



CODING WITHOUT A **COMPUTER**

Coding and Computing are becoming more and more important in the working world, but many teachers are unequipped to teach the subject in the Primary sector. **Daniel Sydes**, Computer Science expert at UTC Reading, proposes an innovative new Computing curriculum to tackle the problem.

There is a nationwide push for switching out IT for Computer Science, and no one is as glad as me. After eight years teaching spreadsheets, Desktop Publishing and (*shudder*) slideshows, I was just about ready to start looking for another career. My three years of Computer Science at University was not being particularly well used.

No matter your thoughts on Michael Gove, I am thankful for his support of the Computer Science movement. Looking at some of the leading schools from the 2015 league tables, Pate's Grammar School

Gloucestershire ([#1 GCSE results 2015](#)) offers Computing GCSE on top of core non-examined IT. Computing is also an option at The Tiffin's Girls School and The Reading School (#2 and #3 respectively). I would not be surprised if they move out of the options pool and into the core curriculum in the next few years.

Through my current role at the UTC I have had a lot of contact with the technology industry. I have had numerous meetings with companies like Microsoft, Cisco, Oracle and Fujitsu. I would heartily recommend members of SLT in any School or College to seek out working partnerships with local industries. I have been amazed at how willing even large multinational companies with billions of dollars annual turnover are



to get directly involved with and contribute towards education.

There are many invaluable things I learn by working with industry on a regular basis. Some are little titbits – one of the world's leading consultancy firms prefer Java and C# knowledge to any other programming language when recruiting IT graduates, for example. But from all the contact I have had with companies large and small, IT specialised and not, one consistent message has come through to me; they struggle to find young people with adequate coding ability from the UK talent pool.

How long would a big multinational technology firm keep its international HQ in the UK if they have to recruit all their coding talent from elsewhere?

And you can think again if you believe this problem only affects the technology industry. I have also worked closely with employees at director level with a rail company and a supermarket. Coding skills are needed across the whole of UK industry. You want to go into graphics? Vector graphics can be coded, Photoshop runs complex algorithms that the best artists need to understand. You want to be a sporting hero? You need a level of technical performance analysis.

Even if a job does not require any coding directly, young people who can code can break problems down into the smallest elements, and they can solve the problem using logic in a creative way. Employers are demanding this skill of problem solving from their new recruits.

Coding is applied mathematics with real-life function. It also enables creativity and expressiveness like a canvas and palette. I think it is a wrong that young people with a flair for the creative and expressive arts should look away from STEM and coding as a potential career avenue. The tools may be different, but in areas such as digital arts, digital content creation and games development you may find your creativity buttons being pushed.



TEACHING CODING

How do we start to teach coding? Coding is in the national curriculum at Key Stage 1. Students are expected to “create and debug simple programs”. At Key Stage 2 students are expected to:

- “design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts”
- “use sequence, selection, and repetition in programs; work with variables and various forms of input and output”
- “use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs”.

Remember these are 8-11 year olds we are talking about here, often taught by individuals who did their Primary teaching certification in the 1980s. What a terrifying proposition for a teacher who has never had to teach more than typing and slides. Some Primary Schools have access to an IT specialist – but I wonder how many have access to a Computing specialist.

I wholly approve the intentions of the Computing curriculum at Key Stage 1 and 2, but we need to put some thought into how we are going to put these things into practice. I believe a great step in the right direction is secondary education establishment support. Ever since getting my QTS in 2005 all my job roles have involved some work with local primary schools. It is a wise investment of a small proportion of teachers' time to become involved with the feeder Primaries. This enables positive relationships to be built, fulfilling of school's specialist ethos and aims, good marketing and recruitment exercise and generally positive support for the community.



