Microsoft

Reimagining Computer Science in the Curriculum

White Paper

COMPUTER SCIENCE AS A FUNDAMENTAL SKILL

Computer Science (CS) is the study of principles and practices that underpin an understanding of computation. As the role of computing gradually pervades almost all industries and organizational processes, it is becoming increasingly important for people to possess, at varying levels of proficiency, some understanding of CS. The subject covers a vast area. Its scope embraces the most basic tools through to the most extreme complexity imaginable such as artificial intelligence (AI). There are now few areas of human activity that computing does not touch. It crosses and disrupts traditional boundaries and, regardless of their career paths, it is probable that students will need to draw upon CS skills.

Countries across the globe recognize the benefits of having a strong technology industry and are looking towards their education systems to produce more CS graduates. At the same time, education systems are convinced of the importance of AI in education and are keen to ensure that teachers and students have the skills needed to be both users and developers of AI tools.[i]

Since the COVID-19 pandemic, this has accelerated as society has rapidly digitally transformed to preserve social distancing. All business sectors have shifted to more digital delivery and this has had a huge consequence on required job skills. According to analysis by LinkedIn and Microsoft, 149 million more digital jobs will be created by 2025, in fields such as privacy, data analysis, cyber security and AI. All of these jobs require significant knowledge of computer science.

More broadly, Fluck et al. (2016) see three rationales that countries use to justify an increased focus on CS in schools:

1. **Economic rationale:** Technology is increasingly driving the world's economy, and countries with fewer national resources or exports are seeing the need to create a knowledge economy to grab a larger share of the technology market. With the enormous wealth generated by technology giants such as Microsoft, Apple, Google, and Amazon, countries see an economic opportunity associated with creating stronger local Computer Science-based industries.

2. **Social rationale:** Even though most people use technology every day, they are merely passive consumers of technology, rather than active creators who understand how technology works. Our lives now depend on computers in many ways, and technology is used to control and track sensitive data related to our health, finances, and governance. Societies need more people who have in-depth understandings of how technology works in order to make important decisions about the ethics, privacy, and security of technology, among other issues.

3. **Cultural rationale:** Technology can be considered a driver of cultural change, or a way to preserve cultural heritage. A culture's norms and values are transmitted through the mass production of content, such as movies, games, or social media, which now have an enormous global reach. Countries are seeing the need to for CS education both to create a local content production industry and to assess content coming from other cultures.[ii]

For these reasons, many in the field of Computer Science feel that it should be considered a fundamental skill that all students should learn from a young age,[iii] in particular through developing computational thinking as one of a "repertoire of thinking abilities."[iv] Microsoft Research's Simon Peyton Jones, one of the architects of England's National Computing Curriculum, compares learning CS principles in primary school to learning about science at a young age. Schools do not teach young learners about biology because they are hoping for all students to become biologists later in life. Rather, students learn these topics so that they have a better understanding of the world around them. As children increasingly become consumers of technology at younger ages, they need to understand the principles of how things work so that they can be better informed to make decisions about complex issues surrounding technology.

For decades now, education systems have been talking about 21st century skills that need to be developed in learners in order to solve the problems of tomorrow. These skills include communication, collaboration, problem-solving and critical thinking, among others, all of which are integral to the study of Computer Science.[v] Increasingly, countries are adjusting the 21st century skills to also include computational thinking. Contrary to popular belief, CS is not just about coding. It provides an outlet for creativity and innovation. It is about problem-solving and solution-building. And it is about collaboration and teamwork, as groups of experts in various fields work together to address the complex issues facing our societies.

Education systems are taking various approaches to CS curriculum, either through offering CS as a standalone subject, or integrated across the curriculum. Particularly at primary level, CS is often integrated as computational thinking as a transversal topic, bringing together the arts, STEM, and computational thinking[vi] in new ways to make learning more engaging. In secondary schools, where subjects are typically more siloed, CS is also integrated into other subjects such as mathematics. Only a few countries formally present CS in a cross-curricular way, one example Finland.[vii] However, there is a significant grass roots effort by teachers at both primary and secondary level to bring in CS informally across the curriculum through campaigns like Hour of Code and EU Code Week, as well as via after school programs such as Code Club and CoderDojo.

WHY CREATE A NEW COMPUTER SCIENCE CURRICULUM FRAMEWORK FOR SCHOOLS?

Countries around the world have begun to recognize the benefit of developing a national Computer Science curriculum at a school level. Countries such as Israel, New Zealand, Poland, the United Kingdom (England), Vietnam, the Netherlands, Croatia, and the United States, have developed curricula, frameworks, or standards for computer science.

However, the way in which each country or system has chosen to implement a CS curriculum differs, and much can be learned from their experiences.

