The Observer

THE NEW REVIEW

SUNDAY 1 APRIL 2012

WE NEED TO TEACH OUR KIDS TO CODE

A MANIFESTO

John Naughton on why it's time to rethink how we teach computing

For two decades our computer education has been hopeless. It's time to rethink, and let our kids lay claim to the future

Here's the problem: we are producing a generation of technological suckers – people who can play in the digital world but have no understanding of how it works, or how to make it work. We need to overhaul what we teach our children so they are in control, says John Naughton, who offers a manifesto (right). We also talk to young coders and teachers about their ideas for change

A vigorous debate has begun – within government and elsewhere – about what should be done about information and communication technology (ICT) in the school curriculum. Various bodies – the Royal Society, the Association for Learning Technology, Computing at School (a grassroots organisation of concerned teachers) and the British Computer Society, to name just four – have published reports and discussion documents aimed at ministers and the Department for Education. Michael Gove, the education secretary, made an enigmatic speech at the recent BETT technology conference indicating that a rethink is under way in the bowels of Whitehall. Meanwhile, in another part of the forest, there are some astonishing developments happening – such as the fact that more than a million people have already placed orders for Raspberry Pi, the cheap, credit-card-sized computer developed by Cambridge geeks, which began shipping last week.

So something's happening: there's a sense of tectonic plates shifting. But as with most big policy debates, there's a lot of axe-grinding, lobbying and special pleading going on. Universities want to reverse the decline in applicants for computer science courses. Gaming companies want more programmers. The government wants more high-tech start-ups. Manufacturers want trainees who can design embedded systems. And head teachers want bigger budgets for even more computer labs. And so on.

What's missing from all this is a big vision. So here's my shot at one:

Starting in primary school, children from all backgrounds and every part of the UK should have the opportunity to: learn some of the key ideas of computer science; understand computational thinking; learn to program; and have the opportunity to progress to the next level of excellence in these activities.

We'll get to why this is important and necessary in a moment, but first we need to face up to a painful fact. It is that almost everything we have done over the last two decades in the area of ICT education in British schools has been misguided and largely futile. Instead of *educating* children about the most revolutionary technology of their young lifetimes, we have focused on *training* them to use obsolescent software products. And we did this because we fell into what the philosopher Gilbert Ryle would have called a "category mistake" – an error in which things of one kind are presented as if they belonged to another. We made the mistake of thinking that learning about computing is like learning to drive a car, and since a knowledge of internal combustion technology is not essential for becoming a proficient driver, it followed that an understanding of how computers work was not important for our children. The crowning apotheosis of this category mistake is a much-vaunted "qualification" called the European Computer Driving Licence.

What we forgot was that cars don't run the world, monitor our communications, manage our bank accounts, keep our diaries, mediate our social relationships, snoop on our social activities and even – in some countries – count our votes. But

Continued on page 10

'Everyone needs to know coding'

Priscilla Ossai, 18

Where are you studying? I'm at school in Chingford, London.

How did you get into coding? My uncle is a systems analyst, and he works all around the world. I was always interested in what he was doing. Slowly he started teaching me how to do it. It was amazing.

What drew you in? Technology is changing so quickly, you really have to be up to date and you have to learn more and more things. I think that's what really caught my attention.

What's the attitude to coding at school? As soon as I got into sixth form, my ICT lessons helped me grasp it. In the first term I learnt CSS, HTML codes, how to make websites, stuff like that. Before that I was learning, but I wasn't really understanding what I was doing.

Do you think it should be taught more in schools? Yes, because coding makes everything so much quicker – you can just go straight to what you need to do and it saves you so much time. I think everyone needs to know coding. At least a little bit. If they did push it more in school, I think a lot of people would go for computer science instead.

Where do you see yourself in the future? I'd like to be a systems analyst like my uncle, but I think that's all I know – maybe if I saw other avenues in computing I'd change my mind. **Gemma Kappala-Ramsamy**

PORTRAITS BY KATHERINE ROSE

MANIFESTO

To: The Rt Hon Michael Gove MP, Secretary of State for Education
Subject: Proposals for rebooting the computing curriculum

1 We welcome the clear signs that the government is alert to the deficiencies in the teaching of information and communications technology (ICT) in the national curriculum, and the indications you and your ministerial colleagues have made that it will be withdrawn and reviewed. We welcome your willingness to institute a public consultation on this matter and the various responses you have already made to submissions from a wide spectrum of interested parties.

