# School B Teacher 2 - 16 Jan 2017

Speaker 1: I'm going to record ... I realise I'm speaking now. I'm going to record the interview and basically that's just for me. I will transcribe at some point. Once I've transcribed it I will probably send you a copy.

Speaker 2: Okay, that will be fine.

Speaker 1: It will probably be further in the future than I want.

Speaker 2: That's no problem at all.

Speaker 1: I won't make ... I made the mistake previously of saying, "Oh, I will do it in the next couple of weeks."

Speaker 2: [crosstalk 00:00:22].

Speaker 1: I'm sorry. We'll probably send you a transcription you can look at more less that seems right.

Speaker 2: No problem.

Speaker 1: It'll be far enough in the future.

Speaker 2: I will have forgotten it.

Speaker 1: What was that all about? I'm trying to think if there's anything else.

This isn't about judging your teaching. There are no right or wrong answers. This is far broader than that. Part of my approach has been to work with the school's we're engaged with through the Honorary Teaching Partnership. Partly because I can ensure a degree of ... I don't want to say quality but the schools that are involved in that are schools that are doing a good job generally and are interested and enthusiastic, [inaudible 00:01:14] taken as written and I can explore more interesting stuff.

I think there's a lot of schools I would go to and I'd ask them these questions and they look at me like, "What? What's commuting? What's code? What do you mean there?"

Happy?

Speaker 2: Yeah, that would be fine. No problem.

Speaker 1: Great. Let me know if you have any questions throughout. Let me know if you have an answer that doesn't fit the question, don't worry about it.

Speaker 2: Just go with the flow.

Speaker 1: Far more of a conversation.

Speaker 2: No problems.

Speaker 1: Great. Let me put that where we can see it. That's the way I keep track of how long I've been talking.

The first section is a whole bunch of new ones I should have added before but are contextual.

Speaker 2: Okay.

Speaker 1: How long have you been a teacher and how long have you been teaching here?

Speaker 2: I've been teaching at this school for just over four years. I've been teaching ... I got my QTS in 2011. This has been my sixth year of teaching.

Speaker 1: You've done most of your teaching here?

Speaker 2: Yeah. I did a year and a half at [Astoria 00:02:29] Lancaster. I was a teacher of ICT and I was brought over here to be a teacher of IT and Computer Science as well.

Speaker 1: You started as an ICT teacher?

Speaker 2: That's what it was five or six years ago. My degree's both in Computer Science ... I have a degree in Computer Science at Lancaster and I did a Masters Degree in Advanced Computer Science at Lancaster as well. From a programme background, I'm a programmer.

Speaker 1: You'd call yourself more of a programme ...

Speaker 2: Yeah.

Speaker 1: Right. Have you done anything else other than teaching? You said you've got a background in Computer Science.

Speaker 2: I've not been in the industry or anything like that. No, I've more or less gone straight from degrees to ...

Speaker 1: Why did you ... that's quite unique. Why did you decide to go from Computer Science into ...

Speaker 2: Into teaching?

Speaker 1: Yeah.

Speaker 2: Generally it was because of the frustration I think. The frustration that a lot of people that were teaching at that time when I went through the school system [inaudible 00:03:25], there weren't specialists. They had an interest in the subject but not a passion. With the way Computer Science is going at the moment and over the past few years technology has come into a lot of things, you need some of that passion to actually go for it. I show the kids how it actually works.

I think, to me, it was more a frustration of why is this not happening and the answer really was, "Do it yourself" and that sort of thing. Go into the teaching ...

Speaker 1: That's absolutely right and I think it's really interesting ... it would be great if more people did that. I think it's a real problem. I've recognised it in that there's so many ... across the economy, Computer Science is one of those places that there are plenty of jobs. If you want to go into the industry, it's very easy to go into the industry.

As I think with Physics for example. If you don't get a great degree in Physics, you may struggle to get a job so then you can go and get a teaching degree [crosstalk 00:04:28] and go into teaching with a background in Physics. You may not be the best physicist but you still have that grounding in Physics.