- In some countries, even though the CS curriculum is a national, mandatory curriculum, not all students are able to take advantage of it. This is the case in the United Kingdom, for example, where the national computing curriculum was introduced in 2015 in England only. Even though it is mandatory across primary and secondary schools, not all schools are equipped or able to offer all parts of it.
- In other countries, the CS curriculum is not mandatory for all students. In Croatia, the introduction of CS into the curriculum was part of a larger revision of the entire national curriculum. CS is now a mandatory subject for some age groups of students, but not all. In the United States, there is no prescribed national curriculum, so a consortium of organizations led by the Computer Science Teachers Association (CSTA) have developed a framework and set of standards that are available to any state who wishes to adopt them. It is up to the individual state – and then the districts within that state – to determine which students will have access to a Computer Science education and at which point in their schooling.
- Some countries or systems only offer their CS curriculum to secondary students. Computer Science is part of the International Baccalaureate program of study at a diploma level, and in Israel, it is offered at secondary school.
- Still other countries offer CS principles as part of other "fundamental" classes. In Estonia and Norway, CS is integrated into the teaching of many of their core subjects (Mathematics, Science, Music, and Art) throughout primary and secondary school.

Many of these implementations of a "national" CS curriculum still do not provide equitable access to a CS education to all of a nation's students. Even in England, which was one of the earliest countries to adopt a mandatory CS curriculum for all levels of schooling, problems of access and equity still exist. Three years after the new national computing curriculum became compulsory in England, the Royal Society found that only 11% of students had chosen to study CS in secondary school,[1] and only one in five of those students were female. Furthermore, 44% of teachers in England only felt comfortable teaching the lowest levels of the curriculum as they did not believe they had adequate skills in computer science to teach the more advanced topics.[viii]

Moreover, research in New Zealand has indicated that, much like learning a second language, it becomes more challenging for learners when CS is introduced at a later stage of schooling – in secondary school, for example. It is necessary for CS concepts to be introduced at an earlier age so that they can be embedded and built upon progressively.[ix]

The MCSF endeavors to provide a curriculum framework that provides a complete CS education across all levels of schooling. Learnings from examples such as these have informed the development of this curriculum framework, which tries to avoid the pitfalls below.

UPDATING THE CURRICULUM TO REFLECT TECHNOLOGICAL CHANGE

In recent times, computing has evolved at a breathtaking pace, and it will continue to accelerate into the future. Near-constant changes in technology can quickly render a Computer Science curriculum obsolete. England's National Computing curriculum, for example, was published in September 2013 and predates the rapid rise in significance of key areas such as artificial intelligence (AI) and blockchain. It is thus important for curricula to be regularly updated to ensure ongoing relevance.

MAKING CS MORE APPEALING AND RELEVANT TO YOUNG PEOPLE

National level CS curricula can be academic and dry. The curricula seldom link to real-world societal problems, inspiring young people to apply their learnings to develop solutions that might impact the world around them. As a result, too few young people are choosing to study computer science in school, while at the same time the popularity of Computer Science as course choice for university undergraduates grows.

IMPROVING EQUITY IN COMPUTING EDUCATION

Equity in computing education is not just a question of increasing the number of girls who take computing, although that is certainly a global issue. Although many of the great pioneers of computing were women, not enough girls are taking the subject. The US-based National Science Foundation notes that although girls and boys take similar levels of CS in general secondary education, in advanced programs there is significant skew with only 23% of girls and only 18% in bachelor CS degrees. In England, when the subject becomes optional, there is only 20 percent uptake from girls, and this falls to 9 percent for more advanced level studies.

Equity also relates to making sure all students, regardless of socio-economic status, race, ethnicity, or special learning needs, attend schools that offer a computing curriculum and have the opportunity to take computing courses that are appropriate for their needs and abilities. Sadly, this is also an issue even in countries in which a computer science curriculum is compulsory.

---

[1] This is measured by the percentage of student who chose to focus on CS for one of the subjects of their GCSE exams at age 16.

COMPUTER SCIENCE PEDAGOGY CAN BE INVIGORATED WITH NEW APPROACHES

Computer science is often criticized by students as being dry and difficult, with a focus on teacher-centered pedagogies. Teachers also self-report that they need more support in evolving their pedagogy to be more effective in CS education,[x] and to find the right balance between teacher-centric and student-centric methods. New student-centered approaches rooted in constructivism and social constructivism support more hands-on, problem-solving, and inquiry-based approaches,[xi] which in turn are more appealing to diverse groups of students.

A PROGRESSIVE MODEL TO ACHIEVE MASTERY

Currently, there is little collective understanding about what concepts students need to grasp at which age, and how to progressively increase their mastery of CS content, skills, and mindset through their school careers. A new approach, which builds knowledge of fundamental concepts, revisiting and refining them throughout the school career according to student developmental stages, is important.

CONCLUSION

The rapid rise in significance of CS and AI, exacerbated by COIVD-19, adds an urgency to implementing a modern approach to CS curricula. Just as the digital revolution changed the world of information and knowledge, AI is changing the world of intelligence, which is something which students must surely learn to navigate. Students need to become active builders of CS solutions, understand what AI is, learn how to adapt to a constantly changing digital landscape, and build skills that machines cannot emulate. Due to these factors, education decision makers should consider new approaches to CS curriculum for K-12 education.

