Computer Science Courses as a Graduation Requirement at the State and National Level: A Policy Brief

This manuscript has been peer-reviewed, accepted, and endorsed by the International Council of Professors of Educational Leadership (ICPEL) as a significant contribution to the scholarship and practice of school administration and K-12 education.

Eric Bass
Deuel (South Dakota) School School District

David De Jong
University of South Dakota

Computer science education has seen a large national push to be included in the K-12 curriculum (Fluck, Webb, Cox, Angeli, Malyn-Smith, Voogt, & Zagami, 2016; Theresa Avancena & Nishihara, 2014; Veletsianos, Beth, Lin, & Russell, 2016; Yadav, Gretter, Hambrusch, & Sands, 2016). Students are leaving schools and entering an economy driven by technology and by a need to understand how to use and develop that technology (Armoni & Gal-Ezer, 2014; Bozick, Srinivasan, & Gottfried, 2017; Burke, 2016; Stuikys, Burbaite, Blazauskas, Barisas, & Binkis, 2017). As the State of South Dakota begins the process of reexamining their graduation requirements, the State Department of Education should consider the importance of computer science education in the curriculum as a standalone graduation requirement.

Three aspects of computer science call for its inclusion in the curriculum. These three aspects include introducing computer science at elementary or middle school grade levels, ensuring the computer science courses contain skills that can be transferred to other subjects, and providing the necessary skills to meet the changing demands of today's society. Including computer science in the curriculum will force high schools in South Dakota to examine their curriculum to identify where changes can be made to make room for the necessary computer science course or courses. The inclusion of computer science in the South Dakota graduation requirements would also force the State Department of Education to create professional development opportunities, possibly with the help of higher education or local technology companies, for teachers to develop better pedagogical strategies. The policy brief that follows outlines the argument, expands on the important aspects, and describes the implications of the inclusion of computer science as a standalone requirement in the South Dakota high school graduation requirements.

High school graduation requirements have largely remained unchanged since the time of their original inceptions (Silva, White, & Toch, 2015, p. 8). Students in high schools have been required to take multiple classes in the areas of mathematics, science, social studies, and English with various other half or full credit requirements in elective areas. The original and current intent was to provide high school students with the necessary skills and knowledge to be successful in postsecondary colleges and universities. The graduation requirements currently in place have failed to address the rising influence of technology on modern high school students. As technology has become a more prevalent part of the daily lives of students both in and out of classrooms, schools have been slow to adopt programs that help students understand how to use that technology.

With the rise of technology and computer-based jobs, schools need to a hard look at how they are preparing students to obtain the necessary skills needed to train students to enter these fields. Technology in schools has often been limited to providing students with desktop computers or laptops that they can use to help complete schoolwork. The classes designed to teach the application or understanding of how technology works are virtually non-existent (Fluck, et al., 2016, pp. 39-40). Students in some school districts may have elective computer science classes available to them but are often limited to a single technological subject or skill. Schools will point to a variety of reasons for not adding more computer science classes to their curriculum with the lack of a state requirement being a part of that reason.

States have attempted to add computer science requirements to the high school graduation requirements, but often include them in the Career and Technical Education (CTE) requirement (Nager & Atkinson, 2016). The problem with including computer science courses in the CTE requirement is that they are often paired as an *or* with a world language or other CTE program. In these cases, a student can meet graduation requirements by taking a foreign language credit or credits in other CTE programs and never have to take a computer science course. As school districts and the South Dakota Department of Education begin to review graduation requirements, computer science programs need to be considered as a standalone requirement to meet the rising demand for skills in those areas.

Approach and Results

There are three aspects within the teaching of computer science to justify the inclusion of a computer science requirement in the South Dakota graduation requirements.

Aspect 1. Computer science can be introduced early on in school and built upon.

