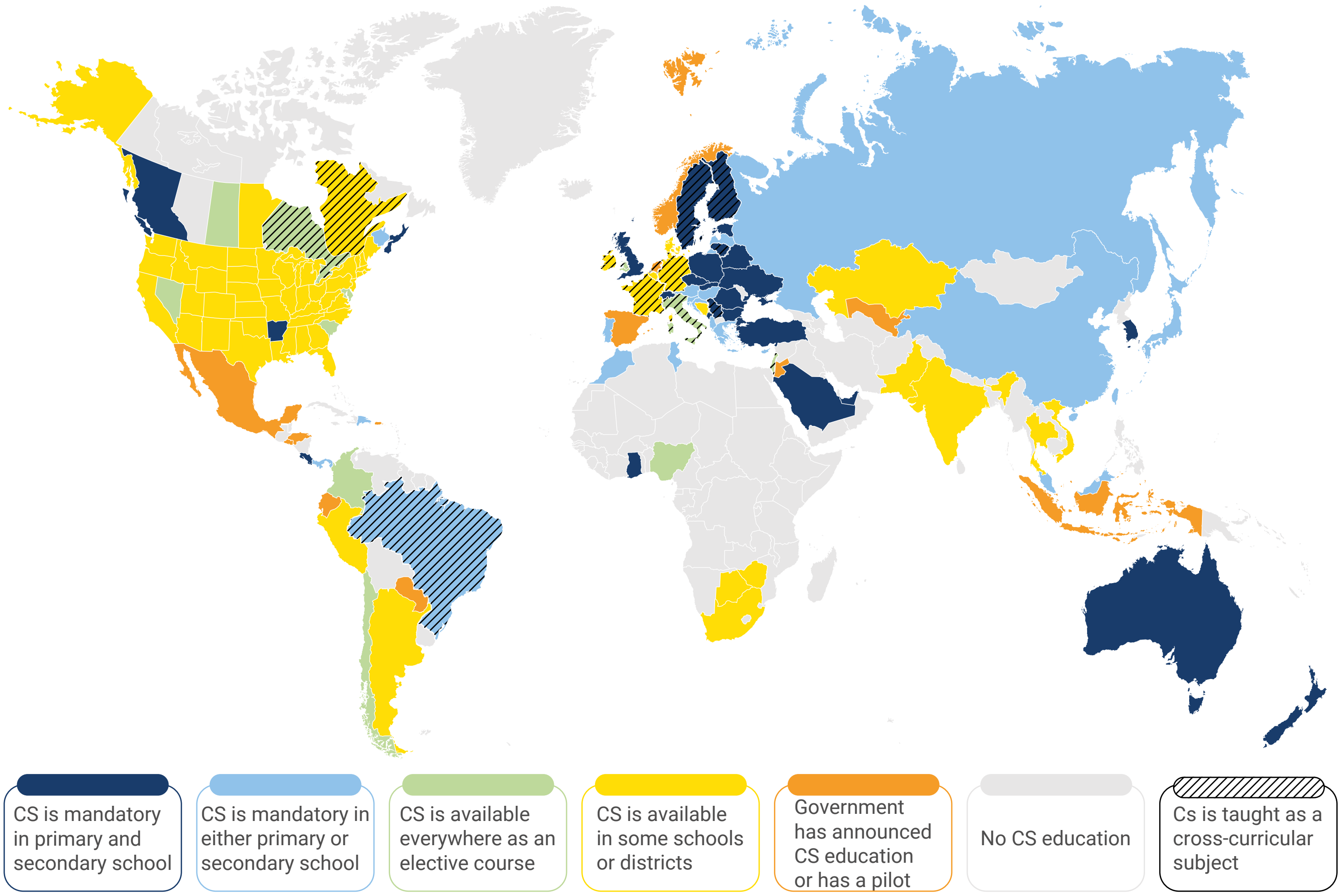


To better understand what is needed to expand CS education, we first took stock of the extent to which countries around the world have integrated CS education into primary and secondary schools, and how this varied by region and income level. We also reviewed the existing literature on integrating CS into K-12 education to gain a deeper understanding of the key barriers and challenges to expanding CS education globally. Then, we selected jurisdictions at various stages of progress in implementing CS education programs in from multiple regions of the world and income levels, and drafted in-depth case studies on the origins, key milestones, barriers, and challenges of CS expansion.

Progress in expanding CS education across the globe

As shown in Figure 4, the extent to which CS education is offered in primary and secondary schools varies across the globe. Countries with mandatory CS education are geographically clustered in Eastern Europe and East Asia. Most states and provinces in the U.S. and Canada offer CS on a school-to-school basis or as an elective course. Multiple countries in Western Europe offer CS education as a cross-curricular topic integrated into other subjects. Latin America and Central and Southeast Asia have the most countries that have announced CS education programs or pilot projects. Countries in Africa and the Middle East have integrated the least amount of CS education into school curricula. Nevertheless, the number of countries piloting programs or adopting CS curricula indicate a global trend of more education systems integrating the subject into their curriculum.

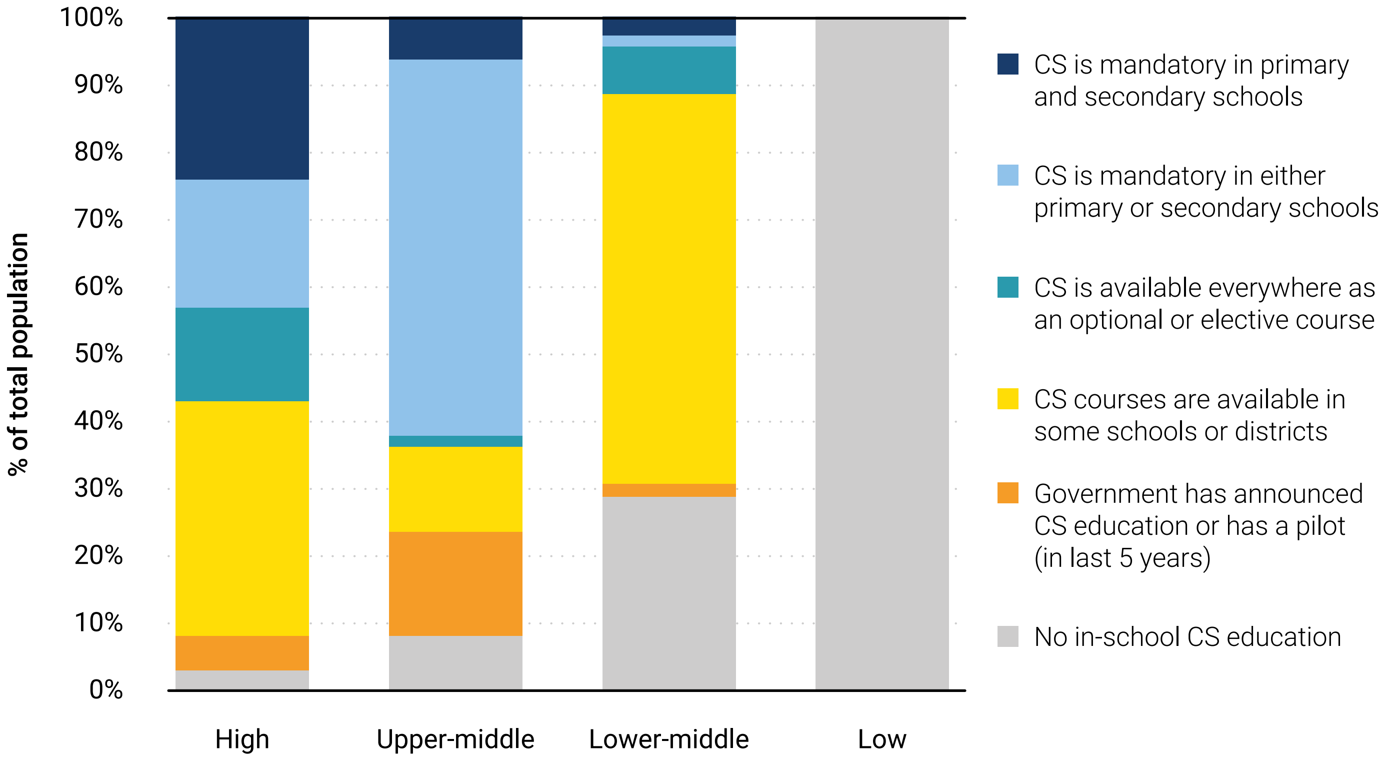
Figure 4. Computer science education varies across the globe


CS is mandatory in primary and secondary school
CS is mandatory in either primary or secondary school
CS is available everywhere as an elective course
CS is available in some schools or districts
Government has announced CS education or has a pilot
No CS education
Cs is taught as a cross-curricular subject

Note: Data come from 2021, and the authors used the World Bank income-level classification system.
Source: Authors' review of each education system's official curricula and related documents.

As expected, students living in higher-income countries generally have better access to CS education. As Figure 5 shows, 43 percent of high-income countries require students to learn CS education in primary and/or secondary schools. Additionally, high-income countries also offer CS as an elective course to the largest share of the population. A further 35 percent of high-income countries offer CS on a school-to-school basis while not making it mandatory for all schools. Interestingly, upper-middle income countries host the largest share of students (62 percent) who are required to learn CS at any point in primary or secondary schools. Presumably, many upper-middle income countries likely have national economic development strategies focused on expanding tech-related jobs, and thus see the need to expand the labor force with CS skills. By contrast, only 5 percent of lower-middle income countries require CS during primary or secondary school, while 58 percent may offer CS education on a school-to-school basis.

Figure 5. Computer science education is most widely available in high-income countries



Note: The authors used the World Bank income-level classification system.
Source: Authors' review of each education system's official curricula and related documents.

Key barriers and challenges to expand CS education globally

To expand quality CS education, education systems must overcome enormous challenges. Many countries do not have enough teachers who are qualified to teach CS, and even though there is growing interest among students to pursue CS, relatively few students pursue more advanced training like CS testing certifications (Department for Education, 2019) or CS undergraduate majors compared to other STEM fields like engineering or biology (Hendrickson, 2019). This is especially true for girls and underrepresented minorities, who generally have fewer opportunities to develop an interest in CS and STEM more broadly (Code.org & CSTA, 2018). Our review of the literature identified four key challenges to expanding CS education.:

1. Providing access to ICT infrastructure to students and educators

Student access to ICT infrastructure, including both personal access to computing devices and an internet connection, is critical to a robust CS education. Without this infrastructure, students cannot easily integrate CS skills into their daily lives, and they will have few opportunities to experiment with new approaches on their own.

However, some initiatives have succeeded by introducing elements of CS education in settings without adequate ICT infrastructure. For example, many educators use alternative learning strategies like CS Unplugged to teach CS and computational thinking when computers are unavailable (Bell & Vahrenhold, 2018). One study shows that analog lessons can help primary school students develop computational thinking skills (Harris, 2018). Other research has shown that introducing key programming concepts in unplugged ways helps students understand these concepts better in subsequent coding contexts (Grover, Jackiw and Lundh, 2019). Even without laptops or desktop computers, it is still possible for teachers to use digital tools for computational thinking. In South Africa, Professor Jean Greyling of Nelson Mandela University Computing Sciences co-created Tanks, a game that uses puzzle pieces and a mobile application to teach coding to children (Ellis, 2021). This is an especially useful concept, as many households and schools in South Africa and other developing countries have smartphones and access to analog materials but do not have access to personal computers or broadband connectivity (McCrocklin, 2021).