2 However, we are concerned that the various rationales currently being offered for radical overhaul of the ICT curriculum are short-sighted and limited. They give too much emphasis to the special pleading of particular institutions and industries (universities and software companies, for example), or frame the need for better teaching in purely economic terms as being good for "UK plc". These are significant reasons, but they are not the most important justification, which is that in a world shaped and dependent on networking technology, an understanding of computing is essential for informed *citizenship*.

3 We believe every child should have the opportunity to learn computer science, from primary school up to and including further education. We teach elementary physics to every child, not primarily to train physicists but because each of them lives in a world governed by physical systems. In the same way, every child should learn some computer science from an early age because they live in a world in which computation is ubiquitous. A crucial minority will go on to become the engineers and entrepreneurs who drive the digital economy, so there is a complementary economic motivation for transforming the curriculum.

4 Our emphasis on computer science implies a recognition that this is a serious academic discipline in its own right and not (as many people mistakenly believe) merely acquiring skills in the use of constantly outdated information appliances and shrink-wrapped software. Your BETT speech makes this point clearly, but the message has not yet been received by many headteachers.

5 We welcome your declaration that the Department for Education will henceforth not attempt to "micro-manage" curricula from Whitehall but instead will encourage universities and other institutions to develop high-quality qualifications and curricula in this area.

6 We believe the proper role of government in this context is to frame high-level policy goals in such a way that a wide variety of providers and concerned institutions are incentivised to do what is in the long-term interests of our children and the society they will inherit. An excellent precedent for this has in fact been set by your department in the preface to the *National Plan for Music Education*, which states: "High-quality music education enables lifelong participation in, and enjoyment of, music, as well as underpinning excellence and professionalism for those who choose not to pursue a career in music. Children from all backgrounds and every part of the UK should have the opportunity to learn a musical instrument; to make music with others; to learn to sing; and to have the opportunity to progress to the next level of excellence." Substituting "computing" for "music" in this declaration would provide a good illustration of what we have in mind as a goal for transforming the teaching of computing in schools. Without clear leadership of this sort, there is a danger schools will see the withdrawal of the programme of study for ICT in England as a reason for their school to withdraw from the subject in favour of English baccalaureate subjects.

7 Like you, we are encouraged by the astonishing level of public interest in the Raspberry Pi project, which can bring affordable, programmable computers within the reach of every child. But understanding how an individual machine works is only part of the story. We are rapidly moving from a world where the PC was the computer to one where "the network is the computer". The evolution of "cloud computing" means that the world wide web is morphing into the "world wide computer" and the teaching of computer science needs to take that on board.

8 In considering how the transformation of the curriculum can be achieved, we urge you to harness a resource that has hitherto been relatively under-utilised – school governors. It would be very helpful if you could put the government's weight behind the strategic information pack on *Teaching Computer Science in Schools* prepared by the Computing at School group, which has been sent to every head teacher of a state-maintained secondary school in England to ensure that this document is shared with the governors of these schools.

9 We recognise that a key obstacle to achieving the necessary transformation of the computing curriculum is the shortage of skilled and enthusiastic teachers. The government has already recognised an analogous problem with regard to mathematics teachers and we recommend similar initiatives be undertaken with respect to computer science. We need to a) encourage more qualified professionals to become ICT teachers and b) offer a national programme of continuing professional development (CPD) to enhance the teachers' skills. It is unreasonable to expect a national CPD programme to appear out of thin air from "the community": your department must have a role in resourcing it.

10 We recognise that teaching of computer science will inevitably start from a very low base in most UK schools. To incentivise them to adopt a rigorous discipline, computer science GCSEs must be added to the English baccalaureate. Without such incentives, take-up of a new subject whose GCSE grades will be more maths-like than ICT-like will be low. Like it or not, headteachers are driven by the measures that you create.

11 In summary, we have a once-in-a-lifetime opportunity to prepare our children to play a full part in the world they will inherit. Doing so will yield economic and social benefits – and ensure they will be on the right side of the "program or be programmed" choice that faces every citizen in a networked world.

It's time to change the way computing is taught

Continued from page 8

networked computers do all of these things, and a lot more besides.