I think that process isn't as common in Computer Science. I do think it probably means that people like yourself are probably better because you're more passionate about it.

Speaker 2: Absolutely. You can see where kids go wrong a lot quicker.

Speaker 1: It does mean there's a lacking of people with a good grounding in Computer Science in the teaching profession. You see a mixture as well.

I don't know if you know P– down at School A

Speaker 2: P–

Speaker 1: Yeah. His background is very much ICT. He's got a passion for computing. There's no question that he's good at that. His background is still ICT, not computing. It is a different approach, whereas Mr N– is far more of a computer scientist in different ways.

What's it like teaching here in general? This is not an inquisition. How's that changed over time? It'll help me get a context because I'm not here.

Speaker 2: The school's a nice school. We've got a lot of what we call, "Higher performing kids", it's quite an academic school in the way it does things. We tend to have kids that are slightly lazy I would probably say, if that makes sense. They want the answers instead of having to find it. [crosstalk 00:05:47]. Absolutely.

Speaker 1: ... having been there.

Speaker 2: What you find a lot of challenges are trying to get things to work through themselves or they go, "That was too big a task." I'm going sort that [inaudible 00:06:00] trying to break it down, decompose it into the different parts that you need to do and then once you've got your basics, start to build it back up. That's kind of the approach we take a lot more with our kids.

Speaker 1: Do you feel like with those sort of higher performing kids sometimes it's ... with something like ICT and computing, they're used to getting the answers quite quickly almost instinctually.

Speaker 2: Yeah. With the more complex it becomes, it's not a straightforward answer. Especially when we sort of ... this sort of way instead ... which way's better and why and that's when the kids start to, "Oh wow, there's more than one answer." It doesn't scare them, they just sort of go, "Wow, this is a lot better."

Speaker 1: Very different though. Especially if you're thinking about kids who are very good at math. [crosstalk 00:06:46].

Speaker 2: A lot of the kids are getting a more understanding of what Computer Science actually is over the past four years I've been here. If you have a [inaudible 00:06:55] three or four years ago, you've probably got a few kids going, "It's mostly about computers isn't it?" [inaudible 00:06:59] computing. That's gotten a lot less now. We've got very few kids that will now think it's [inaudible 00:07:05] because they know exactly what's expected.

Speaker 1: Do you call it Computer Science? Do you call it computing?

Speaker 2: It depends on the [GCSE 00:07:12]. At the moment the [inaudible 00:07:15] are doing PC computing. That finishes this year. New syllabus, new specs. Different grading systems from next year. The year tens and below, they've been learned it as Computer Science because that's what their [GCSE 00:07:28] would be in.

Speaker 1: Right. It's one of the things I didn't fully appreciate until I started this project. Although there's the statutory curriculum from key stage one to key stage three-ish and then key stage four a bit, they [GCSE 00:07:47] structure has a huge influence on everything below it.

Speaker 2: Absolutely.

Speaker 1: That's quite school dependent because it depends on what [GCSE's 00:07:54] you're choosing to push kids through.

Speaker 2: A lot of the kids go into year seven, I think they're doing ICT. That's probably the misconception from primary schools. They start to understand IT is part of Computer Science. This is the IT part. This is the computing part. This is the literacy part, [inaudible 00:08:14] literacy because there's three strands that we try and encompass through that. On their timetables, year seven, eight and nine, it's ITM computer.

Speaker 1: So it's both?

Speaker 2: It's more the way we just teach it. We say, "This is our computing part. This is our more IT part." The IT part is getting less and less because we only get on lesson a week with the kids. We have to really structure our key stage three curriculum. [crosstalk 00:08:41].

Speaker 1: ... it's something you see quite often is that ICT is getting less and less. It's not because necessarily those skills are less important, it's just the way the priorities are and what's being delivered is the Computer Sciences is emphasised more, also I think more difficult to teach. You end up spending a long time on something like [Python 00:09:11] because it's quite hard to teach and then you've used up all the time.

Speaker 2: That's right and you lose everything then half way in.