Taking a full CS curriculum to scale, however, requires investing in adequate access to ICT infrastructure for educators and students (Lockwood & Cornell, 2013). Indeed, as discussed in Section 3, our analysis of our case study analysis indicates that ICT infrastructure in schools provides a critical foundation to expand CS education.

2. Ensuring qualified teachers through teacher preparation and professional development

Many education systems encounter shortages of qualified CS teachers, contributing to a major bottleneck in CS expansion. A well-prepared and knowledgeable teacher is the most important component for instruction in commonly taught subjects (Chetty et al. 2014 a,b; Rivkin et al., 2005). We suspect this is no different for CS, though major deficiencies in the necessary CS skills among the teacher workforce are evident. For example, in a survey of preservice elementary school teachers in the United States, only 10 percent responded that they understood the concept of computational thinking (Campbell & Heller, 2019). As recently as six years ago, 75 percent of teachers in the U.S. incorrectly considered “creating documents or presentations on the computer” as a topic one would learn in a CS course (Google & Gallup, 2015), demonstrating a poor understanding of the distinction between CS and computer literacy. Other case studies, surveys, and interviews have found that teachers in India, Saudi Arabia, the U.K., and Turkey self-report low confidence in their understanding of CS (Ramen et al., 2015; Alfayez & Lambert, 2019; Royal Society, 2017; Gülbahar & Kalelioğlu, 2017). Indeed, developing the necessary skills and confidence levels for teachers to offer effective CS instruction remains challenging.

To address these challenges, school systems have introduced continuous professional development (PD), postgraduate certification programs, and CS credentials issued by teacher education degree programs. PD programs are common approaches, as they utilize the existing teacher workforce to fill the needs for special skills, rather than recruiting specialized teachers from outside the school system. For example, the British Computing Society created 10 regional university-based hubs to lead training activities, including lectures and meetings, to facilitate collaboration as part of the network of excellence (Dickens, 2016; Heintz et al., 2016; Royal Society, 2017). Most hubs involve multi-day seminars and workshops meant to familiarize teachers with CS concepts and provide ongoing support to help teachers as they encounter new challenges in the classroom. Cutts et al. (2017) further recommend teacher-led PD groups so that CS teachers can form collaborative professional networks. Various teacher surveys have found these PD programs in CS helpful (Alkaria & Alhassan, 2017; Goode et al., 2014). Still, more evidence is needed on the effectiveness of PD programs in CS education.

Less commonly, some education systems have worked with teacher training institutions to introduce certification schemes so teachers can signal their unique qualifications in CS to employers. This signal can make teacher recruitment more transparent and incentivize more teachers to pursue training. This approach does require, though, an investment in developing CS education faculty at the teacher training institution, which may be a critical bottleneck in many places (Delyser et al., 2018). Advocates of the approach have recommended that school systems initiate certification schemes quickly and with a low bar at first, followed by improvement over time (Code.org, 2017; Lang et al., 2013; Sentance & Csizmadia, 2017). Short-term recommendations include giving temporary licenses to teachers who meet minimum content and knowledge requirements. Long-term recommendations, on the other hand, encourage preservice teachers to take CS courses as part of their teaching degree programs or in-service teachers to take CS courses as part of their graduate studies to augment their skillset. Upon completing these courses, teachers would earn a full CS endorsement or certificate.

3. Fostering student engagement and interest in CS education

Surveys from various countries suggest that despite a clear economic incentive, relatively few K-12 students express interest in pursuing advanced CS education. For example, three out of four U.S. students in a recent survey declared no interest in pursuing a career in computer science. And the differences by gender are notable: nearly three times as many male students (33 percent) compared to female students (12 percent) expressed interest in pursuing a computer science career in the future (Google & Gallup, 2020).

Generally, parents view CS education favorably but also hold distinct misconceptions. For instance, more than 80 percent of U.S. parents surveyed in a

Google and Gallup (2016) study reported that they think CS is as important as any other discipline. Nevertheless, the same parents indicated biases around who should take CS courses: 57 percent of parents think that one needs to be "very smart" to learn CS (Google & Gallup, 2015). Researchers have equated this kind of thinking to the idea that some people could be inherently gifted or inept at CS, a belief that could discourage some students from developing an interest or talent in CS (McCartney, 2017). Contrary to this belief, Patitsas et al. (2019) found that only 5.8 percent of university-level exam distributions were multimodal, indicating that most classes did not have a measurable divide between those who were inherently gifted and those who were not. This signals that CS is no more specialized to specific groups of students than any other subject.

Fostering student engagement, however, does not equate to developing a generation of programmers. Employment projections suggest the future demand for workers with CS skills will likely outpace supply in the absence of promoting students' interest in the field. Yet, no countries expand access to CS education with the expectation of turning all students into computer programmers. Forcing students into career paths that are unnatural fits for their interests and skill levels result in worse outcomes for students at the decision margins (Kirkeboen et al., 2016). Rather, current engagement efforts both expose students to foundational skills that help navigate technology in 21st century life and provide opportunities for students to explore technical fields.

A lack of diversity in CS education not only excludes some people from accessing high-paying jobs, but it also reduces the number of students who would enter and succeed in the field (Du & Wimmer, 2019). Girls and racial minorities have been historically underrepresented in CS education (Sax et al., 2016). Research indicates that the diversity gap is not due to innate talent differences among demographic groups (Sullivan & Bers, 2012; Cussó-Calabuig et al., 2017), but rather a disparity of access to CS content (Google & Gallup 2016; Code.org & CSTA, 2018; Du & Wimmer, 2019), widely held cultural perceptions, and poor representation of women and underrepresented minorities (URMs) among industry leaders and in media depictions (Google & Gallup, 2015; Ayebi-Arthur, 2011; Downes & Looker, 2011).

To help meet the demand for CS professionals, government and philanthropic organizations have implemented programs that familiarize students with CS. By increasing student interest among K-12 students who may eventually pursue CS professions, these strategies have the potential to address the well documented lack of diversity in the tech industry (Harrison, 2019; Ioannou, 2018).

For example, some have used short, one-time lessons in coding to reduce student anxiety around CS. Of these lessons, perhaps the best known is Hour of Code, designed by Code.org. In multiple surveys, students indicated more confidence after exposure to this program (Phillips & Brooks, 2017; Lang et al., 2016; Doukaki et al., 2013). It is not clear, however, whether these programs make students more likely to consider semester-long CS courses (Phillips & Brooks, 2017; Lang et al., 2016).

Other initiatives create more time-intensive programs for students. The U.S. state of Georgia, for example, implemented a program involving after-school, weekend, and summer workshops over a six-year period. Georgia saw an increase in participation in the Advanced Placement (AP) CS exam during the duration of the program, especially among girls and URMs (Guzdial et al., 2014). Other states have offered similar programs, setting up summer camps and weekend workshops in universities to help high school students become familiar with CS (Best College Reviews, 2021). These initiatives, whether one-off introductions to CS or time-intensive programs, typically share the explicit goal of encouraging participation in CS education among all students, and especially girls and URMs.

Yet, while studies indicate that Hour of Code and summer camps might improve student enthusiasm for CS, they do not provide the kind of rigorous impact assessment one would need to make a definitive conclusion of their effectiveness. They do not use a valid control group, meaning that there is no like-for-like comparison to students who are similar except for no exposure to the program. It is not clear that the increase in girls and URMs taking CS would not have happened if it were not for Georgia's after-school clubs.

4. Generating and using evidence on curriculum and core competencies, instructional methods, and assessment

There is no one-size-fits-all CS curriculum for all education systems, schools, or classrooms. Regional contexts, school infrastructure, prior access, and exposure to CS need to be considered when developing CS curricula and competencies (Ackovska et al., 2015). Some CS skills, such as programming language, require access to computer infrastructure that may be absent in some contexts (Lockwood & Cornell, 2013). Rather than prescribing a curriculum, the U.S. K-12 Computer Science Framework Steering Committee (2016) recommends foundational CS concepts and competencies for education systems to consider. This framework encourages curriculum developers and educators to create learning experiences that extend beyond the framework to encompass student interests and abilities.

There is increasing consensus around what core CS competencies students should master when they complete primary and secondary education. Core competencies that students may learn by the end of *primary* school include:

1. abstraction—creating a model to solve a problem;
2. generalization—remixing and reusing resources that were previously created;
3. decomposition—breaking a complex task into simpler subtasks;
4. algorithmic thinking—defining a series of steps for a solution, putting instructions in the correct sequence, and formulating mathematical and logical expressions;
5. programming—understanding how to code a solution using the available features and syntax of a programming language or environment; and
6. debugging—recognizing when instructions do not correspond to actions and then removing or fixing errors (Angeli, 2016).

Competencies that *secondary* school students may learn in CS courses include:

1. logical and abstract thinking;
2. representations of data, including various kinds of data structures;
3. problem-solving by designing and programming algorithms using digital devices;
4. performing calculations and executing programs;
5. collaboration; and,
6. ethics such as privacy and data security (Syslo & Kwiatkowska, 2015).

Several studies have described various methods for teaching CS core competencies. Integrated development environments are recommended especially for teaching coding skills (Florez et al., 2017; Saez-Lopez et al., 2016).[2] These environments include block-based programming languages that encourage novice programmers to engage with programming, in part by alleviating the burden of syntax on learners (Weintrop & Wilensky, 2017; Repenning, 1993). Others recommend a variety of teaching methods that blend computerized lessons with offline activities (Taub et al. 2009; Curzon et al., 2009, Ackovska et al., 2015). This approach is meant to teach core concepts of computational thinking while keeping students engaged in physical, as well as digital, environments (Nishida et al., 2009). CS Unplugged, for example, provides kinesthetic lesson plans that include games and puzzles that teach core CS concepts like decomposition and algorithmic thinking.

Various studies have also attempted to measure traditional lecture-based instruction for CS (Alhassan 2017; Cicek & Taspinar, 2016).[3] These studies, however, rely on small sample sizes wherein the experiment and control group each comprised of individual classes. More rigorous research is required to understand the effectiveness of teaching strategies for CS.

[2] Integrated development environments include programs like Scratch (Resnick et al., 2009), Code.org (Kelelioglu, 2015), and CHERP3 Creative Hybrid Environment for Robotics Programming (Bers et al., 2014).

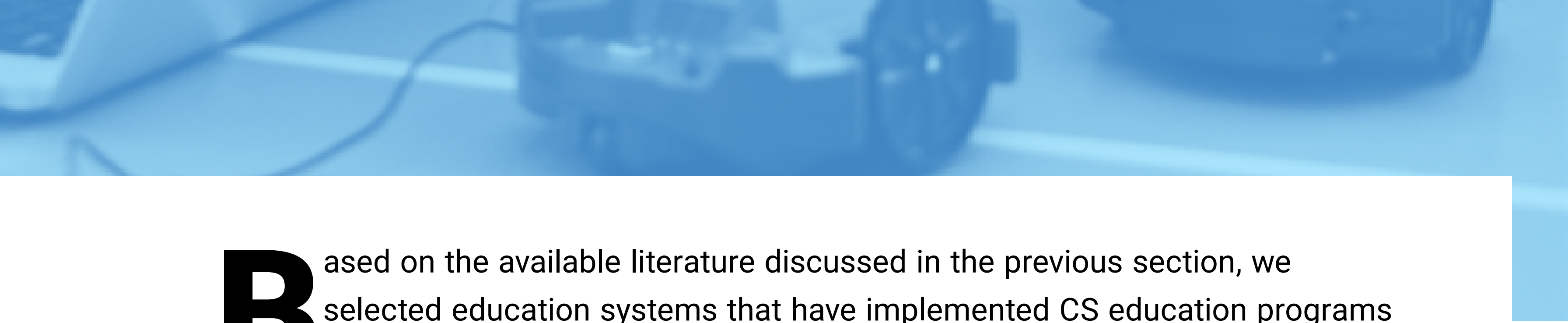
[3] The authors of these studies conclude that self-teaching methods and laboratory control methods may be effective for teaching programming skills.