So we need to admit that "ICT in schools" has become a toxic brand. We have to replace it with a subject that is relevant, intellectually sustaining and life-enhancing for students. For want of a better name, let us call it computer science. This is an umbrella term that covers two distinct areas. First a set of key concepts that are essential if schoolchildren are to understand the networked world in which they are growing up. And second, computer science involves a new way of thinking about problem-solving: it's called computational thinking, and it's about understanding the difference between human and artificial intelligence, as well as about thinking recursively, being alert to the need for prevention, detection and protection against risks, using abstraction and decomposition when tackling large tasks, and deploying heuristic reasoning, iteration and search to discover solutions to complex problems.

There will be lots of interesting discussions about the key concepts that students will need to understand,

If we don't act now, our children will grow up as passive consumers of closed devices and services

but here's one possible list for starters. Kids need to know about: algorithms (the mathematical recipes that make up programs); cryptography (how confidential information is protected on the net); machine intelligence (how services such as YouTube, NetFlix, Google and Amazon predict your preferences); computational biology (how the genetic code works); search (how we find needles in a billion haystacks); recursion (a method where the solution to a problem depends on solutions to smaller instances of the same problem); and heuristics (experience-based techniques for problem-solving, learning, and discovery).

If these concepts seem arcane to most readers, it's because we live in a culture that has systematically blindsided them to such ideas for generations. In that sense, CP Snow's "Two Cultures" are alive and well and living in the UK. And if you think they are too sophisticated to be taught to small children, then that's because you've never seen gifted and imaginative teachers go to work on them. In fact many UK readers in their 30s will have been exposed to recursion, for example, because once upon a time many UK schools taught Logo programming, enabling children to learn how a mechanised turtle could be instructed to carry out complex manoeuvres. But in the end most of those schools gave up teaching Logo and moved backwards to training kids to use Microsoft Word.

Incidentally, the Logo story provides a good illustration of why teaching kids to write computer programs has to be an integral part of any new computer science curriculum. The reason is that there's no better way of helping someone to understand ideas such as recursion or algorithms than by getting them to write the code that will implement those concepts. That's why the fashionable mantra that emerged recently – that "code is the new Latin" – is so perniciously clueless. It implies that programming is an engaging but fundamentally useless and optional skill. Latin is an intriguing, but dead, language; computer code is the lingo of networked life – and also, it turns out, of genetic replication.

Another misconception that is currently rife in the debate about a new curriculum is that the primary rationale for it is economic: we need more kids to understand this stuff because our "creative" industries need an inflow of recruits who can write code, which in turn implies our universities need a constant inflow of kids who are turned on by computers. That's true, of course, but it's not the main reason why we need to make radical changes in our educational system.

The biggest justification for change is not economic but moral. It is that if we don't act now we will be short-changing our children. They live in a world that is shaped by physics, chemistry, biology and history, and so we – rightly – want them to understand these things. But their world will be also shaped and configured by networked computing and if they don't have a deeper understanding of this stuff then they will effectively be intellectually crippled. They will grow up as passive consumers of closed devices and services, leading lives that are increasingly circumscribed by technologies created by elites working for huge corporations such as Google, Facebook and the like. We will, in effect, be breeding generations of hamsters for the glittering wheels of cages built by Mark Zuckerberg and his kind.

Is that what we want? Of course not. So let's get on with it.

'I have already got my dream job'

Jordan Hatch, 17

What's your background? I was born in Preston and went to Kingsfold school. I still go home at weekends.

How did you get into coding? When I was six, we got our first computer and I absolutely loved it. My parents bought me my own a few years later. I started reading a lot online about programming and I loved learning to build things I find useful myself.

Should coding be part of the curriculum? Definitely. I went to quite a small secondary school, and our IT teacher really encouraged me and a few others. She let us build things for the school, for example the intranet, which was really useful. Kids should learn to code because you learn problem-solving skills, and that's really valuable.

What are you doing now? I work for the Cabinet Office, for the Government Digital Service (GDS). We're the team building the small-domain new government platform, and we intend to replace online services over the next few years. I've been living in London, staying in hotels during the week, for five months. It's good fun at first but the novelty wears off.

Have you ever hacked? I have, in a good way. At hack days people come together and are set challenges in the morning using government data or whatever. We all brainstorm some projects, build a prototype and show it to the press at the end of the weekend. It's really good fun.