Traditional K-12 education has been built on the premise of an introduction of a specific topic or subject and the subsequent expansion of that knowledge in the following years. Through online coding and programming programs such as code.org, Alice, and Scratch, schools can implement computer science skills at an early age in school. According to Theresa Avancena and Nishihara (2014), "Success in advanced computing courses is still often associated with success in introductory programs" (p. 139). With a school-wide initiative to identify and implement coding or programming lessons and units into the regular curriculum, students can be exposed to the necessary fundamental skills that will allow them to become successful in future high school computers science courses. Much like reading and math skills at the elementary level, computer science skills can be worked on and developed alongside the more traditional topics. Also, as with math and reading skills, computer science skills will not come naturally or quickly to everyone and encourage students to work through the problems to become successful. According to Fluck et al. (2016), “Naturally, in the learning process, some early efforts result in less than complete success – or failure. Handled well with the classroom, this can be an opportunity to build resilience” (pp. 42-43). Resilience in learning is necessary for any student to become successful in their educational goals and with the right approach to computer science programs, students can learn to evaluate where the program is going wrong and how to correct that issue (Aburn, Gott, & Hoare, 2016).

Aspect 2. Computer science contains skills that can be transferred to other subjects.

Computer science courses can teach or reinforce many of the necessary skills that students learn in other areas of their education (Nager & Atkinson, 2016). In mathematics courses such as algebra and geometry, students are required to follow particular rules, to apply different strategies, and to complete problems in sequential order. Science courses apply the scientific method for discovery which includes the observation, hypothesizing, testing, and analyzing of data in order to come up with a working conclusion on a specific problem. English courses often ask students to review pieces of literature and specify the thought the author may have had during the writing of the piece. The ability of students to narrow their focus, to think, and to communicate precisely are necessary skills to be successful in computer science courses. For example, Fluck et al. (2016) declare, “Coding is about thinking. Putting process into a particular code requires precision. Therefore a child skilled at coding, may be transference, be more precise in their thought and have greater capacity to communicate” (p. 42). Burke (2016) also insists, “Programming is increasingly being recognized as not just as economic skill set but a potential pathway by which to get youth more engaged in the workings of the web-based media that surround them” (p. 211). When we can provide students with the ability to effectively communicate their thoughts and ideas, explore and interact with the world around them, we have given them the tools necessary to be successful once they have graduated from high school.

Aspect 3. Computer science can provide the necessary skills to meet the changing technology demands of modern society.

Ernst and Clark (2012) state, "Computer science literacy has become an important aspect to learning and living in an information age" (p. 40). It is difficult to argue that technology is not intruding on all aspects of our daily lives. We are using and relying on technology at an ever-increasing rate, and it is necessary for users to understand how technology can be used effectively and safely. Due to this intrusion and constant flow of information through these devices, students need to build an understanding of how to properly verify information and determine which is reliable and which is not. Fluck et al. (2016) assert, “Technology can change our sense of ethical behavior. Rather than being oppressed by innovation shock, a society equipped with its own creative proponent of new ideas is more likely to sift them and control their impact” (p. 41). Computer science programs can be geared to help middle and high school students understand how to determine which information is accurate and how that information can be used to form opinions on topics and hot button issues that seem to pop up in the news and social media. Through proper education, we can help students become producers and active creators rather than just passive consumer of technology (Webb, Davis, Bell, Katz, Reynolds, Chambers, & Sylo, 2017).

Conclusions

Conclusions can be drawn at the state level and the national level.

Computer Science Throughout South Dakota

Since computer science courses fall under the Career and Technical Education (CTE) umbrella in South Dakota, many high school students across the state are taking non-computer science courses to meet the requirement for graduation. By taking computer science out of the CTE requirement and making it a stand-alone requirement, schools can provide the students the opportunity to explore the world of computer science and opportunities it affords them in the future. The addition of a computer science credit, whether a half or full, to the South Dakota high school graduation standards has long reaching positive implications for both students and the local and state economies in which these students live.

Computer Science Throughout the United States

Throughout the United States, the correlation between mathematics, science, and computer science allows cross-curricular connections to be made in which students can begin to integrate problem-solving and the scientific method to address real-world problems (Bozick, Srinvasan, & Gottfried, 2017). Being able to think in abstraction, logically, and algorithmically about various bits of information as well as analyze, evaluate, and apply that information to a problem are all necessary for success in computer science as well as many other areas of the high school curriculum (Fluck et al., 2016). The inclusion of computer science in the required curriculum throughout the United States can begin to breakdown the silos; that tend to exist between subjects matters in modern high schools and help students think about the bigger picture of their education in general.