No consensus has emerged on the best ways to assess student competency in core CS concepts, (So et al., 2019; Djambong & Freiman, 2016). Though various approaches to assessment are widely available—including classical cognitive tests, standardized tests in digital environments, and CS Unplugged activity tests—too many countries have yet to introduce regular assessments that may evaluate various curricula or instructional methods in CS. While several assessments have been developed for CS and CT at various grade levels as part of various research studies, there have been challenges to broader use. This is due to either a lack of large-scale studies using these assessments or diversity in programming environments used to teach programming and CS or simply a lack of interest in using objective tests of learning (as opposed to student projects and portfolios).

Fortunately, a growing number of organizations are developing standardized tests in CS and computational thinking. For example, the International Computer and Information Literacy Study included examinations in computational thinking in 2018 that had two 25-minute modules, where students were asked to develop a sequence of tasks in a program that related to a unified theme (Fraillon et al., 2020). The OECD's PISA will also include questions in 2021 to assess computational thinking across countries (Schleicher & Partovi, 2019). The AP CS exam has also yielded useful comparisons that have indirectly evaluated CS teacher PD programs (Brown & Brown, 2019).

In summary, the current evidence base provides little consensus on the specific means of scaling a high-quality CS education and leaves wide latitude for experimentation. Consequently, in this report we do not offer prescriptions on how to expand CS education, even while arguing that expanding access to it generally is beneficial for students and the societies that invest in it. Given the current (uneven) distribution of ICT infrastructure and CS education resources, high-quality CS education may be at odds with expanded access. While we focus on ensuring universal access first, it is important to recognize that as CS education scales both locally and globally, the issues of curricula, pedagogies, instructor quality, and evaluation naturally become more pressing.

3. Lessons from education systems that have introduced CS education



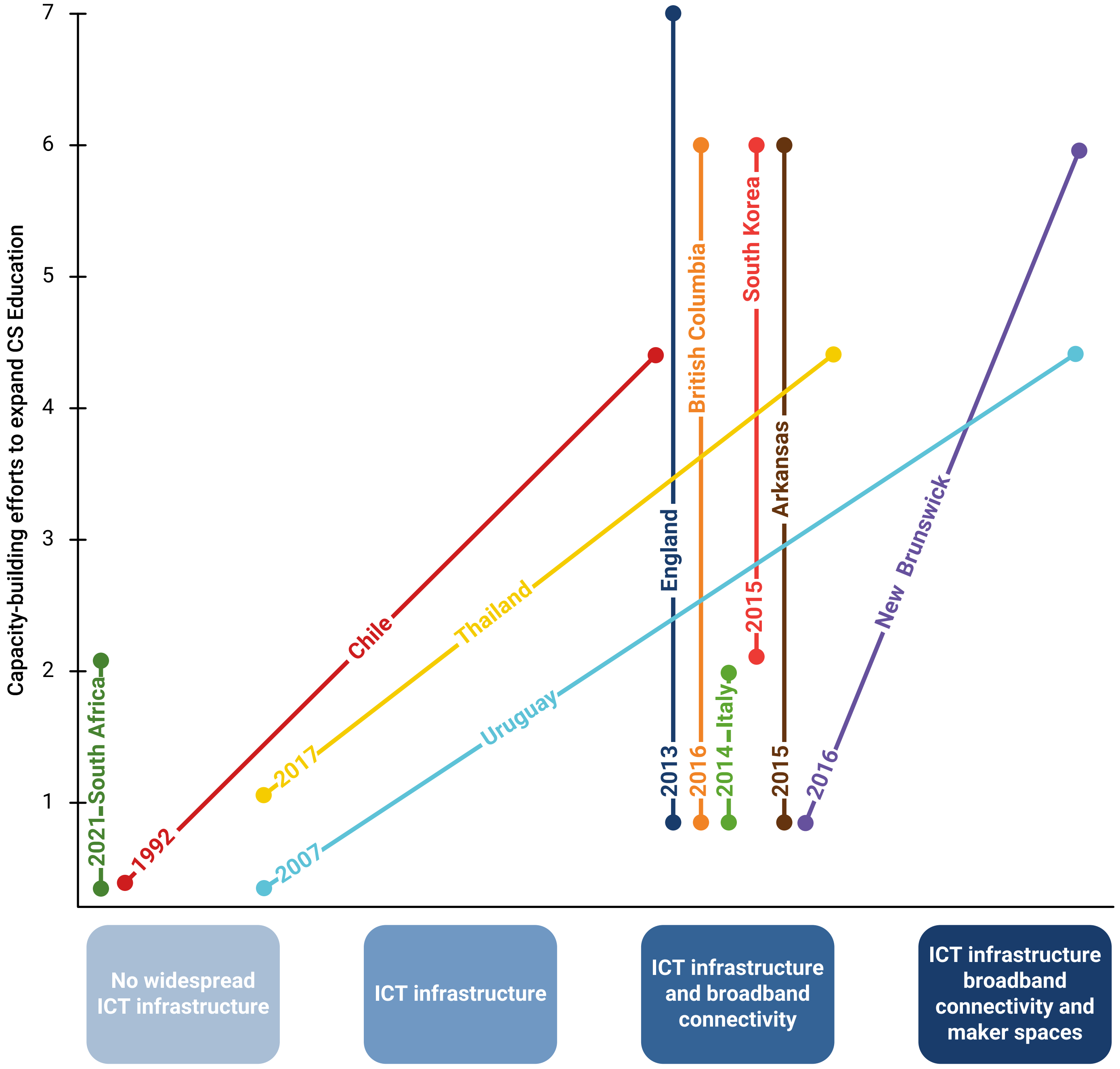
Based on the available literature discussed in the previous section, we selected education systems that have implemented CS education programs and reviewed their progress through in-depth case studies. Intentionally, we selected jurisdictions at various levels of economic development, at different levels of progress in expanding CS education, and from different regions of the world. They include Arkansas (U.S.), British Columbia (Canada), Chile, England, Italy, New Brunswick (Canada), Poland, South Africa, South Korea, Thailand, and Uruguay. For each case, we reviewed the historical origins for introducing CS education and the institutional arrangements involved in CS education's expansion. We also analyzed how the jurisdictions addressed the common challenges of ensuring CS teacher preparation and qualification, fostering student demand for CS education (especially among girls and URMs), and how they developed curriculum, identified core competencies, promoted effective instruction, and assessed students' CS skills. In this section, we draw lessons from these case studies, which can be found in separate case studies online.[4]

Figure 6 presents a graphical representation summarizing the trajectories of our selected case studies as they expanded CS education. Together, the elements in the figure provide a rough approximation of how CS education has expanded in recent years in each case. For example, when South Korea started its efforts to focus on universal CS education in 2015, basic ICT infrastructure and broadband connectivity were already available in all schools and two CS expansion policies had been previously implemented. Its movement since 2015 is represented purely in the vertical policy action space, as it moved up four intervals on the index. Uruguay, conversely, started its CS expansion at a lower level both in terms of ICT infrastructure (x-axis) and existing CS policies (y-axis). Since starting CS expansion efforts in 2007, though, it has built a robust ICT infrastructure in its school system and implemented 4 of 7 possible policy actions.

[4] In addition to finding the case studies on the report landing page, they are located here: https://www.brookings.edu/product/computer-science-education/.

Figure 6 suggests that first securing access to ICT infrastructure and broadband connectivity allows systems to dramatically improve access to and the quality of CS education. Examples include England, British Columbia, South Korea, and Arkansas. At the same time, Figure 6 suggests that systems that face the dual challenge of expanding ICT infrastructure and broadband connectivity *and* scaling the delivery of quality CS education, such as Chile, South Africa, Thailand, and Uruguay, may require more time and/or substantial investment to expand quality CS education to match the former cases. Even though Chile, Thailand, and especially Uruguay have made impressive progress since their CS education expansion efforts began, they continue to lag a few steps behind those countries that started with established ICT infrastructure in place.

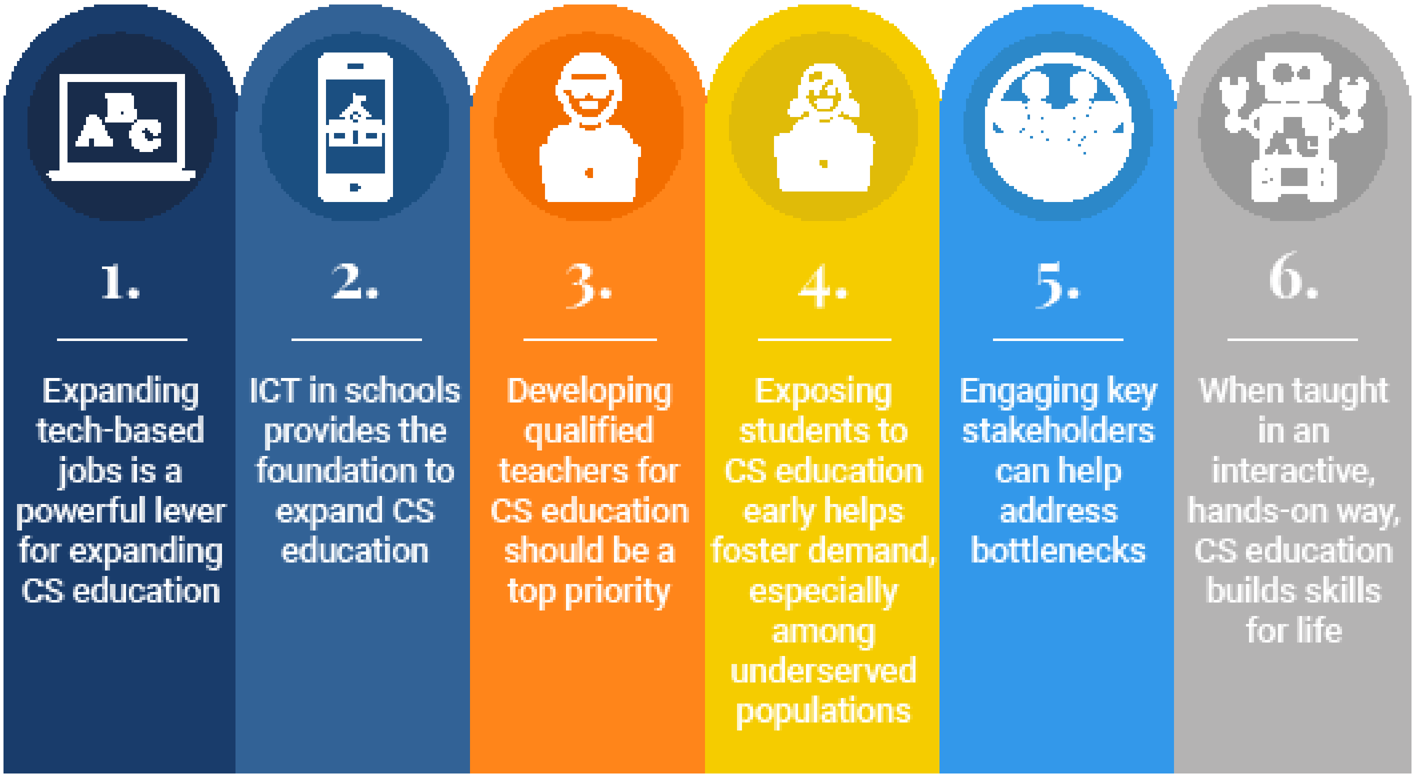
Figure 6. ICT infrastructure improves access to computer science education but more capacity is needed to ensure quality