What music/blogs/magazines do you like? I like indie rock. I like John Gruber's blog Daring Fireball and a technology news site called the Verge.

How do you see your coding future? This is already very much my dream job. I'm not sure where it'll take me but there is a lot of potential at GDS. **Kathy Sweeney**

'Suddenly everyone's saying, let's program'

Kush Depala, 17

Background I grew up in London. I'm still at school in London.

How did you get into coding? It started last August, at a Rewired State event. It was a bit daunting but I teamed up with someone called Josh and we put together our first project, a little thing called Snackonomics, which translates economic concepts like debt into things you'd understand, like Jammie Dodgers and coffees from Starbucks.

Since then, I've been learning as much as I can. I got on to the Microsoft student partnership, which is a cool little thing. They give us support and in return we write really basic stuff for them. And in January I started my lunchtime club at school, teaching younger students how to use code.

Do you want to make a career out of it? I've always liked technology and it's been interesting to go from being a consumer to actually making stuff. But I'll probably end up doing a degree in classics.

What is the attitude towards coding at school? It was awkward up to the point when Michael Gove decided to scrap ICT education and then all of a sudden everyone started saying, "Yeah, let's start programming." It used to be very outdated. My coursework was a 60-page Word document explaining how I made an Excel document and how it worked and it was painful and tedious. Now the school is embracing new kinds of things, including programming, to give a sense of what's going on behind the scenes.

Do you think it is important to learn the discipline of coding? I don't know much about economics but to me the only sector that will last for any length of time is the technology sector. Everything else is stagnant and oil will eventually run out, so I guess the only place we can really go is technology.

How many languages have you mastered? I've mastered none, but I have exposure to HTML, CSS, a little bit of JavaScript, a tiny bit of Ruby and I've started looking at C#, a slightly bigger one, which means I can start programming things properly. **Killian Fox**

THE EXPERT'S VIEW
DALLAS CAMPBELL

Dallas Campbell is a presenter on the BBC1 show *Bang Goes the Theory*

I remember the five dusty BBC Micros in our school – I'd goof off games, copying lines of Basic from magazines. It was slavish but there was a sense of responsibility, creation – even if all you're doing is getting "Dallas is great" to scroll up and down the screen. Understanding how things work has obvious economic incentives, but beyond that it actually adds value and wonder to our engagement with the world. Arthur C Clarke said any sufficiently advanced technology is indistinguishable from magic. We're fluent in technology's uses but not its creation, how stuff actually gets on the screen. It's important to get under the bonnet of the modern world, or we disconnect. It's a question of visibility – say coding and most people don't know what you're talking about; but what you're talking about is the language of the future.

THE EXPERT'S VIEW
IAIN DODSWORTH

Iain Dodsworth is the founder of tweetdeck

My kids – five and 9 – are already coding but don't know it. Programs like Kodu (Xbox) are great. They guide a fish round a world they create – but they're also building variables, objects, methods, events. Older kids' identity is fundamentally constructed online – it's a small step to take control of construction, build your own website. A natural step from there is hacking together solutions to work problems. Hacking has negative, Hollywood-influenced connotations, but is just makeshift problem-solving. If your job needs you to filter lots of data it can take days to go through IT departments. Put together your own scrappy tool, get there quicker. It's not about high-end software engineering; I'm not a great technical coder. We don't learn English and maths for specific jobs: they're life skills. Coding is too.

'You can learn quicker away from school'

Lawrence Job, 17

Background I'm from Worthing and in my first year studying computer engineering at Sussex University.