Speaker 1: I think the other thing that probably will change in time is that certain languages ... all languages, some kids get them, some kids don't. You can only move as quickly as your slowest kids.

Speaker 2: That's right.

Speaker 1: You get stuck delivering something or some of the kids [inaudible 00:09:39].

If, in any way, how have the overall policy changes affected you in your teaching role?

Speaker 2: In terms of computing?

Speaker 1: Yeah, I've left that broad because it's recognising part of introducing computers, part of a broader educational policy change. It was a whole new curriculum. That affects you guys at secondary, far less than primary ...

Speaker 2: Definitely.

Speaker 1: ... primary schools, they're delivering all of it. I just figured it's worth throwing out there and getting it in context. Has that broader change affected your computing ...

Speaker 2: I'd say with government policy change, I would suggest they probably try to make Computer Science not too difficult but they've tried to go all guns blazing; "This is a very academic subject. We want the best of the best." In some ways, trying to prove a point it's a hard science ... it always is a hard science, but they seem to be pushing the certain [GCSE's 00:10:43] to that extra limit that you wouldn't normally expect.

An example would be something that we've probably seen in our A level teaching has now gone down to [GCSE 00:10:51] and that's to give you an example of boundary shifts. Left and right boundary shifts, that would only have been in our year 12 or year 13 course. That's now gone down to our year 10s.

Speaker 1: When you say, "Hard science", do you mean difficult or do you mean ...?

Speaker 2: Rigorous I would probably say. They're trying to make it this fully rigorous qualification in its own right. They're trying to go away form what the ICT is and make it into a different thing. In some ways I can see why they've done that because the old, old ICT course is ... the OCO Nationals, things like that, they were kind of ... not pieces of paper being handed out but they were quite easy to get the actual qualifications. There did need to be a standard of being able to be competent and being able to actually get a piece of paper as well, if that makes sense.

In some ways it's good [inaudible 00:11:48] recognise that it is an important science to work with but in some ways it probably just goes a little bit over the top in the way they've changed the policy. They must have this, this and this.

Speaker 1: I think that term, "Hard science" is actually really interesting. I can think of three ways of interpreting that. All of them fit the situation.

I think you're right, there's an emphasis with it being rigorous. There's an emphasis on it being difficult and also an emphasis on it being more like a physical science than a social science.

Speaker 2: That's right.

Speaker 1: I think it's one of those difficulties with Computer Science in general that actually when you drill into it a little bit more, it's somewhere in between.

Speaker 2: Yeah.

Speaker 1: There is a degree of physical science and electronics in there and you need to ... to understand the binary stuff, you kind of need to understand the physics, which I don't. How a transistor works is quite complicated really, but the way a code works and what we generally think as ...

Speaker 2: The theological side isn't it, your math, not everything else.

Speaker 1: Math and a bit of social science because each language is written by a person.

Speaker 2: That's right and it's a language that you're learning.

Speaker 1: I think teaching a language as a science is quite tempting but also misleading. To forget that somebody wrote all the high level languages. As soon as you're no longer working on the machine in code and binary, those have all been written so they're also routines that somebody's [crosstalk 00:13:20] routines. That's quite a hard message I think to get across.

It's missing from the emphasis on [inaudible 00:13:27] level. Possibly because the people that wrote it on a quality level ...

Speaker 2: That's probably the biggest one for us from how the policies have changed, from how the examples have changed, they probably made it more rigorous on our controlled assessments. The actual programme and task they have to do is part of their qualification. They're making it a lot more tighter and I suspect that needs to stop. Professionals aren't quite sure how to do the test or how to run the test to stop them from explaining the answers and handing the answers out.

I put more emphasis on those to make sure other kids are in test conditions, which they always are. Our new course, we have the ultimate threat; we may have an examine or visitors at any time to see what we're actually doing with our kids on the control assessments, which is good in a way because it makes sure that every single school is then following set rules. That added bit of pressure as well, I guess, to make sure that everything is absolutely ...

Speaker 1: Does that require you to kind of teach in a different way?