Notes: The horizontal axis indicates the level of ICT infrastructure present in each education system. The vertical axis indicates the degree to which the education system has implemented the (noninfrastructure) capacity to expand CS education. The measure here is based on a 7-point index that is an additive sum of common policy actions taken to expand access and improve the quality of a system's CS education program. These policy actions include: (1) introduction of ICT education programs; (2) requirement for CS in primary education; (3) requirement for CS in secondary education; (4) introduction of in-service CS teacher education programs; (5) introduction of preservice teacher education programs; (6) availability of a specialized center or institution focused on CS education research and training; and (7) access to regular funding allocated to CS education by the legislative branch of government. The lines represent trajectories in these two dimensions for each country—each line records both a starting point (the bottom / left point) and a point representing the present (the top / right point). Each line in Figure 6 also notates a year, which corresponds to each education system's start date for implementing CS education in earnest.

Our analysis of these case studies surfaced six key lessons (Figure 7) for governments wishing to take CS education to scale in primary and secondary schools, which we discuss in further detail below.

Figure 7. Six key lessons to scale CS education



1. Expanding tech-based jobs is a powerful lever for expanding CS education

2. ICT in schools provides the foundation to expand CS education

3. Developing qualified teachers for CS education should be a top priority

4. Exposing students to CS education early helps foster demand, especially among underserved populations

5. Engaging key stakeholders can help address bottlenecks

6. When taught in an interactive, hands-on way, CS education builds skills for life

1. Expanding tech-based jobs is a powerful lever for expanding CS education

In several of the case studies, economic development strategies were the underlying motivation to introduce or expand CS education. For example, Thailand's 2017 20-year Strategic Plan marked the beginning of CS education in that country. The 72-page document, approved by the Thai Cabinet and Parliament, explained how Thailand could become a more "stable, prosperous, and sustainable" country and proposed to reform the education curriculum to prepare students for future labor demands (20-year National Strategy comes into effect, 2018). Similarly, Arkansas's Governor Hutchinson made CS education a key part of his first campaign in 2014 (CS for All, n.d.), stating that "Through encouraging computer science and technology as a meaningful career path, we will produce more graduates prepared for the information-based economy that represents a wide-open job market for our young people" (Arkansas Department of Education, 2019).

Uruguay's Plan Ceibal, named after the country's national flowering tree, was likewise introduced in 2007 as a presidential initiative to incorporate technology in education and help close a gaping digital divide in the country. The initiative's main objectives were to promote digital inclusion, graduate employability, a national digita culture, higher-order thinking skills, gender equity, and student motivation (Jara, Hepp, & Rodriguez, 2018)

Last, in 2018, the European Commission issued the Digital Education Action Plan that enumerated key digital skills for European citizens and students, including CS and computational thinking (European Commission, 2018). The plan encouraged young Europeans to understand the algorithms that underpin the technologies they use on a regular basis. In response to the plan, Italy's 2018 National Indications and New Scenarios report included a discussion on the importance of computational thinking and the potential role of educational gaming and robotics in enhancing learning outcomes (Giacalone, 2019). Then, in 2019, the Italian Ministry of Education and the Parliament approved a legislative motion to include CS and computational thinking in primary school curricula by 2022 (Orizzontescuola, 2019).

In some cases, the impetus to expand CS education came more directly from demands from key stakeholders, including industry and parents. For example, British Columbia's CS education program traces back to calls from a growing technology industry (Doucette, 2016). In 2016, the province's technology sector employed 86,000 people—more than the mining, forestry, and oil and gas sectors combined, with high growth projections (Silcoff, 2016). The same year, leaders of the province's technology companies revealed in interviews that access to talent had become their biggest concern (KPMG, 2016). According to a 2016 B.C.

Technology Association report, the province needed 12,500 more graduates in CS from tertiary institutions between 2015 and 2021 to fill unmet demand in the labor market (Orton, 2018). The economic justification for improving CS education in the province was clear.

Growing parental demand helped create the impetus for changes to the CS curriculum in Poland. According to Kozlowski (2016), Polish parents perceive CS professions as some of the most desirable options for their children. And given the lack of options for CS education in schools, parents often seek out extracurricular workshops for their children to encourage them to develop their CS skills (Panskyi, Rowinska, & Biedron, 2019). The lack of in-school CS options for students created the push for curricular reforms to expand CS in primary and secondary schools. As former Minister of Education Anna Zalewska declared, Polish students "cannot afford to waste time on [the] slow, arduous task of building digital skills outside school [and] only school education can offer systematic teaching of digital skills" (Szymański, 2016).

2. ICT in schools provides the foundation to expand CS education

Previous efforts to expand access to devices, connectivity, or basic computer literacy in schools provided a starting point in several jurisdictions to expand CS education. For example, the Uruguayan government built its CS education program after implementing expansive one-to-one computing projects, which made CS education affordable and accessible. In England, an ICT course was implemented in schools in the mid-1990s. These dedicated hours during the school day for ICT facilitated the expansion of CS education in the country.

The Chilean Enlaces program, developed in 1992 as a network of 24 universities, technology companies, and other organizations (Jara, Hepp, & Rodriguez, 2018; Sánchez & Salinas, 2008) sought to equip schools with digital tools and train teachers in their use (Severin, 2016). It provided internet connectivity and digital devices that enabled ICT education to take place in virtually all of Chile's 10,000 public and subsidized private schools by 2008 (Santiago, Fiszbein, Jaramillo, &

Radinger, 2017; Severin et al., 2016). Though Enlaces yielded few observable effects on classroom learning or ICT competencies (Sánchez & Salinas, 2008), the program provided the infrastructure needed to begin CS education initiatives years later.

While a history of ICT expansion can serve as a base for CS education, institutional flexibility to transform traditional ICT projects into CS education is crucial. The Chilean Enlaces program's broader institutional reach resulted in a larger bureaucracy, slower implementation of new programs, and greater dependence on high-level political agendas (Severin, 2016). As a result, the program's inflexibility prevented it from taking on new projects, placing the onus on the Ministry of Education to take the lead in initiating CS education. In Uruguay, Plan Ceibal's initial top-down organizational structure enabled relatively fast implementation of the One Laptop per Child program, but closer coordination with educators and education authorities may have helped to better integrate education technology into teaching and learning. More recently, Plan Ceibal has involved teachers and school leaders more closely when introducing CS activities. In England, the transition from ICT courses to a computing curriculum that prioritized CS concepts, instead of computer literacy topics that the ICT teachers typically emphasized before the change, encountered some resistance. Many former ICT teachers were not prepared to implement the new program of study as intended, which leads us to the next key lesson.

3. Developing qualified teachers for CS education should be a top priority

The case studies highlight the critical need to invest in training adequate numbers of teachers to bring CS education to scale. For example, England took a modest approach to teacher training during the first five years of expanding its CS education K-12 program and discovered that its strategy fell short of its original ambitions. In 2013, the English Department for Education (DfE) funded the BCS to establish and run the Network of Excellence to create learning hubs and train a pool of "master" CS teachers. While over 500 master teachers were trained, the numbers were insufficient to expand CS education at scale. Then, in 2018 the DfE substantially increased its funding to establish the National Center for Computing Education (NCCE) and added 23 new computing hubs throughout England. Hubs offer support to primary and secondary computing teachers in their designated areas, including teaching, resources, and PD (Snowdon, 2019). In just over two years, England has come a long way toward fulfilling its goals of training teachers at scale with over 29,500 teachers engaged in some type of training (Teach Computing, 2020).

Several education systems partnered with higher education institutions to integrate CS education in both preservice and in-service teacher education programs. For example, two main institutions in British Columbia, Canada—the University of British Columbia and the University of Northern British Columbia—now offer CS courses in their pre-service teacher education programs. Similarly, in Poland, the Ministry of National Education sponsored teacher training courses in university CS departments. In Arkansas, state universities offer CS certification as part of preservice teacher training while partnering with the Arkansas Department of Education to host in-service professional development.

Still other systems partnered with nonprofit organizations to deliver teacher education programs. For instance, New Brunswick, Canada, partnered with the nonprofit organization Brilliant Labs to implement teacher PD programs in CS (Brilliant Labs, n.d.). In Chile, the Ministry of Education partnered with several nongovernmental organizations, including Code.org and Fundación Telefónica, to expand teacher training in CS education. Microsoft Philanthropies launched the Technology Education and Literacy in Schools (TEALS) in the United States and Canada to connect high school teachers to technology industry volunteers. The volunteer experts support instructors to learn CS independently over time and develop sustainable high school CS programs (Microsoft, n.d.).

To encourage teachers to participate in these training programs, several systems introduced teacher certification pathways in CS education. For example, in British Columbia, teachers need at least 24 credits of postsecondary coursework in CS education to be qualified to work in public schools. The Arkansas Department of

Education incentivizes in-service teachers to attain certification through teaching CS courses and participating in approved PD programs (Code.org, CSTA, ECEP, 2019). In South Korea, where the teaching profession is highly selective and enjoys high social status, teachers receive comprehensive training on high-skill computational thinking elements, such as computer architecture, operating systems, programming, algorithms, networking, and multimedia. Only after receiving the "informatics–computer" teacher's license may a teacher apply for the informatics teacher recruitment exam (Choi et al., 2015).

When faced with shortages of qualified teachers, remote instruction can provide greater access to qualified teachers. For example, a dearth of qualified CS teachers has been and continues to be a challenge for Uruguay. To address this challenge, in 2017, Plan Ceibal began providing remote instruction in computational thinking lessons for public school fifth and sixth graders and integrated fourth-grade students a year later. Students work on thematic projects anchored in a curricular context where instructors integrate tools like Scratch.[5] During the school year, a group of students in a class can work on three to four projects during a weekly 45-minute videoconference with a remote instructor, while another group can work on projects for the same duration led by the classroom teacher. In a typical week, the remote instructor introduces an aspect of computational thinking. The in-class teacher then facilitates activities like block-based programming, circuit board examination, or other exercises prescribed by the remote teacher (Cobo & Montaldo, 2018).[6] Importantly, Plan Ceibal implements Pensamiento Computacional, providing a remote instructor and videoconferencing devices at the request of schools, rather than imposing the curriculum on all classrooms (García, 2020). With the ongoing COVID-19 pandemic forcing many school systems across the globe to adopt remote instruction, at least temporarily, we speculate that remote learning is now well poised to become more common in expanding CS education in places facing ongoing teacher shortages.