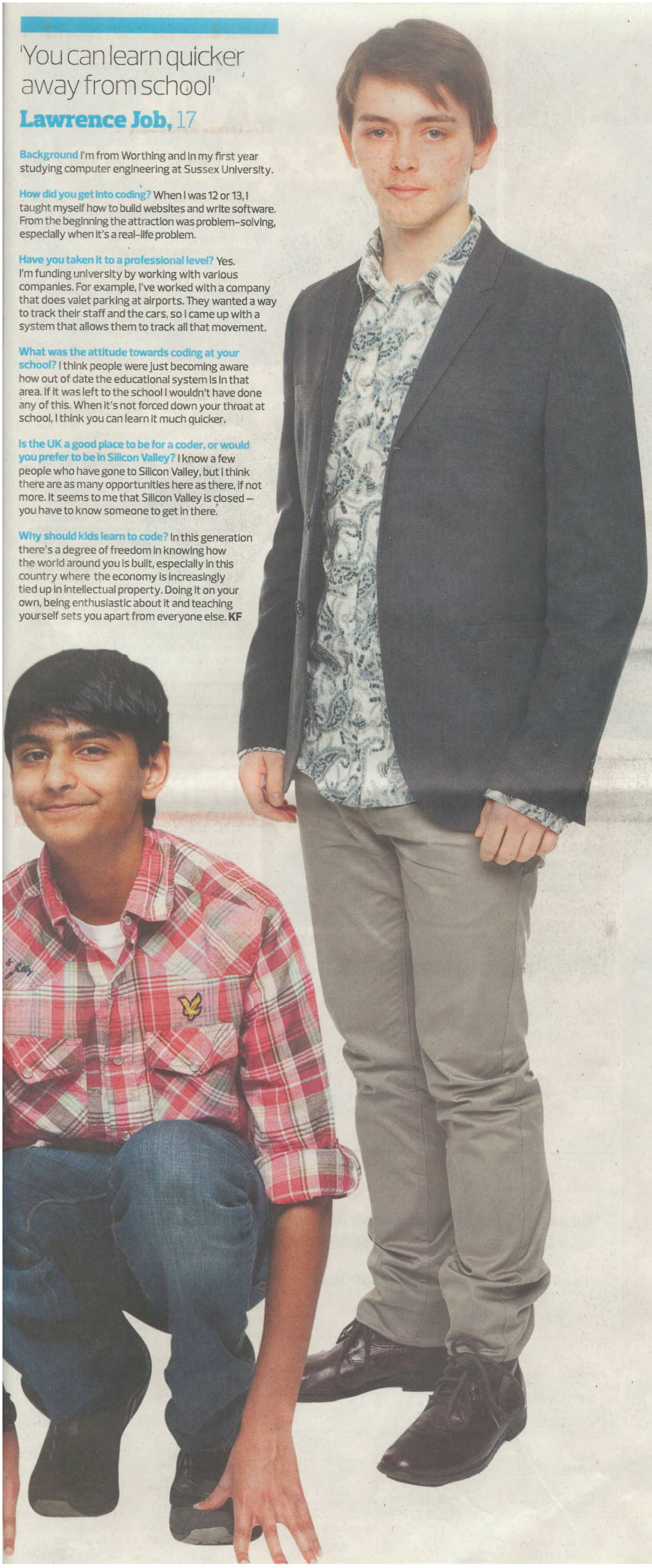
How did you get into coding? When I was 12 or 13, I taught myself how to build websites and write software. From the beginning the attraction was problem-solving, especially when it's a real-life problem.

Have you taken it to a professional level? Yes. I'm funding university by working with various companies. For example, I've worked with a company that does valet parking at airports. They wanted a way to track their staff and the cars, so I came up with a system that allows them to track all that movement.

What was the attitude towards coding at your school? I think people were just becoming aware how out of date the educational system is in that area. If it was left to the school I wouldn't have done any of this. When it's not forced down your throat at school, I think you can learn it much quicker.

Is the UK a good place to be for a coder, or would you prefer to be in Silicon Valley? I know a few people who have gone to Silicon Valley, but I think there are as many opportunities here as there, if not more. It seems to me that Silicon Valley is closed – you have to know someone to get in there.

Why should kids learn to code? In this generation there's a degree of freedom in knowing how the world around you is built, especially in this country where the economy is increasingly tied up in intellectual property. Doing it on your own, being enthusiastic about it and teaching yourself sets you apart from everyone else. **KF**



A computer studies class at school in Hazira, Gujarat, India. Alamy

What's happening in the rest of the world?

The IT teaching debate is taking place around the world. Here are some of the latest developments

How is computing (ie computer science) taught in other countries? A look around the world shows that most other countries are experiencing the same epiphany as the UK. The current ferment of excitement around computer science as a school subject here (Google chairman Eric Schmidt's speech criticising British educatioin, Gove's announcements at BETT, the Royal Society report, new GCSEs announced, Raspberry Pi and .NET Gadgeteer, the Turing centenary, etc) makes the UK unique. Other countries are now looking to Britain for lessons about how to do it right. Comparisons are not straightforward, because of the diversity of educational systems. Nevertheless, there are some lessons we can learn from the international experience:

1) It is vital to make a clear distinction between computer science as a rigorous subject discipline on the one hand, and IT applications and/or digital literacy on the other.