Speaker 2: Yeah. It's making sure they can actually programme and be able to programme independently by themselves whereas I guess the key bit for other people might be to teach them the skills they need to do the tasks. You can't do that. You've got to teach them how to be a general, good programme.

Speaker 1: General, good ...

Speaker 2: That is the hard bit. We've only got a year properly to get them from a novice to a [GSCE 00:15:04] standard.

Speaker 1: That's another word that comes up a lot is, "Programming" and because as we've already touched on, Computer Science and Programming are separate. It's not a programming curriculum, it's a computing curriculum.

Speaker 2: That's right.

Speaker 1: I feel like that ... again, it's not clear in the document, which is then ... I think there's way of delivering that's far more Programming and Computer Science. What do you ...?

Speaker 2: The emphasis at the moment is more on the programme side. There's some theory that we touch on, but it is much more the logic creating an algorithm, decomposing an algorithm, or actually [inaudible 00:16:44] it goes through. That is very much the focus at the moment. The kids will do ...

Speaker 1: As someone who has a background in Computer Science, what do you think?

Speaker 2: You mean both don't you? It's like the chicken or the egg. In that syndrome, you need to have a good programme and knowledge to see how you can experiment. As a science member, you have to write your programmes so you can experiment what's actually happening with the hardware or why is the software doing it this way.

You also need the technical knowledge of how the software works or how the computer's displaying that in different ways. Just trying to make sure you can marry the two together at the right level for the kids.

Speaker 1: Yeah. Also finding that mixture, I think Programming, you can teach as an ICT almost, as a skill.

Speaker 2: You can. Again, it's done as a competency of the teacher of what happens when a kid gets stuck. That's probably our hardest bit. I could teach any teacher how to programme but if they don't actually do the programming themselves, or they don't go a little bit further ahead of themselves, if the child gets stuck or they go a little bit further ahead and then get stuck, it's that [crosstalk 00:16:59]

Speaker 1: Do you have kids who go off and learn other languages independently?

Speaker 2: Yeah. We've got kids who have done A level who have done Java, who've done C, who've done Python. From previous years kids [inaudible 00:17:15] and Ruby languages.

Speaker 1: The ones who do the [GCSE 00:17:23], they often will go ...

Speaker 2: Go off and do their own languages before or after.

Speaker 1: For some teachers that can be quite challenging because those are languages ...

Speaker 2: [crosstalk 00:17:36] or anything like that.

Speaker 1: To start with, they're not aware of them and then they don't know anything about them. How do you find that?

Speaker 2: With most languages I'm all right. Then again, I've done programme at three levels so the idea when I did my programming at Lancaster and [inaudible 00:17:49] will teach you proficiency in Java and from that, you should be able to teach yourself or the languages.

We still have lectures in C, the C family of [inaudible 00:17:59], but the expectation was you've got to finely research those. It's not that sort of skill that you go off to. I never taught visual basic, or looked to these [inaudible 00:18:08] here. It all started in a week. You use previous languages. Swap the syntax over and you're on your way.

It's a hard skill to learn. You're not always going to be perfect, but it's always going to make sure ...

Speaker 1: These are one of those things I think is probably not enough teachers have. Also, I think is not necessarily being taught, is not to be whetted to a language. Learning how to ... learning the basic underlying structure. I think it's hard to identify what is the best language that will ... that you can get proficient at that language but also have a degree of proficiency in some other things. Proficiency at that general programming. What is general programming? I don't know if there is the perfect language yet for teaching.

Speaker 2: There probably isn't because there's all sorts of different pros and cons for each one. What we try and do with our key stage three is we'll show them in year seven and eight now, how to be basics in Python. We'll extend that and they've got an idea how the Python language, which is microbits for example. Once we get to your name, we're going offer you a different text language for making [inaudible 00:19:23] advantages of using visual basic because the [inaudible 00:19:27] to that, we've got the pseudo ...

Speaker 1: How many languages do you think a kid here would get introduced to?

Speaker 2: In [inaudible 00:19:35]?

Speaker 1: Yeah.