[5] In 2019, President Tabaré Vázquez stated that "All children in kindergartens and schools are programming in Scratch, or designing strategies based on problem-solving" (Uruguay Presidency, 2019).

[6] Remote instruction via videoconferencing technology improved learning in mathematics in an experiment in Ghana (Johnston & Ksoll, 2017). It is very plausible that Uruguay's approach to giving computational thinking instruction via videoconference could also be effective.

4. Exposing students to CS education early helps foster demand, especially among underserved populations

Most education systems have underserved populations who lack the opportunity to develop an interest in CS, limiting opportunities later in life. For example, low CS enrollment rates for women at Italian universities reflect the gender gap in CS education. As of 2017, 21.6 percent and 12.3 percent of students completing bachelor's degrees in information engineering and CS, respectively, were women (Marzolla, 2019). Further, female professors and researchers in these two subjects are also underrepresented. In 2018, only 15 percent and 24 percent of professors and researchers in CS and computer engineering, respectively, were women (Marzolla, 2019). Similar representation gaps at the highest levels of CS training are common globally. Thus, continuing to offer exposure to CS only in post-secondary education will likely perpetuate similar representation gaps.

To address this challenge, several education systems have implemented programs to make CS education accessible to girls and other underserved populations in early grades, before secondary school. For instance, to make CS education more gender balanced, the Italian Ministry of Education partnered with civil society organizations to implement programs to spur girls' interest in CS and encourage them to specialize in the subject later (European Commission, 2009). An Italian employment agency (ironically named Men at Work) launched a project called *Girls Code It Better* to extend CS learning opportunities to 1,413 middle school girls across 53 schools in 2019 (Girls Code It Better, n.d.). During the academic year, the girls attended extracurricular CS courses before developing their own technologically advanced products and showcasing their work at an event at Bocconi University in Milan (Brogi, 2019). In addition to introducing the participants to CS, the initiative provided the girls with role models and generated awareness on the gender gap in CS education in Italy.

In British Columbia, students are exposed to computational thinking concepts as early as primary school, where they learn how to prototype, share, and test ideas. In the early grades of primary education, the British Columbia curriculum emphasizes numeracy using technology and information technology. Students develop numeracy skills by using models and learn information technology skills to apply across subjects. In kindergarten and first grade, curricular objectives include preparing students for presenting ideas using electronic documents. In grades 2 to 3, the curricular goals specify that students should "demonstrate an awareness of ways in which people communicate, including the use of technology," in English language arts classes, as well as find information using information technology tools. By the time students are in grades 4 and 5, the curriculum expects students to focus more on prototyping and testing new ideas to solve a problem (Gannon & Buteau, 2018).

Several systems have also increased participation in CS education by integrating it as a cross-curricular subject. This approach avoids the need to find time during an already-packed school day to teach CS as a standalone subject. For example, in 2015, the Arkansas legislature began requiring elementary and middle school teachers to embed computational thinking concepts in other academic courses. As a result, teachers in the state integrate five main concepts of computational thinking into their lesson plans, including (1) problem-solving, (2) data and information, (3) algorithms and programs, (4) computers and communications, and, importantly, (5) community, global, and ethical impacts (Watson-Fisher, 2019). In the years following this reform, the share of African-American students taking CS in high school reached 19.6 percent, a figure that slightly exceeds the percentage of African Americans among all students—a resounding sign of progress in creating student demand for CS education (Computer science on the rise in Arkansas schools, Gov. drafts legislation to make it a requirement for graduation, 2020).

After-school programs and summer camps, jointly organized with external partners, have also helped promote demand for CS education through targeted outreach programs to commonly underserved populations. For example, Microsoft Thailand has been holding free coding classes, Hour of Code, in partnership with nonprofit organizations, to encourage children from underprivileged backgrounds to pursue STEM education (Microsoft celebrates Hour of Code to build future ready generations in Asia, 2017). In the past decade, Microsoft has extended opportunities for ICT and digital skills development to more than 800,000 youth from diverse backgrounds—including those with disabilities and residents of remote communities (Thongnab, 2019). Their annual #MakeWhatsNext event for young Thai women showcases STEM careers and the growing demand for those careers (Making coding fun for Thailand's young, 2018). Also in Thailand, Redemptorist Foundation for People with Disabilities, with over 30 years of experience working with differently abled communities in that country, expanded their services to offer computer trainings and information technology vocational certificate programs for differently abled youth (Mahatai, n.d.).

In British Columbia, Canada, the Ministry of Education and other stakeholders have taken steps to give girls, women, and aboriginal students the opportunity to develop an interest in CS education. For example, after-school programs have taken specific steps to increase girls' participation in CS education. The UBC Department of Computer Science runs GIRLsmarts4tech, a program that focuses on giving 7th-grade girls role models and mentors that encourage them to pursue technology-related interests (GIRLsmarts4tech, n.d.). According to the latest census, in 2016, British Columbia's First Nations and Indigenous Peoples (FNIP) population—including First Nations, Metis, and Inuits—was 270,585, an increase of 38 percent from 2006. With 42.5 percent of the FNIP population under 25, it is critical for the province to deliver quality education to this young and growing group (Ministry of Advanced Education, Skills and Training, 2018). To this end, part of the British Columbia curriculum for CS education incorporates FNIP world views, perspectives, knowledge, and practices in CS concepts. In addition, the B.C. based ANCESTOR project (AborigiNal Computer Education through STORytelling) has organized courses and workshops to encourage FNIP students to develop computer games or animated stories related to their culture and land (Westor & Binn, 2015).

As these examples suggest, private sector and nongovernmental organizations can play an important role in the expansion of CS education, an issue we turn to now.

5. Engaging key stakeholders can help address bottlenecks

In most reviewed cases, the private sector and nongovernmental organizations played a role in promoting the expansion of CS education. Technology companies not only helped to lobby for expanding CS education, but often provided much-needed infrastructure and subject matter expertise in the design and rollout of CS education. For example, Microsoft Thailand has worked with the Thai government since 1998 in various capacities, including contributing to the development and implementation of coding projects, digital skills initiatives, teacher training programs, and online learning platforms (Thongnab, 2019; Coding Thailand, n.d.). Since 2002, Intel's Teach Thailand program has trained more than 150,000 teachers. Additionally, Google Coding Teacher workshops train educators on teaching computational thinking through CS Unplugged coding activities (EduTech Thailand, 2019). The workshop is conducted by Edutech (Thailand) Co., Ltd., an educational partner of Google, which adapted the Google curriculum to the Thailand education context. Samsung has been engaged in a smart classroom project that has built futuristic classroom prototypes and provided training for 21st century competencies (OECD/UNESCO, 2016).

In England, nongovernmental organizations have played an important role in supporting the government's expansion of CS education. The DfE has relied on outside organizations for help in executing its CS education responsibilities. The DfE's NCEE, for instance, is delivered by a consortium including the British Computing Society, STEM Learning, and the Raspberry Pi Foundation—three nonprofit organizations dedicated to advancing the computing industry and CS education in the country (British Computing Society, n.d; STEM Learning, n.d.; Raspberry Pi Foundation, n.d.).

Chile's Ministry of Education developed partnerships with individual NGOs and private companies to engage more students, especially girls. These initiatives offer the opportunity for hands-on learning projects and programming activities that students can perform from their home computers. Some of the same partners also provide online training platforms for teacher PD.

Industry advocacy organizations can also play an important role in the expansion of CS education. For example, in Arkansas, the state’s business community has long supported CS education (Nix, 2017). Accelerate Arkansas was established in 2005 as an organization of 70 private and public sector members dedicated to moving Arkansas into a more innovation- and knowledge-based economy (State of Arkansas, 2018). Similarly, in England, a network of organizations called Computing at School established a coalition of industry representatives and teachers. It played a pivotal role in rebranding the ICT education program in 2014 to the computing program that placed a greater emphasis on CS (Royal Society, 2017).

To ensure sustainability, one key lesson is that the government should coordinate across multiple stakeholders. The reliance on inputs from external organizations to drive CS education implies that the heavy reliance on NGO-provided training and resources in Chile have been insufficient to motivate more schools and teachers to include CS and computational thinking in classroom learning activities. By contrast, England's DfE has effectively coordinated across various nongovernmental organizations to expand CS education. Similarly, Arkansas’s Department of Education is leading an effort to get half of all school districts to form partnerships with universities and business organizations to give students opportunities to participate internships and college-level CS courses while in high school (Talk Business & Politics, 2020). In sum, the experience of decades of educational policies across the education systems reviewed shows that schools require long-lasting, coordinated, and multidimensional support to achieve successful implementation of CS in classrooms.

6. When taught in an interactive, hands-on way, CS education builds skills for life

Several of the cases studied introduced innovative pedagogies using makerspaces (learning spaces with customizable layouts and materials) and project-based learning to develop not only skills specific to CS but also skills that are relevant more broadly for life. For example, Uruguayan CS education features innovative concepts like robotics competitions and makerspaces that allow students to creatively apply their computational thinking lessons and that can spark interest and deepen understanding. In addition, computational thinking has been integrated across subject areas (e.g., in biology, math, and statistics) (Vázquez et al., 2019) and in interdisciplinary projects that immerse students in imaginative challenges that foster creative, challenging, and active learning (Cobo & Montaldo, 2018). For instance, students can use sensors and program circuit boards to measure their own progress in physical education (e.g., measuring how many laps they can run in a given period).

Similarly, in New Brunswick, Brilliant Labs provide learning materials to schools so they can offer students CS lessons using makerspaces that encourage students to develop projects, engage with technology, learn, and collaborate. These makerspaces enable students to creatively apply their CS and computational thinking lessons, sparking interest and deepening understanding of CS and computational thinking.

Thailand's curricular reforms also integrated project-based learning into CS education. Thai students in grades 4-6 learn about daily life through computers, including skills such as using logic in problem-solving, searching data and assessing its correctness, and block coding (e.g., Scratch). Then, students in grades 7-9 focus on learning about primary data through objectives that include using programming to solve problems, collecting, analyzing, presenting, and assessing data and information, and textual programming such as Python. Finally, students in grades 10-12 focus on applying advanced computing technology and programming to solve real-world problems, using knowledge from other subjects and data from external sources (Piamsa-nga et al., 2020).

After two years of nationwide discussions from 2014 to 2016, the Polish Ministry of National Education announced the creation of a new core curriculum for CS in primary and secondary schools (Syslo, 2020). The new curriculum's goals included students using technology to identify solutions for problems in every day and professional situations and supporting other disciplines—such as science, the arts, and the social sciences—in innovation (Panskyi, Rowinska, & Biedron, 2019).

In sum, these examples showcase efforts across the world to leverage CS education to develop skills useful not just in the technology industry, but also in many occupations, as well as for active citizenship in the 21st century.

Conclusion

CS skills are increasingly necessary to function in today's technology-advanced world and for the future. They enable individuals to understand how technology works, and how best to harness its potential to improve lives. As these skills take preeminence in the rapidly changing 21st century, CS education promises to significantly enhance student preparedness for the future of work and active citizenship.

But the benefits of CS education go beyond economic benefits and civic participation. Given the increasing integration of technology into many aspects of daily life in the 21st century, a functional knowledge of how computers work will benefit all individuals, enabling them to participate effectively and safely in an increasingly digital world.