2) Every other country has recognised, or is recognising, that computer science can and should be learned by every child, in the same way they learn science or maths. Many countries, such as the US, already teach computer science in some form to schoolchildren from an early age.

3) The continuing strong employer demand for IT professionals reduces the supply of well-qualified potential teachers. Moreover, the very ubiquity of information technology means that schools often use non-specialists to teach ICT. These factors conspire to mean that ICT/computing teachers are undervalued and underqualified.

Scotland

Scotland is in the midst of a major revision of the school curriculum, called Curriculum for Excellence. Computing science is firmly in the new curriculum, and the Royal Society of Edinburgh is now running an "exemplification" programme to develop teaching materials and training to support this strand, particularly at the pre-14 age level, at which all students will encounter computing.

Israel

Israel undertook a major review of computing at school around the turn of the century, and now has the most rigorous computer science high school programme in the world.

New Zealand

New Zealand has revamped its school curriculum in digital technologies, and from 2011 has had an explicit strand entitled "programming and computer science". The famous *Computer Science Unplugged* book, which describes dozens of activities that teach computer science without going near a computer, was written and trialled in New Zealand.

United States of America

There is a great deal of activity in the US aimed at improving the state of high-school education in computer science, albeit vastly complicated by the need to deal with 50 states and thousands of autonomous school districts. For example a new "computer science principles" course aims to focus on the foundational principles of computer science. At the state level "exploring computer science" is a significant new introductory-level course developed in Los Angeles, with a particular concern being lack of access to rigorous computer science courses in schools with high populations of under-represented minorities.

India

Computing education is not yet mandatory in schools in India. It is an optional subject from the 9th grade (age 14) onwards. However, this scenario is changing and computing education is likely to get standardised in the next couple of years.

Greece

ICT and computer science were recently introduced in primary schools as a pilot. There is no recommended teacher supporting material nor student textbooks, so the implementation of the curriculum depends on the teachers' own skills and disposition.

South Korea

South Korea has a highly digital society and a long tradition of teaching computing in schools. At all levels of school in South Korea the curriculum contains a substantial amount on the use of computers (ICT). Many middle and high schools teach introductory material from computer science, and a proposed new curriculum substantially strengthens this component.

Information taken from computing atschool.org.uk/index.php?id=documents (International Comparisons)

THE EXPERT'S VIEW
KEVIN FONG

Kevin Fong is a TV presenter and senior lecturer in physiology at UCL

I remember the excitement of coding at the beginning of PCs – the Observer mag would run adverts for the ZX81, and there was a sense that coding was within reach. Since Xbox it's harder to see how you would code one of those games yourself. Jeremy Clarkson described the VW Beetle as a car for people who didn't like cars: previously unless you could change the clutch you weren't a motorist. Apple did that for computer science – the subconscious statement is it's cool to not know. The skill is being lost even as the appetite for it grows. Teachers used to be enthusiasts, but now there is more of a remove from that excitement. Machines like the BBC Micro were training grounds. Raspberry Pi is an interesting attempt to return to that – but it's a start, not an end.

More on page 13

'It's cool being a girl who knows coding'

Isabell Long, 17

Background I'm studying for A-levels at college in Woking. I spent four years in France, so most of my secondary education took place there.

How did you get into coding? I got my first computer at 10 but I didn't know about the coding side until I was 15, at Young Rewired State. At last year's YRS I counted three girls and 50 boys! They do talk to me – we're all interested in the same stuff. It's cool being a girl who knows about coding.

How many programming languages do you know? Three: visual basic, HTML and CSS but I want to learn as many as possible.

Should coding be part of the national curriculum? Yes, though teachers need to realise it's not for everyone. But people should have the chance to get over their fear of it. It has to be more useful than Latin!

What do your friends think of your interest? My friends are also geeks, so they understand. I'm not interested in clothes or makeup. I don't watch TV, but I do like music, I'm listening to Drowning Pool at the moment. I like metal and indie stuff.