I have been involved with some exciting projects at UTC Reading and some of our partnerships with local Primary Schools (we are next door to Alfred Sutton Primary - think Suttons Seeds). We ran a programming after school club for a term with some of our Key Stage 4 and 5 students supporting Year 4-6 students building simple apps, games and websites. We hosted a female only robotics competition where girls built Lego robots and programmed them to be the best robot golfers with prizes for the furthest hit of a golf ball. This was part of our commitment to bringing in more females into the technology industry and aim for a more equal ratio of boys to girls at our school in the long term.

The exciting and interesting thing about running these projects is to witness first-hand the capability of students in Key Stage 2. Coming from a secondary school background, I am amazed by how quickly concepts are picked up and put into practice by a ten year old. The thirst for learning, willingness to have a go, and fearlessness of Primary aged students makes it clear to me that it is right that we strive to find a way to bring effective Computing teaching and learning to this age group. I wonder what else we can get Key Stage 1 students to do using the core concepts of coding?

Of course, to be able to provide support for the local primary schools, the secondary school must have an adequate Computing knowledge base in the first place. In my experience in Senior Secondary School Leadership, recruitment of Computer Science specialist teachers is not the world's easiest feat. Realistically we may not be able to rely entirely on the secondary sector.

My proposed solution is to provide a "Computing without Code" curriculum with easy to use resources for all teachers. We can use a range of simple props and role playing activities that will help students.

After experimenting with different teaching ideas for coding I have found that even for a high-end A Level student some new basic concepts need that visual stimulus and real world metaphor to provide the "light bulb" moment. It is entirely possible while we wait for the core computing curriculum to filter through to all young people in the UK that we may also get Higher Education students benefitting from Coding Without Computers.

PROVIDING A WORKING CONTEXT

The first thing we have to get our students comfortable with is the idea of a variable. I use an empty lunchbox as a visual prop. The lunchbox on its own is no use; we have to put something into it to get some use out of it. We can put a sandwich in it on Monday, a pasty on Tuesday and a salad on Wednesday. Every day the contents of the lunchbox vary. The lunchbox on its own is the variable and its contents are the data. We set the variable to the value of the data.

“ I REITERATE THAT I USE THIS TEACHING METHOD WITH STUDENTS RIGHT UP TO A LEVEL, ESPECIALLY AS MANY STILL HAVE NOT TAKEN ANY COMPULSORY COMPUTING EDUCATION UP TO THAT POINT.

To bring aspects of literacy into your Computing teaching you can encourage students to start to use terms like ‘vary’, ‘variable’, ‘integer’ and ‘string’.

In my experience, the most common challenge that students have when understanding the concept of a variable is that it is a vessel rather than data itself and it can change as many times as you like. By using the visual aid of the lunchbox the student can easily relate the concept to something they are very familiar with. I reiterate that I use this teaching method with students right up to A Level, especially as many still have not taken any compulsory Computing education up to that point.

I am sure that teachers from the Primary sector who have little experience with Computing can also relate to the concept using props, visual aids and easily relatable metaphors. Over time teachers can develop new and exciting ways of teaching Coding Without Computers to deliver across the early Key Stages, then students coming up through the system will already be familiar and confident with core concepts can then focus on building high level in-demand skills.

I have used Coding without Computers on other core concepts as well including IF THEN ELSE, For and Do loops, arrays, validation checking, sorting algorithms, searching algorithms, nested loops and so on and so on.

Spending time away from computers using visual aids, discussion and role play helps students of all ages to grasp coding and I find a powerful starting point leading onto actual coding challenges.

I am sure there are plenty of examples of good practice in classrooms across the UK. I would love to see all these ideas compiled into a teacher’s guide in the near future.

In the meantime I am looking forward to the challenge of helping bring effective Computer Science and coding education into the Primary sector and seeing how far we can go with code at Secondary. I hope the drive for improved Computer Science is more than a flash in the pan and the UK is able to produce more graduates (and Apprenticeship-ready young people) in the future to keep the UK at the cutting edge of technology.

Daniel Sydes is Assistant Principal for Computer Science and Mathematics at UTC Reading.
daniel.sydes@utcreading.org.uk