While we do not argue that all students need to become computer scientists, foundational CS skills are required for all students to reach their full potential in the economy and society. We consider CS skills a "new basic skill," as important as foundational literacy and numeracy to thrive in the 21st century.

We recognize that efforts to expand CS education require financial resources and capacity building. And although especially after the COVID-19 pandemic, governments around the world are facing competing demands amid financial constraints, the costs of not investing in CS education will leave many of today's children and youth further behind. Indeed, more than ever, failing to invest in technology and CS education may result in constrained global competitiveness, leaving a government less able to support its citizens.

The already large gaps in skills and competitiveness between the richest and poorest are likely to continue to grow, unless action is taking sooner rather than later. Indeed, the extent to which CS education is offered in primary and secondary schools varies across the globe. Students living in higher-income countries generally have better access to CS education. And while upper-middle income countries have the highest rates of mandatory CS education, students in the poorest countries of the world have the least access to CS education.

To expand quality CS education, education systems must overcome four main challenges: (1) provide primary and secondary school students and teachers access to ICT infrastructure; (2) ensure qualified teachers through teacher preparation and qualification; (3) foster student demand for CS education; and (4) develop and use evidence on curriculum and core competencies, instructional methods, and assessment of CS skills.

Several jurisdictions around the world have made substantial efforts to provide CS education in primary and secondary schools and offer lessons for others committed to doing so. We analyzed jurisdictions at various levels of economic development, at different levels of progress in expanding CS education, and from different regions of the world.

Our findings suggest six recommendations for governments interested in taking CS education to scale in primary and secondary schools. First, governments should use economic development strategies focused on expanding technology-based jobs to engage all stakeholders and expand CS education in primary and secondary schools. Indeed, such a strategy helps attract and retain investors and foster CS education demand among students. Second, provide access to ICT infrastructure in primary and secondary schools to facilitate the introduction and expansion of CS education. Third, developing qualified teachers for CS should be a top priority. The evidence is clear that a qualified teacher is the most important factor in student learning, and thus preparing the teacher force needed for CS at scale is crucial. Fourth, expose students early to CS education to increase their likelihood of pursuing it. This is especially important for girls and other groups historically underrepresented in STEM and CS fields. Fifth, engage key stakeholders (including educators, the private sector, and civil society) to help address bottlenecks in physical and technical capacity. Finally, teach CS in an interactive, hands-on way to build skills for life.

Through studying the cases of regional and national governments at various levels of economic development and progress in implementing CS education programs, governments from around the globe can learn how to expand and improve CS education and develop a new basic skill necessary for the future of work and active citizenship.

References

20-YEAR NATIONAL STRATEGY COMES INTO EFFECT. (2018, OCTOBER 13). *Bangkok Post*. Retrieved from https://www.bangkokpost.com/thailand/general/1557462/20-year-national-strategy-comes-into-effect.

ACKOVSKA, N., ERDŐSNÉ, N., A., STANKOV, E., & JOVANOV, M. (2015). Report of the IOI Workshop, “Creating an international informatics curriculum for primary and high school education.” Olympiads in Informatics.

ALFAYEZ, A. A., & LAMBERT, J. (2019). Exploring Saudi computer science teachers’ conceptual mastery level of computational thinking skills. *Computers in the Schools,* 36(3), 143-166.

ALHASSAN, R. (2017). The effect of employing self-explanation strategy with worked examples on acquiring computer programing skills. *Journal of Education and Practice,* 8(6), 186-196.

ALKARIA, A., & ALHASSAN, R. (2017). The effect of in-service training of computer science teachers on scratch programming language skills using an electronic learning platform on programming skills and the attitudes towards teaching programming. *Journal of Education and Training Studies,* 5(11), 1-12.

ALTONJI, J. G., & ZHONG, L. (2021). The labor market returns to advanced degrees. *Journal of Labor Economics,* 39(2), 303-360.

ALTONJI, J. G., ARCIDIACONO, P., & MAUREL, A. (2016). The analysis of field choice in college and graduate school: Determinants and wage effects. In *Handbook of the Economics of Education* (Vol. 5, pp. 305-396).

ALTONJI, J. G., BLOM, E., & MEGHIR, C. (2012). Heterogeneity in human capital investments: High school curriculum, college major, and careers. *Annual Review of Economics,* 4(1), 185-223.

ANGELI, C., VOOGT, J., FLUCK, A., WEBB, M., COX, M., MALYN-SMITH, J., & ZAGAMI, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society,* 19(3), 47-57.

ARFÉ, B., VARDANEGA, T., & RONCONI, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education,* 148, 103807.

ARKANSAS DEPARTMENT OF EDUCATION. (2019). Cracking the code: How Arkansas became a national leader in computer science and computing. Retrieved from *https://dese.ade.arkansas.gov/Files/20201217110343_ARKidsCanCode%202020%20Report.pdf*

AYEBI-ARTHUR, K. (2011). Interest in ICT studies and careers: Perspectives of senior high school female students in three districts in the Central Region of Ghana. *Ghana Journal of Education and Teaching,* 12.

BELFIELD, C., BRITTON, J., BUSCHA, F., DEARDEN, L., DICKSON, M., VAN DER ERVE, L., ... & ZHU, Y. (2019). The impact of undergraduate degrees on early-career earnings: Research report.

BELL, T., & VAHRENHOLD, J. (2018). CS unplugged—how is it used, and does it work? *Adventures between lower bounds and higher altitudes* (pp. 497-521).

BEST COLLEGE REVIEWS. (2021). Summer technology programs and bootcamps for high school students. Retrieved from *https://www.https://www.bestcollegereviews.org/features/pre-college-summer-technology-programs-high-school-students/*

BRILLIANT LABS. (N.D.). Retrieved from *https://www.brilliantlabs.ca/*

BRITISH COMPUTING SOCIETY. (N.D.) BCS subject knowledge enhancement course in computer programming. Retrieved from *https://www.bcs.org/get-qualified/training-and-scholarships-for-teachers/bcs-subject-knowledge-enhancement-course-in-computer-programming/*

BROGI, L. (2019, JUNE 25). "Girls Code It Better: a new policy for fighting gender stereotypes in Italy [Blog post]." https://leapblogsite.wordpress.com/2019/06/25/girls-code-it-better-a- new-policy-for-fighting-gender-stereotypes-in-italy/

BROWN, E., & BROWN, R. (2020, MARCH 4). The effect of Advanced Placement computer science course taking on college enrollment. West Coast Analytics. http://www.westcoastanalytics.com/uploads/6/9/6/7/69675515/longitudinal_study_-_combined_report_final_3_10_20__jgq_.pdf

BROWN, R. S. & BROWN, E. A. (2019). Estimating the effect of a teacher training program on Advanced Placement outcomes. *International Journal of Computer Science Education in Schools*, 2(4), 3-21.

BUITRAGO FLÓREZ, F., CASALLAS, R., HERNÁNDEZ, M., REYES, A., RESTREPO, S., & DANIES, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834-860.

CAMPBELL, L. O., & HELLER, S. (2019). Building computational thinking: Design and making in teacher education. In *Recruiting, Preparing, and Retaining STEM Teachers for a Global Generation* (pp. 163-189).

CARNEVALE, A. P., CHEAH, B., & STROHL, J. (2013). Not all college degrees are created equal. College, Majors, Unemployment, and Earnings. *Hard Times*. Retrieved from https://repository.library.georgetown.edu/bitstream/handle/10822/559308/Unemployment.Final.update1.pdf?sequence=1

CHERYAN, S., MASTER, A., & MELTZOFF, A. N. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology*, 6, 49.

CHETTY, R., FRIEDMAN, J. N., & ROCKOFF, J. E. (2014). Measuring the impacts of teachers I: Evaluating bias in teacher value-added estimates. *American Economic Review*, 104(9), 2593-2632.

CHETTY, R., FRIEDMAN, J. N., & ROCKOFF, J. E. (2014). Measuring the impacts of teachers II: Teacher value-added and student outcomes in adulthood. *American Economic Review*, 104(9), 2633-79.

CHOI, J., AN, S., & LEE, Y. (2015). Computing education in Korea—current issues and endeavors. *ACM Transactions on Computing Education (TOCE)*, 15(2), 1-22.

CICEK, F. G., & TASPINAR, M. (2016). Laboratory control system's effects on student achievement and attitudes. *Eurasian Journal of Educational Research*, 16(64).

COBO, C., & MONTALDO, M. (2018). Plan Ceibal in Uruguay: How do you educate in learning to decode the unknown? *UNESCO International Bureau of Education*. Retrieved from https://digital.fundacionceibal.edu.uy/jspui/bitstream/123456789/269/1/265994E.pdf

CODE.ORG & CSTA. (2018). State of computer science education: Policy and implementation. Retrieved from https://code.org/files/2018_state_of_cs.pdf

CODE.ORG, CSTA, ECEP. (2019). State of computer science education: Policy and implementation. Retrieved from https://code.org/files/2018_state_of_cs.pdf

CODE.ORG. (2017). *Recommendations for states developing computer science teacher pathways*. Retrieved from https://code.org/files/TeacherPathwayRecommendations.pdf

CODING THAILAND (N.D.). แพลตฟอร์มส่งเสริมการเรียนรู้ เกี่ยวกับวิชาวิทยาการคำนวณสำหรับเยาวชนไทยทุกคน. Retrieved from https://codingthailand.org/

COMPUTER SCIENCE ON THE RISE IN ARKANSAS SCHOOLS, GOV. DRAFTS LEGISLATION TO MAKE IT A REQUIREMENT FOR GRADUATION. (2020, NOVEMBER 18). 5NEWS. Retrieved from https://www.5newsonline.com/article/news/education/computer-science-on-the-rise-in- arkansas-schools-gov-drafts-legislation-to-make-it-a-requirement-for-graduation/527- 0261e975-cf05-4385-b752-e60a125a0965

CS FOR ALL. (N.D). K-12 computer science in Arkansas. Retrieved from https://www.csforall.org/projects_and_programs/k-12-computer-science-in-arkansas/

CURZON, P., MCOWAN, P. W., CUTTS, Q. I., & BELL, T. (2009, JULY). Enthusing & inspiring with reusable kinaesthetic activities. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education* (pp. 94-98).

CUSSÓ-CALABUIG, R., FARRAN, X. C., & BOSCH-CAPBLANCH, X. (2018). Effects of intensive use of computers in secondary school on gender differences in attitudes towards ICT: A systematic review. *Education and Information Technologies*, 23(5), 2111-2139.

CUTTS, Q., ROBERTSON, J., DONALDSON, P., & O'DONNELL, L. (2017). An evaluation of a professional learning network for computer science teachers. *Computer Science Education*, 27(1), 30-53.

DELYSER, L. A., J. GOODE, M. GUZDIAL, Y. KAFAI, AND A. YADAV. (2018) *Priming the computer science teacher pump: Integrating computer science education into schools of education*. CSforALL. Retrieved from: https://drive.google.com/file/d/1DXgpLjl_k87TVpQocLusfdjnYySIgIjT/view.

DENNING, P. J., COMER, D. E., GRIES, D., MULDER, M. C., TUCKER, A., TURNER, A. J., & YOUNG, P. R. (1989). Computing as a discipline. *Computer*, 22(2), 63-70.