What do you want to do with your coding skills? Something in computers, but I'm not a natural coder. I'm more of a people person, although the social animal inside me is dying slightly, the more time I spend alone in a room.

Have you ever hacked anything? I've done stuff with data which we call hacking, so ethically I have, making cool stuff out of data. But no breaking into MI5 computers! **KS**

'The fun bit is doing ridiculous things and seeing what happens'

Michael Mokrysz, 18

Background University of York computer science student

Why did you start learning to code? It was something that nobody else knew. Part of the appeal probably was that it wasn't something taught in school, though I'm sure I would have enjoyed it if it had been. I really got into it four years ago, partly motivated by money. I started doing freelance work, like making websites for computer game tournaments that ranked competitors. That really pushed me on to write good code, rather than just things that worked.

Should kids be taught to code? Within a rigid curriculum programming can be incredibly boring. The fun bit is doing ridiculous things and seeing what happens, playing with it. If I'd never gone to Young Rewired State in 2010 and 2011, I wouldn't have come anywhere near as far as I have. I owe a lot to those weeks. That made me realise there are a lot of people doing this and it is absolutely brilliant. Seeing coding being appreciated, it was huge.

What are your plans? To go my own way in the medium term, trying to build my own businesses. You've probably seen all the buzz around tech startups in Shoreditch – in the long run that's where I see myself. I'm spending the summer in San Francisco, so hopefully that will get me a lot of experience working for startups. **GKR**

'I'm proud of my riot cleanup site'

Patrick Socha, 17

Background Kingston grammar school student, London.

How did you get into coding? Playing games like *Halo* on my computer with friends who also coded. For a year or so I dabbled, and then curiosity took over.

What project are you most proud of? The Riot Clean Up website. Last summer I was working in France, and saw reports of the riots on BBC News. A woman called Sophie Collard had started a trend on Twitter to help people organise cleaning up afterwards, but there was this huge stream of tweets going by, too quickly to read. In 20 minutes I wrote some script for a basic listing telling people: "I'm going to help clean up this place at this time on this date." People would tweet their plans and upvote, so the places that needed the most attention would be number one on the list, and the abuse would get sent to the bottom. Then I started talking to the other guys from the RiotCleanUp.co.uk website and we started building a bigger site with more information.

At school, what's the attitude to coding like? I tend not to talk much about it at school, except for with close friends. People are not interested. They say: "*Angry Birds* – wow, this is a great game! I'd love to make something like that." But there's a beautiful system behind it that they don't understand. It's like a bridge – you can cross it, but the engineering behind it is something else.

What do you want to do in the future? I want to become a programmer and get into the startup world. At the moment I'm working on a site called Propelly, which lets people upload digital content they've made themselves – Photoshop files, downloads, zip files, code – and sell it through a link. It's an extremely easy way to sell content online without having to go through lots of different systems. **GKR**

GIRLS AND CODING

Female peer pressure scares them off

How do we encourage girls to get into coding? Catch 'em early, says Emma Mulqueeny

For three years we have run a big coding event across the UK called Young Rewired State . We invite people aged 18 and under to get together at a variety of centres local to them, and build something using open government data. It ends with a big overnight festival of code at the National Museum of Computing, Bletchley Park, with talks and presentations of what the young people have built.

In our first year, 2009, 50 young people took part; this time we are expecting 500. At the festival the one question I am asked regularly is: "Where are the girls?"

The answer is this: a maximum of 5% at the festival each year are girls. As the mother of two daughters – one aged 14 with no interest in programming, and one aged nine who is fascinated by it – I have been interested in exploring the reasons behind the lack of girls in YRS. These are my conclusions:

Girls become self-conscious and socially conscious at around puberty and have enough to contend with as they move to senior school, where they feel more pressure to fit in and build a new social group of friends.

Coding and digital prowess is still niche at a young age. It is often considered a bit nerdy in senior school, where it is not currently taught as a part of the curriculum. Therefore, generally speaking, those who code have taught themselves. Teaching yourself something that should really be covered as a part of lessons is a bit like doing extra homework – *why*, ask many teens, would anyone do that? There is no way the majority of teenage girls would risk ridicule or isolation by doing such a thing.

The fear of ridicule is important and is why I reckon YRS gets a greater female drop-out rate just before the event. They sign up because they want to; they drop out because they cannot face the potential embarrassment. If only they knew how heralded they would be by the achingly cool. But even the achingly cool kudos doesn't win against female peer group pressure.