The Microsoft Computer Science Framework includes curriculum structure and guidance, as well as a program of study objectives and proposed content for Computer Science that spans learners from age 5-18. This curriculum framework is based on our years of expertise as a leader in the computing industry, on academic research around the teaching of Computer Science, and on learnings from countries around the world. Explore the Microsoft Computer Science Framework here, and Structure & Principles here.

MIKE LLOYD Project Leader & Editor
Technology leader Mike Lloyd's passion is to democratise Deep Tech. In September 2013 Mike founded learn-tech.io, a technology start-up, to build learning solutions focussed on AI, Internet of Things (IoT), and climate change mitigation. learn-tech.io has customers for its products and services in Europe and Asia, and Mike works with a wide range of organisations including Pearson, ScienceScope, Microsoft, Queen Mary University of London, Singapore University of Social Sciences, Bankers' Lab and Barclays Bank. Mike is the author of "Schooling at the Speed of Thought", a guide to transforming education, and numerous courses including "AI Demystified", "AI for Leaders" and "Digital Technology Demystified", now published by Pearson. He is also the inventor of "ZEP Island", an educational game that helps people learn how technology, data and IoT can be used to mitigate against climate change.

DR. KRISTEN WEATHERBY is a researcher and education consultant helping start-ups, SMEs and other organisations understand, measure, and communicate their impact in education. Most recently, she helped develop and launch University College London's EDUCATE accelerator supporting edtech start-ups in their development of world class products for education. At the OECD, Dr. Weatherby led the 2013 cycle of the Teaching and Learning International Survey (TALIS), working with governments and partners to survey 104,000 teachers in 34 countries about their working conditions and learning environments. At Microsoft, she led initiatives to support schools and teachers integrating technology into teaching in the United States, the United Kingdom and worldwide. She has a PhD in education technology from the UCL Institute of Education, a Master's in teaching from the University of Michigan and is a governor of her local primary school.

JOHN CURRY is an AI practitioner, solution architect and educator who provides AI based solutions and services to clients across all sectors. He made the transition from solution architect to data scientist in the early 2010's culminating in a Master's in Business Analytics from University College Dublin in 2013. Most recently he has been working with SoptAI of Singapore to develop state-of-the-art demand forecasting solutions for businesses in heavy industry, supply chain and FMCG. He has also partnered with learn-tech-io to develop a framework Computer Science K12 curricula for Microsoft. John has become increasingly involved in the development of AI talent. During his tenure as Product Manager for DataSpark, John founded the internship program in 2014, sourcing talented Computer Science students from 10 countries to work on geospatial analytics. Since 2018, he has been designing and delivering courses on Fintech and Data Science to the Institute of Banking in Ireland. In 2019, John founded Last Mile Artificial Intelligence to help businesses bridge the last mile gap to successful application of AI to practical problems and serves clients in Ireland, the UK, Singapore and the USA.

DONNA BUCKLEY is a mathematics teacher at John Curtin College of the Arts, West Australia's only selective school for gifted and talented students in the Arts. The college is a Teacher Development School in STEM and Donna develops resources to support teachers across the state with their interdisciplinary approaches to their STEM learning programs. Specialising in mathematics, cyber-security, cryptography and the integration of programming across the curriculum she has presented on these topics at Mathematics, STEM and Cyber Conferences across Australia. She is a member of the Maths Association of Western Australia Board, coordinating Maths Talent Quest, a student activity that encourages students to consider real world problems and apply mathematical and statistical thinking processes to solve them. Volunteering as a tutor at the Perth Girls Programming Network, she supports girls in secondary schools to develop their Python programming skills. A member of the Day of STEM Education Advisory Board, she provides feedback on their programs and actively promotes their STEM careers resources in Cyber-security and Data Science. Donna is an author of the MAWA Mathematics Essential Textbooks Units 1 – 4.

---

[i] UNESCO (2019) First ever consensus on Artificial Intelligence and Education published by UNESCO. *UNESCO website*. https://en.unesco.org/news/first-ever-consensus-artificial-intelligence-and-education-published-unesco

[ii] 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.

[iii] Margolis, J., & Goode, J. (2016). Ten lessons for computer science for all. *ACM Inroads*, *7*(4), 52-56.

[iv] Voogt, J., Fisser, P., Good, J., Mishra, P. & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20, 715-728.

[v] Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies*, *22*(2), 421-443.

[vi] Henrique de Paula, B., Burn, A., Noss, R., Valente, J.A. (2018) Playing Beowulf: Bridging computational thinking, arts and literature through game-making. *International Journal of Child-Computer Interaction*, 16 (June), 39-46.

[vii] Balanskat, A. & Engelhardt, K. (2015) *Computing our future. Computer programming and coding priorities, school curricula and initiatives across Europe*. Brussels: European Schoolnet. Accessed from https://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0

[viii] The Royal Society. (2017). *After the reboot: Computing education in UK schools.* London: The Royal Society. Accessed from https://royalsociety.org/~/media/policy/projects/computing-education/computing-education-report.pdf

[ix] 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.

[x] Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. Australian Journal of Teacher Education, 42(3).

[xi] Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I.K., Somanath, S., Weber, J. & Yiu, C. (2017) A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education* vol. 3, pages 154–171(2017