DEPARTMENT FOR EDUCATION. (2019). *Key stage 4 performance*, 2019. Retrieved from https://www.gov.uk/government/statistics/key-stage-4-performance-2019-provisional

DEVOL, R. (2016). Career technical education: Reducing wage inequality and sustaining California's innovation-based economy. *Milken Institute*.

DICKENS, J. (2016, JANUARY 15). Government spends £3m in scramble to get 400 'master' computing teachers. *Schools Week*. Retrieved from https://schoolsweek.co.uk/3m-on-and-where-are-all-the-master-teachers/

DIEKMAN, A. B., BROWN, E. R., JOHNSTON, A. M., & CLARK, E. K. (2010). Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. *Psychological science*, 21(8), 1051-1057.

DJAMBONG, T., & FREIMAN, V. (2016). Task-based assessment of students' computational thinking skills developed through visual programming or tangible coding environments. *International Association for Development of the Information Society.*

DONER, R. F., & SCHNEIDER, B. R. (2000). Business associations and economic development: Why some associations contribute more than others. *Business and politics*, 2(3), 261-288.

DOUCETTE, K. (2016, MAY 06). Racing 'beebots' to teach kids coding in Nova Scotia. CBC News. Retrieved March 02, 2021, from https://www.cbc.ca/news/canada/nova-scotia/beebots-mcneil-caseycoding-programming-education-computer-1.3571568

DOWNES, T., & LOOKER, D. (2011). Factors that influence students' plans to take computing and information technology subjects in senior secondary school. *Computer Science Education*, 21(2), 175-199.

DU, J., & WIMMER, H. (2019). Hour of Code: A study of gender differences in computing. *Information Systems Education Journal*, 17(4), 91.

EDUTECH THAILAND. (2019). *Google Coding Teacher 2019*. Retrieved from https://www.edutechthailand.com/coding-teacher

ELLIS, E. (2021, FEBRUARY 1). Puzzle game provides missing piece in teaching children computer code. *Daily Maverick*. Retrieved from https://www.dailymaverick.co.za/article/2021-02-01-puzzle-game-provides-missing-piece-in-teaching-children-computer-code/

EUROPEAN COMMISSION. (2018). Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions on the Digital Education Action Plan. Brussels, Belgium. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52018DC0022&from=EN

EUROPEAN COMMISSION. (2009). *Gender differences in educational outcomes: A study on the measures taken and the current situation in Europe Italy*. Retrieved from https://annazavaritt.blog.ilsole24ore.com/wp-content/uploads/sites/54/files/euridice-educational-gender-issue-italy.pdf

FILMER, D., LANGTHALER, M., STEHRER, R., & VOGEL, T. (2018). *Learning to realize education's promise.* The World Bank. https://www.worldbank.org/en/publication/wdr2018

FONG, M. W. (2009). Technology leapfrogging for developing countries. In *Encyclopedia of Information Science and Technology, Second Edition* (pp. 3707-3713). IGI Global.

FRAILLON, J., AINLEY, J., SCHULZ, W., FRIEDMAN, T., & DUCKWORTH, D. (2020). *Preparing for life in a digital world: IEA international computer and information literacy study 2018 international report* (p. 297). IEA.

GANNON, S., & BUTEAU, C. (2018, APRIL). Integration of computational thinking in Canadian provinces [Scholarly project]. Retrieved from http://ctmath.ca/wpcontent/uploads/2018/10/Symposium_CanadaMap_Gannon-Buteau.pdf

GIACALONE, S. (2019). Che cos'è il pensiero computazionale? *Pearson*. Retrieved from https://it.pearson.com/docenti/primaria/classe-dinamica/che-cosa-pensiero-computazionale.html

GIRLS CODE IT BETTER. (N.D.). *Il progetto*. Retrieved from https://girlscodeitbetter.it/

GIRLSMARTS4TECH (N.D.). *Computer science workshops for girls*. Retrieved March 02, 2021, from https://www.cs.ubc.ca/girlsmarts4tech/

GLOCKER, D., & STORCK, J. (2014). Risks and returns to educational fields–A financial asset approach to vocational and academic education. *Economics of Education Review*, 42, 109-129.

GOODE, J., MARGOLIS, J., & CHAPMAN, G. (2014, MARCH). Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 493-498).

GOOGLE & GALLUP (2016). *Trends in the state of computer science in US K-12 schools*. Retrieved from https://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf

GOOGLE & GALLUP. (2015). *Images of computer science: Perceptions among students, parents and educators in the US*. Retrieved from https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf

GOOGLE & GALLUP (2020). *Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools*. Retrieved from: https://services.google.com/fh/files/misc/computer-science-education-in-us-k12schools-2020-report.pdf

GROVER, S., JACKIW, N., & LUNDH, P. (2019). Concepts before coding: non-programming interactives to advance learning of introductory programming concepts in middle school. *Computer Science Education, 29* (2-3), 106-135. https://doi.org/10.1080/08993408.2019.1568955

GÜLBAHAR, Y., & KALELIOĞLU, F. (2017, NOVEMBER). Competencies of high school teachers and training needs for computer science education. In *Proceedings of the 6th Computer Science Education Research Conference* (pp. 26-31).

GUZDIAL, M., ERICSON, B., MCKLIN, T., & ENGELMAN, S. (2014). Georgia computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education (TOCE)*, 14(2), 1-29.

HANSON, G., & SLAUGHTER, M. (2016). High-skilled immigration and the rise of STEM occupations in US employment (No. w22623; p. w22623).

HARRIS, C. (2018). Computational thinking unplugged: Comparing the impact on confidence and competence from analog and digital resources in computer science professional development for elementary teachers. (Education Doctoral Paper). St. John Fisher College.

HARRISON, S. (2019, OCT. 1). Five years of tech diversity reports—and little progress. *Wired Magazine.* Retrieved from https://www.wired.com/story/five-years-tech-diversity-reports-little-progress

HASTINGS, J. S., NEILSON, C. A., & ZIMMERMAN, S. D. (2013). Are some degrees worth more than others? Evidence from college admission cutoffs in Chile (No. w19241). National Bureau of Economic Research.

HEINTZ, F., MANNILA, L., & FARNQVIST, T. (2016). A review of models for introducing computational thinking, computer science and computing in K–12 education. Linköping University.

HENDRICKSON, K. (2019). Computer science climbs to 4th most popular STEM major for college-bound students. Retrieved from *https://codeorg.medium.com/computer-science-climbs-to-4th-most-popular-stem-major-for-college-bound-students-773ce681b96c*

HILL, H. C. (2009). Fixing teacher professional development. *Phi Delta Kappan,* 90(7), 470-476. *https://oecdedutoday.com/computer-science-and-pisa-2021/*

HUBWIESER, P., MAGENHEIM, J., MÜHLING, A., & RUF, A. (2013, AUGUST). Towards a conceptualization of pedagogical content knowledge for computer science. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 1-8).

LEE, K. (2019). Economics of technological leapfrogging. Inclusive and Sustainable Working Paper Series. *UNIDO*. Retrieved from *https://www.unido.org/api/opentext/documents/download/16414872/unido-file-16414872*

IOANNOU, L. (2018, JUN. 20). Silicon Valley's Achilles' heel threatens to topple its supremacy in innovation. CNBC. Retrieved from *https://www.cnbc.com/2018/06/20/silicon-valleys-diversity-problem-is-its-achilles-heel.html*

JARA, I., HEPP, P., & RODRIGUEZ, J. (2018). Policies and practices for teaching computer science in Latin America. Retrieved from *https://www.researchgate.net/publication/335611704_Policies_and_Practices_for_Teaching_Computer_Science_in_Latin_America*

JOHNSTON, J., & KSOLL, C. (2017). Effectiveness of interactive satellite-transmitted instruction: Experimental evidence from Ghanaian primary schools. Center for Education Policy Analysis (CEPA) working paper, (17-08).

JORGENSON, D. W., & VU, K. M. (2016). The ICT revolution, world economic growth, and policy issues. *Telecommunications Policy,* 40(5), 383-397.

K-12 COMPUTER SCIENCE FRAMEWORK STEERING COMMITTEE. (2016). K-12 computer science framework. ACM. Retrieved from https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

KALELIOĞLU, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior,* 52, 200-210.

KIRKEBOEN, L. J., LEUVEN, E., & MOGSTAD, M. (2016). Field of study, earnings, and self-selection. *The Quarterly Journal of Economics,* 131(3), 1057-1111.

KOZŁOWSKI, W., & MATCZAK, E. (2016). Aspiracje zawodowe rodziców wobec synów i córek. *EDUKACJA Quarterly,* 138(3).

KPMG (2016). British Columbia technology report card: Scaling up BC's tech ecosystem. https://wearebctech.com/2016-bc-technology-report-card/

LANG, C., CRAIG, A., & EGAN, M. (2016). The importance of outreach programs to unblock the pipeline and broaden diversity in ICT education. *International Journal of Information and Communication Technology Education* (IJICTE), 12(1), 38-49.

LANG, K., GALANOS, R., GOODE, J., SEEHORN, D., TREES, F., PHILLIPS, P., & STEPHENSON, C. (2013). Bugs in the system: Computer science teacher certification in the US. The Computer Science Teachers Association and The Association for Computing Machinery.

LEMIEUX, T. (2014). Occupations, fields of study and returns to education. *Canadian Journal of Economics/Revue canadienne d'économique,* 47(4), 1047-1077.

LIU, J., LIN, C. H., HASSON, E. P., & BARNETT, Z. D. (2011, MARCH). Introducing computer science to K-12 through a summer computing workshop for teachers. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 389-394).

LIVINGSTONE, S., HADDON, L., GÖRZIG, A., & ÓLAFSSON, K. (2011). Risks and safety on the internet: The perspective of European children: Full findings and policy implications from the EU Kids Online survey of 9-16-year-olds and their parents in 25 countries. London School of Economics.

LOCKWOOD, B., & CORNELL, R. (2013). School ICT infrastructure requirements for teaching computing. A Computing at School (CAS) Whitepaper.

MAHATAI (N.D.). Retrieved from *https://www.google.com/search?q=Redemptorist+Foundation+for+People+with+Disabilities+http%3A%2F%2Fwww.mahatai.org%2F&oq=Redemptorist+Foundation+for+People+with+Disabilities+http%3A%2F%2Fwww.mahatai.org%2F&aqs=chrome..69i57.2295j0j7&sourceid=chrome&ie=UTF-8*

MAKING CODING FUN FOR THAILAND'S YOUNG. (2018, FEBRUARY 28). Bangkok Post. Retrieved from *https://www.bangkokpost.com/tech/1419663/making-coding-fun-for-thailands-young*

MARGOLIS, J., & FISHER, A. (1997). Geek mythology and attracting undergraduate women to computer science. Women in Engineering ProActive Network.

MARYSKA, M., DOUCEK, P., & KUNSTOVA, R. (2012). The importance of ICT sector and ICT university education for the economic development. Procedia-Social and Behavioral Sciences, 55, 1060-1068.

MARZOLLA, M., & MIRANDOLA, R. (2019, SEPTEMBER). Gender balance in computer science and engineering in Italian universities. In *Proceedings of the 13th European Conference on Software Architecture-Volume 2* (pp. 82-87).

MCCARTNEY, R., BOUSTEDT, J., ECKERDAL, A., SANDERS, K., & ZANDER, C. (2017, MARCH). Folk pedagogy and the geek gene: geekiness quotient. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 405-410).