So what's the answer? I hate to limit this to just the girl geek question, but perhaps in solving the problem of a dearth of female coders we can make a big dent in the broader issue of teaching programming as a part of the national curriculum – in spite of Michael Gove's vision of open education. Gove's solution of opening up the possibility of teaching computer science still focuses on senior school education, and so will still suffer the same difficulties attracting the girls.

In my opinion, Year 8 is too late. We need to start teaching digital literacy and coding as a part of the curriculum in Year 5, when most children's maths is strong enough. The curriculum has fostered a familiarity with computing and computers and the young minds are ready to start learning programming languages. Indeed primary school children are creative, excited and less likely to have developed associations, good or bad, with certain subjects.

If programming can be introduced as a part of the central curriculum in Year 5, I'd bet my last penny that by the time those kids are drawn up through the education system, you would find far less disparity between the sexes. Not to mention an increased number of talented young people with an ability to manipulate open data, relate to code and challenge one another to design and build digital products that you and I have not even begun to imagine.

So, make one simple change: teach programming in Year 5 and thereafter make it a relevant and necessary part of the curriculum. Then you'll see the girls.

Emma Mulqueeny is the co-founder of Rewired State

YOU CAN HACK IT
Here's how

Rewired State runs hack days. Every summer they organise a week of mentored programming for young people across the UK.

This year's event runs from 6–11 August. During that week businesses, small and large, around the country act as hosts to local young people, YRS alumni, Rewired State mentors and other volunteers. A challenge is set to build digital products: mobile and web, using at least one piece of open data.

It is free for any child to enter so long as they are aged 18 or under and have a rudimentary understanding of programming. The event is funded through crowdfunding. To make Young Rewired State's Code Fest happen, pledge a tenner at www.peoplefund.it/young-rewired-state/

Details at youngrewiredstate.org

THE TEACHERS' VIEW

'There is only so much Powerpoint and Word that you can teach'

Teachers will play a crucial role in achieving the transformation of the computing curriculum. And while many, not surprisingly, resent the education secretary, Michael Gove, labelling ICT teaching as dull and boring, most agree that sweeping changes need to be made to the way computing is taught in schools.

John Stout, who teaches computing to A-level students at King George V Sixth Form College in Southport, Merseyside, has campaigned for greater emphasis on coding, computational thinking and the theory behind computing for a long time. A member of the Computing at School pressure group, he says: "Many of our students arrive here having had a very unfortunate experience of ICT at secondary school because a lot of it is just teaching them to use Word."

Stout takes his 13-year-old grandson as an example: "At home, in his spare time, he uses a site called Mindcraft, which is an online game. He's creating new systems for it, setting up servers, writing bits of programming for it – stuff far in advance of what he is doing at school. Kids of his age would be fascinated by more challenging material in their lessons. They need to be learning about the theory behind computing. The story of the Turing machine, especially in this year of the 100th anniversary of Alan Turing's birth, would captivate their generation."

Amy Desmond-Williams, director of teaching and learning in ICT and business studies at Sidney Stringer Academy in Coventry, has already begun the process of changing the school's curriculum to include more computer science. She says: "There is only so much Powerpoint and Word that you can teach. It is important for students to learn office skills but in order to give them the opportunity to move on to taking degrees in computer science we need to be teaching them some programming skills early on as well. There has not been enough of that done up to now."

Teacher training is key, she continues: "When I was training only a few years ago there weren't really any courses in computer science for teachers. It tended to be ICT with a strong bias towards office skills. The subject of computing was very much linked to business studies. There is a bit of a panic now in schools. Developing new teacher training courses has to be a priority."

Mark Clarkson, head of IT and Computing at Egglescliffe School in Teesside, agrees. He says: "There was a time when ICT departments in schools were made up of interested PE teachers who compiled their football teams' results on spreadsheets and a couple of maths and physics teachers who knew a bit of coding. Now it's mainly teachers with business studies backgrounds. There needs to be massive retraining to make sure teachers can deliver different content from what they have been used to delivering. We don't so much need more teachers as teachers with different skills. We've developed a whole generation of teachers with a skillset that is not what we need for the longterm future of computing in schools." **Lisa O'Kelly**