Speaker 2: Two text based languages and if they chose to do [GDSE 00:19:41], we'll teach the same language programme as well as part of the ...

Speaker 1: That was a contextual bit? It ran on far more than it ... we touched on another subject.

Speaker 2: Yeah, sorry.

Speaker 1: That's all right. That's my fault. That's the way I do interviews. I quit doing interviews because of that. Then I look at my sheet and think, "Oh, shoot. We're still on that. Are we still there?"

I'm assuming you've probably heard the term, "Computational thinking." What's it mean to you and what do you think it's meant to mean and what do you think the role of computational thinking is within the computing curriculum?

Speaker 2: Computational thinking to me is going outside of the boxes to actually think of how the computer will programme or would actually work. That doesn't always follow the correct logic I guess. You expect a computer to think this way or that way but actually in reality it's a completely different way. That's the way it's being built of actually put through.

The way we try and teach that sort of thinking is trying to get the kids to think the basic problem they can do. It's exactly what a computer would do, take it down to it's extreme sort of problem and then see if we can build it up from there.

Build a programme that does the following, what's it actually asking us to do for the first bit. Strip it down to its basic; means available, means of input. What does it need from that [inaudible 00:21:06] statement? It might need a loop. We're down to a list now and then they can see it goes through. In a way it's like taking a recipe book or something like that and doing a step by step of what do we need to do.

Speaker 1: Do you feel like computational thinking is quite ... it's almost something you pull from the different sources.

Speaker 2: Yeah you do. Not just from Computer Science, from the math side and from the computing side of it, step by step. The hardest bit that kids sometimes struggle with is that computer will literally go step one, step two, then step three, then step four, then step five. "Hang on, we've missed a step. Why has it not done the following?" It's making sure everything is accurate and specific as possible.

Speaker 1: What do you think the role of computational thinking is in the curriculum and also how hard do you think kids find that?

Speaker 2: It's all about practise for the kids. The more they do, the more confident they become and the easier it becomes for them. At first if you gave them a problem and said, "Let's see how we can work through this." They go, "Can't do it." Once it starts to look like it messes up, we taught them with the computational thinking, it's the idea of strip it down, what can it do? What have you learned before from this? They should be able to find solutions from [inaudible 00:22:25] through.

Speaker 1: Do you think it's a process applicable to other areas or do you think it's mainly useful when you're using computers?

Speaker 2: I think it's useful in other areas, definitely math. You might have an algorithm ... some [inaudible 00:22:38] disorder typically and then we could break it down into the following components to actually work it through.

Certainly Science as well. Definitely Chemistry and Physics with mechanisms and things like that. You have to follow processes throughout.

Speaker 1: That is a relationship, I think complicated. The relationship between Computer Science, Physics, Chemistry and Math. I thought it was really interesting what Wendy was saying about how Oxford and Cambridge say, "We don't want Computer Science, we just want ..."

Speaker 2: Yeah.

Speaker 1: I know a friend of mine who you might know actually, a guy named D–.

Speaker 2: No.

Speaker 1: He used to work in [Weakdon 00:23:21] and then went up to work to be a developer and now works in Egypt, but that's a different story. I want to say his background was in Physics and Math and then he became a games developer before he was a teacher. Nobody wanted any ... at that point it wasn't, "We want a background in computing", it was, "We want a background in ..."

Speaker 2: ICT.

Speaker 1: It was ICT but also math, the sciences. If you're good at the sciences you go on to computing. It's interesting to see ... I think that's a new relationship that's developing for [inaudible 00:23:57], how they interact.

What do you think the main purpose is of switching from ICT to computing as a broad subject?

Speaker 2: As a broad subject? I think it's making the kids better problem solvers, much more active on the problem solving.

Speaker 1: Do you think that's the hope or do you think that's what's happening.

Speaker 2: That's what's starting to happen. They're no longer just using the computer. They're using the computer as a tool now to develop something or to make something better than what we've actually got on the computer.