For the local economies throughout the United States, businesses can hire students coming out of school with a more advanced computer and problem-solving skills (Bozick, Srinvasan, & Gottfried, 2017). Unfortunately, as school districts try to add more elective computer science courses into their curriculum, the number of students taking a course in computer science has seen a drop in recent years. Ryoo, Margolis, Lee, Sandoval, and Goode (2013) report, “In fact, fewer students than ever are studying computer science in our high schools despite the increasing demand for computer scientists whose employment is projected to grow faster than any other occupation in America between 2008 and 2018” (p. 162). To further the point of job opportunities for students, Lockard and Wolf (2012) point out that the “US employment in computer science occupations is projected to grow by 22% between 2010 and 2020, with a growth rate of 32% for system software developers” (p. 102). These employment growth numbers are very encouraging for schools throughout the United States as they look for ways to encourage students to enroll in the computer sciences.

Implications and Recommendations

The addition of computer science as a standalone graduation requirement will require schools to do two things that will be necessary for the successful implementation of a computer science curriculum. For students to benefit from computer science as outlined throughout this policy brief, schools have to examine their high school curriculum and provide teachers with practical and meaningful professional development.

Schools Must Examine the Curriculum

Schools will have to make room for the new computer science requirement in an already packed school day. If a computer science course does not already exist, schools will have to examine where a course can be implemented. Administrators will need to identify what grade level they would be required to take the course as well as if courses will be offered beyond the required credit. Also, schools that do not have a current computer science course, the addition of the necessary course may cause other elective classes to be dropped altogether. If a current computer science course is offered, the changing of the class from an elective to a requirement may force additional sections to be offered as well as the reshuffling of staff to pick up the classes lost by the computer science instructor. Depending on the level of technology available to schools for a computer science course, a financial investment may be necessary to get the needed equipment to implement different aspects of a computer science course. With the tightening of budgets in schools across the United States, some schools may feel this investment may be too costly to make the necessary upgrades to provide the best possible opportunity. Schools in this situation, instead of looking at this as a burden, should look at it as an opportunity to collaborate with local businesses or organizations. In fact, the U.S. National Science Foundation (NSF) has funding available for school to help with computer science education programs. Through the involvement of the NSF, other computing companies and experts have become involved in computer science education (Veletsianos et al., 2016).

Schools Must Provide Computer Science Professional Development Opportunities for Computer Science Teachers

More importantly than the necessary class changes and technology needs, schools will need a teacher with the ability to properly teach computer science concepts and applications (Hubwieser et al., 2015). Often times, computer science teachers come from other curricular backgrounds and lack the necessary pedagogical skills or computer science knowledge to provide effective instruction (Yadav et al., 2016). Teacher training and certification through professional development is a vital need for a successful implementation and to keep up with new advances in the computer science field (Opfer & Pedder, 2011). Providing computer science teachers with the ability to attend various trainings, in-services, and summer workshops, computer science teachers can expand on what they know and how they can provide students with practical and engaging classroom opportunities. To help with this professional development, computer and technology companies have begun to provide software courses for computer science teachers. As Menekse (2015) points out, “Google recently funded four institutions to develop computer science teacher professional development workshops in the form of massive open online course” (p. 326). As more technology companies begin to work with schools to help develop computer science software, full programs, and possibly internships, the computer science experience will become more effective in providing the benefits mentioned earlier in the brief.

When computer science is added as a standalone graduation requirement, teacher preparation programs at colleges and universities will have to adjust to meet the rising demand of computer science teachers. Due to the ever-changing nature of the computer science and technology fields, higher education needs to provide potential computer science teachers with the necessary tools and strategies to understand how to keep up with the current trends and how they can turn those into useful classroom experiences. Universities are in better positions than school districts to work with and form agreements with technology and computing companies. These agreements could help further the understanding of computer science technologies, programs, and applications of future computer science teachers.