MCCROCKLIN, S. (2021). Mobile penetration in South Africa. GeoPoll. Retrieved from https://www.geopoll.com/blog/mobile-penetration-south-africa/

MICROSOFT CELEBRATES HOUR OF CODE TO BUILD FUTURE READY GENERATIONS IN ASIA. (2017, DECEMBER 4). Microsoft Asia News Center. Retrieved from https://news.microsoft.com/apac/2017/12/04/microsoft-celebrates-hour-of-code-to-build-future-ready-generations-in-asia/

MICROSOFT. (N.D.). Microsoft TEALS implementation guide (2021-22 School Year). Retrieved from https://www.tealsk12.org/wp-content/uploads/2020/10/2021-22-TEALS-Implementation-Guide.pdf

NAGER, A., & ATKINSON, R. D. (2016). The case for improving US computer science education. Available at SSRN: http://dx.doi.org/10.2139/ssrn.3066335

NATIONAL ACADEMIES OF SCIENCES, ENGINEERING, AND MEDICINE. (2018). Assessing and responding to the growth of computer science undergraduate enrollments.

NISHIDA, T., KANEMUNE, S., IDOSAKA, Y., NAMIKI, M., BELL, T., & KUNO, Y. (2009). A CS unplugged design pattern. *ACM SIGCSE Bulletin,* 41(1), 231-235.

NIX, N. (2017, SEPTEMBER 18). How Arkansas is teaming up with teachers, Facebook & other tech titans to rethink computer science education. *The 74*. Retrieved from https://www.the74million.org/article/how-arkansas-is-teaming-up-with-teachers-facebook-other-tech-titans-to-rethink-computer-science-education/

OECD. (2013). Trust in government, policy effectiveness and the governance agenda. *Government at a Glance, 2013*. Retrieved from https://www.oecd-ilibrary.org/docserver/gov_glance-2013-6-en.pdf?

OECD. (2021). Employment Outlook 2021: A once-in-a-lifetime opportunity to build a better world of work. Retrieved from https://www.oecd-ilibrary.org/sites/5a700c4b-en/index.html?itemId=/content/publication/5a700c4b-en

ORIZZONTESCUOLA. (2019, MARCH 14). Coding Sarà Obbligatorio Nelle Scuole D'infanzia E Primaria, Accolta Mozione Aprea [Blog post]. Retrieved from https://www.edscuola.eu/wordpress/?p=113070

ORTON, T. (2018, NOVEMBER 16). Kids' coding programs target B.C. tech talent gaps. BIV. Retrieved March 02, 2021, from https://biv.com/article/2018/11/kids-coding-programs-target-bc-techtalent-gaps

PANSKYI, T., ROWINSKA, Z., & BIEDRON, S. (2019). Out-of-school assistance in the teaching of visual creative programming in the game-based environment–Case study: Poland. *Thinking Skills and Creativity,* 34, 100593.

PATITSAS, E., BERLIN, J., CRAIG, M., & EASTERBROOK, S. (2019). Evidence that computer science grades are not bimodal. *Communications of the ACM,* 63(1), 91-98.

PHILLIPS, R. S., & BROOKS, B. P. (2017). The hour of code: Impact on attitudes towards and self-efficacy with computer science. Code.org. Retrieved from https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

RAMAN, R., VENKATASUBRAMANIAN, S., ACHUTHAN, K., & NEDUNGADI, P. (2015). Computer science (CS) education in Indian schools: Situation analysis using Darmstadt model. *ACM Transactions on Computing Education* (TOCE), 15(2), 1-36.

RASPBERRY PI FOUNDATION (N.D.). Retrieved from https://www.raspberrypi.org/

REPENNING, A. (1993, MAY). Agentsheets: a tool for building domain-oriented visual programming environments. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 142-143).

RESNICK, M., MALONEY, J., MONROY-HERNÁNDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAM, E., SILVER, E., SILVERMAN, E., & KAFAI, Y. (2009). Scratch: programming for all. *Communications of the ACM,* 52(11), 60-67.

RIVKIN, S. G., HANUSHEK, E. A., & KAIN, J. F. (2005). Teachers, schools, and academic achievement. *Econometrica,* 73(2), 417-458.

ROYAL SOCIETY. (2017). After the reboot: Computing education in UK schools. Retrieved from: https://royalsociety.org/~/media/policy/projects/computing-education/computing-education-report.pdf

SALEHI, S., K. D. WANG, R. TOORAWA, AND C. WIEMAN. CAN MAJORING IN COMPUTER SCIENCE IMPROVE GENERAL PROBLEM-SOLVING SKILLS? (2020). SIGCSE '20: Proceedings of the 51st ACM Technical Symposium on Computer Science Education (pp. 156-161).

SAELI, M., PERRENET, J., JOCHEMS, W. M., & ZWANEVELD, B. (2011). Teaching programming in Secondary school: A pedagogical content knowledge perspective. *Informatics in education,* 10(1), 73-88.

SÁEZ-LÓPEZ, J. M., ROMÁN-GONZÁLEZ, M., & VÁZQUEZ-CANO, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year casestudy using "Scratch" in five schools. *Computers & Education,* 97, 129-141.

SALEHI, S., WANG, K. D., TOORAWA, R., & WIEMAN, C. (2020, FEBRUARY). Can majoring in computer science improve general problem-solving skills? In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 156-161).

SÁNCHEZ, J., & SALINAS, A. (2008). ICT & learning in Chilean schools: Lessons learned. *Computers & Education,* 51(4), 1621-1633.

SANTIAGO, P., FISZBEIN, A., JARAMILLO, S. G., & RADINGER, T. (2017). *OECD reviews of school resources: Chile 2017.* OECD Publishing. 2, rue Andre Pascal, F-75775 Paris Cedex 16, France.

SAX, L. J., LEHMAN, K. J., JACOBS, J. A., KANNY, M. A., LIM, G., MONJE-PAULSON, L., & ZIMMERMAN, H. B. (2017). Anatomy of an enduring gender gap: The evolution of women's participation in computer science. *The Journal of Higher Education,* 88(2), 258-293.

SCHLEICHER & PARTOVI. (2019). *Computer science and PISA 2021.* OECD. Retrieved from https://oecdedutoday.com/computer-science-and-pisa-2021/

SENTANCE, S., & CSIZMADIA, A. (2017, MARCH). Professional recognition matters: Certification for in-service computer science teachers. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 537-542).

SEVERIN, E. (2016). *Building and sustaining national ICT/education agencies.* World Bank. Retrieved from https://openknowledge.worldbank.org/bitstream/handle/10986/26264/113221-NWP-PUBLIC-ADD-SERIES-Agencies-Chile-Enlaces-SABER-ICTno07.pdf?sequence=1&isAllowed=y

SILCOFF, S. (2016, JANUARY 17). B.C. to add computer coding to school curriculum. *The Globe and Mail.* Retrieved from https://www.theglobeandmail.com/technology/bc-government-adds-computer-coding-to-school-curriculum/article28234097/

SNOWDON, K. (2019, JULY 2). First 23 computing hubs in England announced. *Schools Week.* https://schoolsweek.co.uk/computing-hubs-england-announced-ncce/

SO, H. J., JONG, M. S. Y., & LIU, C. C. (2020). Computational thinking education in the Asian Pacific region. *The Asia-Pacific Education Researcher,* 29, 1-8.

STATE OF ARKANSAS. (2018). State Science and Technology Plan. Retrieved from https://www.arkansasedc.com/docs/default-source/s-t/state_st_plan.pdf?sfvrsn=c3c7704a_2

STEM LEARNING. (N.D.). Retrieved from https://www.stem.org.uk/

SULLIVAN, A., & BERS, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education,* 23(3), 691-702.

SYSŁO, M. M. (2020). The new computer science curriculum in Poland–challenges and solutions. Warsaw School of Computer Science. Retrieved from *http://ceur-ws.org/Vol-2755/paper10.pdf*

SYSŁO, M. M., & KWIATKOWSKA, A. B. (2015, SEPTEMBER). Introducing a new computer science curriculum for all school levels in Poland. In *International conference on informatics in Schools: Situation, evolution, and perspectives* (pp. 141-154). Springer, Cham.

SZYMAŃSKI, L. (2016, JUNE 1). Zalewska: We are starting to teach the basics of programming in Polish schools. *Science in Poland*. Retrieved from *https://scienceinpoland.pap.pl/en/news/news%2C407906%2Czalewska-we-are-starting-to-teach-the-basics-of-programming-in-polish-schools.html*

TALK BUSINESS & POLITICS. (2020, JANUARY 9). Gov. Hutchinson, education officials tout five years of computer coding program. Retrieved from *https://talkbusiness.net/2020/01/gov- hutchinson-education-officials-tout-five-years-of-computer-coding-program/*

TAUB, R., BEN-ARI, M., & ARMONI, M. (2009). The effect of CS unplugged on middle-school students' views of CS. *ACM SIGCSE Bulletin,* 41(3), 99-103.

TEACH COMPUTING. (2020). New NCCE report shows impact of work to support thousands of teachers. Retrieved from *https://blog.teachcomputing.org/new-ncce-report-shows-impact-3/*

TEW, A. E. (2010). Assessing fundamental introductory computing concept knowledge in a language independent manner (Doctoral dissertation). Georgia Institute of Technology.

THONGNAB, T. (2019, JULY 5). Microsoft continues commitment to drive digital skills for Thais in technology-driven era. Microsoft. Retrieved from https://news.microsoft.com/th-th/2019/07/05/philanthropies19_en/

TUCKER, A. (2003). *A model curriculum for k-12 computer science: Final report of the ACM k-12 task force curriculum committee.* ACM.

URUGUAY PRESIDENCY. (2019, SEPTEMBER 19). ANEP realiza Encuesta Nacional de Relevamiento sobre Pensamiento Computacional en Educación. https://www.presidencia.gub.uy/comunicacion/comunicacionnoticias/rel evamiento

VARMA, R. (2007). Women in computing: The role of geek culture. *Science as Culture,* 16(4), 359-376.

VASQUEZ, E. A., BOTTAMEDI, J., & BRIZUELA, M. L. (2019). Pensamiento computacional en el aula: el desafío en los sistemas educativos de Latinoamérica. *Revista Interuniversitaria de Investigación en Tecnología Educativa*.

VEGAS, E. (2020). School closures, government responses, and learning inequality around the world during COVID-19. Brookings. Retrieved from https://www.brookings.edu/research/school-closures-government-responses-and-learning-inequality-around-the-world-during-covid-19/

WATSON-FISHER, J. (2019, SEPTEMBER). Arkansas governor touts high school computer science. *Government Technology*. Retrieved from https://www.govtech.com/education/arkansas- governor-touts-high-school-computer-science.html

WEBB, M., DAVIS, N., BELL, T., KATZ, Y. J., REYNOLDS, N., CHAMBERS, D. P., & SYSŁO, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies,* 22(2), 445-468.

WEINTROP, D., & WILENSKY, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education* (TOCE), 18(1), 1-25.

WESTON, M., & BIIN, D. (2015). The ancestor project: Aboriginal computer education through storytelling. In *E-Learning Systems, Environments and Approaches* (pp. 287-304).

WING, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* 366(1881), 3717-3725.

BROOKINGS

1775 MASSACHUSETTS AVE NW
WASHINGTON, D.C. 20036
https://www.brookings.edu/