THE ROYAL SOCIETY

Making computer science education relevant, engaging and accessible

by Joysy John

Joysy John is an advisor, speaker, and consultant for Edtech, and a council member of the Foundation for Education Development (FED). She was previously the CEO of 01 Founders and is a former Director of Education at Nesta. She holds an Honours degree in Computer Engineering from Nanyang Technological University and attained an MBA with Distinction from London Business School, during which she was awarded a Forte Foundation Fellowship. She helped to launch three non-profits focused on education, including being the Founder of the London chapter of the Startup Leadership Program, of which she was also the Director until 2020. In 2016 she was a Finalist for Woman of the Year at the 16th Asian Achievers Award, and she was named in the *Financial Times* list of the Top 100 Most Influential BAME Leaders in UK Tech Sector in 2019.

There is an unprecedented digital transformation that has happened across different industries due to the coronavirus pandemic. The way we live, work, learn and play has changed dramatically.

The demand for digital skills – especially specialist skills in cybersecurity, blockchain, artificial intelligence, cloud computing and data science – is high and it is growing at an extraordinary rate.

The need for digitally skilled talent is being experienced by every business, every organisation and every charity and not just companies that might be labelled as 'tech' or 'digital'. And so, correspondingly, there is an equally urgent need to figure out how the education system should respond. How can we reimagine learning such that it empowers young people to study computing, build these high demand skills and develop real-world problem-solving capabilities?

Current initiatives haven't worked

Much has been written about the need to reform computing education in the UK, and there have even been fairly dramatic efforts at a ministerial level to do something about it. Secretary of State for Education Michael Gove even made computer science compulsory in 2012. The computer science curriculum was revised and from 2017, students no longer had the option to study GCSE and A-levels in Information and Communication Technology (ICT).

> "What went wrong? The answer lies in a fundamental mistake: attempting to create a new school subject – 'computer science' – and shoe-horning it into an out-of-date curriculum that is too dependent on subject silos and an old-fashioned idea of academic disciplines."

But so far it hasn't worked; in fact, quite the opposite. There has been no dramatic increase in the take up of the subject. In 2020, according to BCS (the Chartered Institute for IT) Computer Science was ranked as:

- The 16th most popular GCSE with 13.4% of the cohort taking the examination;
- The 18th most popular A level with 6% of the cohort taking the examination;
- At both A level and GCSE, it is the least popular of the four main sciences, with there being twice as many Physics GCSE entrants and three times as many A level Physics entrants.

"By the time students graduate, the digital skills needed for future jobs will have been transformed out of all recognition. Even people working in the sector know that tech skills have a short shelf-life and they need to invest in upskilling."

It risks leaving girls and pupils from poorer backgrounds and ethnic minorities behind, as evidenced by the number and diversity of students studying the subject. Whilst previously the male:female student ratio in ICT participation at GCSE and A level, was at about 2 – 3:1, for Computer Science this has remained at 4:1 for GCSE while at A level it has been as high as 10:1. Education experts have reiterated their concerns about the unrepresentative nature of entries for the Computer Science GCSE. Teacher supply, lack of confidence and the gender gap are issues that have hindered progress.

What went wrong? The answer lies in a fundamental mistake: attempting to create a new school subject – 'computer science' – and shoe-horning it into an out-of-date curriculum that is too dependent on subject silos and an old-fashioned idea of academic disciplines.

Computer science is constantly evolving. It is evolving at a much faster pace than a rigid school curriculum can keep up with. By the time students graduate, the digital skills needed for future jobs will have been transformed out of all recognition. Even people working in the sector know that tech skills have a short shelf-life and they need to invest in upskilling.

As it stands the curriculum is a set body of work and it takes a long time for it to be changed. The skills that the digital economy needs cannot be reduced to a core set of unchangeable facts or disciplines. We need people who can solve problems, communicate ideas and analyse challenges.

Some unconventional solutions

More generally, we need to broaden out the curriculum and to stop thinking about narrow subject areas each with a closed body of knowledge. We need to evolve what is taught, how it is taught and to think about how we make it more engaging, more accessible and more relevant.

Specifically, we need to stop thinking about computer science as a standalone subject. It is, of course a discipline that needs specialism but it also needs to be embedded across the curriculum. We need to think about how we can make it more practical than theoretical, and how we can integrate it into other subjects. At a younger age, we need to be exposing more kids to computing, to make sure that they see the relevance and they see how this will help in any area that they want to pursue.

Take a subject like geography. Where you might ask students to try to think about the weather patterns, or different levels of population density, or climate change, there are algorithms that could be developed to understand data and predict changes. We need to think about how we get kids to think about algorithms at a much younger age, and to think about how this is applicable in their real life. For example, in maths they might think about comparing data from different mortgage or insurance providers and deciding which option would be most suitable. There is so much that can be done to embed these skills within existing areas of study.

Practical not theoretical

But this is also about making the subject come alive and making it more project-based. It's about linking it back to the interests that people have such as music, or art, or fashion, or sustainability. This is particularly relevant when it comes to girls or people from disadvantaged backgrounds studying computing. They might not identify themselves as being nerdy computer programmers. We must link it back to passions that people already have and to real-world applications.

This is also an opportunity to bring employers into the schools and colleges to offer real-world projects that students can work on. Ada, the National College for Digital Skills has engaged a range of employers to ensure that their students are learning from working on industry briefs and challenges.

"We need to think about how we build agency in both teachers and learners, so that they are curious, and they are looking for ways to build their capabilities and skills in new areas."

Technology is pervasive – it is in every single walk of life – and coding and computing should be seen in the same way in schools. We can build that awareness into young people that it's not just for the smart kids or the kids who are great at maths. Everyone can develop these skills. But you can only do that by getting people excited about it.

Instead of forcing all children to study computer science we need to help them develop the ability to use computational thinking in terms of how they approach problems, abstract them, break them down into subcomponents, and then build out potential solutions. This certainly does not need to happen within a narrow subject area. For example, Computer Based Maths (established by Conrad Wolfram) is helping students solve real world maths problems using computational thinking process rather than historical hand calculating techniques. Estonia was the first country to adopt this novel approach.

The way forward

This sounds great rhetorically – but is the education system in the UK as it is currently configured remotely ready to make this leap? Is the teaching workforce ready?

Just now, the answer would seem to be no. We must start to reimagine the education system if we want a world where all learners are able to make the most of the opportunities presented by our fast-changing world. We need a radical transformation in the way that schools operate and how teaching is organised if we are to guarantee a flow of talent into the digital economy. There is no silver bullet for this, and it will need collaboration across government, industry, education and the voluntary sector.

We need to move away from this notion that school is just what is confined within a particular subject and instead look at it in an interdisciplinary way. We need to think about how we build agency in both teachers and learners, so that they are curious, and they are looking for ways to build their capabilities and skills in new areas. We can take inspiration from new universities like New Model Institute for Technology and Engineering (NMITE), The Engineering and Design Institute (TEDI) and the London Interdisciplinary School (LIS) as we reimagine interdisciplinary teaching and learning in schools.

Computer science is essential to the future of this country – but it touches all aspects of our lives in the here and now too. We must embrace this. We must build out the skills associated in a much more inclusive and open way, and bring computational thinking into everything that we do in education – and beyond.

This document is part of the Royal Society's Envision project. Envision brings together thought leaders to discuss what the UK education system should look like in order to prepare students to flourish in a changing world of work in the 21st Century.

The views in this piece are those of the author and do not necessarily represent those of the Royal Society.