Implications for Educational Policy and Practice

The inclusion of computer science as a high school graduation requirement has implications for educational policy and practice. Any change to the educational policy at the state or national level would require changes for K-12 school leaders, higher education officials, and lawmakers responsible for educational funding. K-12 school leaders would be responsible for working with their state's department of education to develop research-based standards and benchmarks. School leaders must also revise their school schedules to provide an opportunity for computer science education at age-appropriate levels. Higher education officials would need to develop programs, courses, and licensure pathways for teachers of computer science. Higher education must also prepare school leaders for a new focus on computer science. Lawmakers responsible for funding would need to allocate the necessary funds at the K-20 educational levels. These stakeholders all play important roles and must work together to create educational policies which prepare students to use computer science to develop technologies for the future.

References

Aburn, G., Gott, M., & Hoare, K. (2016). What is resilience? An integrative review of the empirical literature. Journal of Advanced Nursing, 72(5), 980-1000.

Armoni, M., & Gal-Ezer, J. (2014). High school computers science paves the way for higher education: the Israeli case. Computer Science Education, 24(2/3), 101-122. doi:10.1080/08993408.2014.936655

Bozick, R., Srinivasan, S., & Gottfried, M. (2017). Do high school STEM courses prepare non-college bound youth for jobs in the STEM economy? Education Economics, 25(3), 234-250. doi:10.1080/09645292.2016.1234585

Burke, Q. (2016). Mind the metaphor: charting the rhetoric about introductory programming in K-12 schools. On The Horizon, 24(3), 210-220. doi: 10.1108/OTH-03-2016-0010

Ernst, J.V. & Clark, A.C. (2012). Fundamental Computer Science Conceptual Understandings for High School Students Using Original Computer Design. Journal of STEM Education: Innovations & Research, 13(5), 40-45.

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J. Voogt, J, & Zagami, J. (2016). Arguing for Computer Science in the School Curriculum. Journal of Educational Technology & Society, 19(3), 38-46.

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., ... & Jasute, E. (2015, July). A global snapshot of computer science education in K-12 schools. In Proceedings of the 2015 ITiCSE on working group reports (pp. 65-83). ACM.

Lockard, C.B., & Wolf, M. (2012). Occupational employment projections to 2020. Monthly Labor Review, 135(1), 84-108.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. Computer Science Education, 24(4), 325-350. doi: 10.1080/08993408.2015.1111645

Nager, A., & Atkinson, R. D. (2016). The case for improving US computer science education. Available at SSRN 3066335.

Opfer, W. R., & Pedder, D. (2011). Conceptualizing teacher professional learning. Review of Educational Research, 81, 376-407.

Ryoo, J.J., Margolis, J., Lee, C.H., Sandoval, C.M., & Goode, J. (2013). Democratizing computer science knowledge: transforming the face of computer science through public high school education. Learning, Media & Technology, 38(2), 161-181. doi:10.1080/17439884.2013.756514

Silva, E., White, T., & Toch, T. (2015). The Carnegie Unit: A Century-Old Standard in a Changing Education Landscape. The Carnegie Foundation for the Advancement of Teaching. Retrieved from: https://www.carnegiefoundation.org/resources/publications/carnegie-unit/

Stuikys, V., Burbaite, R., Blazauskas, T., Barisas, D., & Binkis, M. (2017). Model for Introducing STEM1 into High School Computer Science Education. International Journal of Engineering Education, 33(5), p 1684-1698.

Theresa Avancena, A. & Nishihara, A. (2014). Cognitive Correlates of Performance in Algorithms in a Computer Science Course for High School. Journal of Computers in Mathematics & Science Teaching, 33(2), 137-155.

Veletsianos, G., Beth, B., Lin, C., & Russell, G. (2016). Design Principals for Thriving in Our Digital World: A High School Computer Science Course. Journal of Educational Computing Research, 54(4), 443-461. doi:10.1177/0735633115625247

Webb, M., Davis, N., Bell, T., Katz, Y., Reynolds, N., Chambers, D., & Syslo, M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when?. Education & Information Technologies, 22(2), 445-468. doi: 10.1007/s10639-016-9493-x

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), p. 235-254. doi:10/1080/08993408.2016.1257418