It's kind of like driving a car in a way I guess. The old idea with ICTs, you'd be using the car, using what's already out there. Now with the idea of computing we're actually helping the buyer of the car see what's actually inside and how can we make that better so it can become a tool you use to make a better sort of product.

Speaker 1: Do you think that's necessary?

Speaker 2: Necessary for the [GCSE 00:24:56]? Yes.

Speaker 1: For the kids.

Speaker 2: For the kids, I think it's giving them better life skills but I think the problem we've got now is that we're taking away the core life skills that they should have before they leave school.

Speaker 1: What do you think those core life skills ...

Speaker 2: Core life skills, being able to use a computer correctly and efficiently.

Speaker 1: Okay.

Speaker 2: Every job that you go into these days has some sort of computer. As well as knowing how computers work, which is the Computer Science side, you also need to be able to know how to use one competently as well.

Speaker 1: You think they're losing that?

Speaker 2: We're still doing it to an extent but it's a much more streamline process than what we used to do.

Speaker 1: Do you think that should be fitting in somewhere?

Speaker 2: Somewhere it should be, but the problem schools have is time table and time.

Speaker 1: Do you think it has to be taught discretely, those general computer skills? Do you think they could just be, in theory ...

Speaker 2: Used slowly some sort specialist I would say. I think the problem is if you gave a general teacher to teach, it would be their ideas or what they're competencies are in using general computer systems and that varies so much. It's not fair to the child. That's like asking a music teacher to teach math because there's discrete math in music, you would get a variety of different [crosstalk 00:26:16].

Speaker 1: You could teach math through music and there's a lot of mathematical ...

Speaker 2: Absolutely but you get a different [crosstalk 00:26:22] wouldn't you? Of different things.

Speaker 1: Yeah. What are my questions to you? Do you think ... this is kind of what we were just saying actually. Do you think the people will use the skills and concepts they're learning through the computing curriculum in the future?

Speaker 2: Certainly for problem solving. Certainly if they've got a big task to do when they're working a job and can break it down into smaller jobs. I think that is going to be a key skill a lot of children will pick up.

Speaker 1: Is that a computing skill?

Speaker 2: It's used in the computers to work it through.

Speaker 1: I've heard some people say at different points that actually it would have been better to call computing, "Problem solving" because a lot of the general computing ...

Speaker 2: With the way it's going with the programme, yes you could say it's problem solving in the computer. The theory side, the technical theory, that would be people going into a computer career but apart from that it's probably ... it's going to give people an idea of how to use computers but it's not going to let them fix their computers or anything like that. That's the academic side of the Computer Science.

Speaker 1: That's the way it comes up often when I speak to the kids. In some way they will say something about wanting to be able to fix their computers, either hardware or software. Sometimes they say, "I'd like to know more about getting rid of viruses." You do feel like they'd quite like to know that but they don't.

I wonder if there's something about that car analogy which has come up before and I think you're right. To a certain extent, we're teaching how the car works. Maybe it's more teaching how the engine works but not teaching people how to change the oil.

Speaker 2: That's right, how to change the oil. That's why when they made the curriculum, there's the three strands, the IT, the computer and digital literacy. Most schools are either very heavy computing or very heavy IT. It's the crossover which is where does that go, who teaches it and why.

Speaker 1: I feel like the statements within the digital literacy aren't as clear.

Speaker 2: No. It's been secondly thought of as let's put it in quick. That's the way it comes across ... "Let's just put it in quick." The problem is, where does it go? Like you say, "Who's going to teach that?" Is it going to be the computing staff or is it going to be the PHSE staff? You can take it from either way as to ...

Speaker 1: The term I've started to use instead of digital literacy is digital citizenship. Again, there's a question of ...

Speaker 2: Citizenism. Yeah.

Speaker 1: Who has the knowledge, the skill and the time to deliver how to interact with public services online in a digital way? There's a big push for public services to be accessibly digital. How are people being taught to access those services? We can talk about the risk of things like digitally voting.

Speaker 2: Yeah. I guess the issue is there's no qualification at the moment. There's no [inaudible 00:29:41] on schools wanting to embed that into the curriculum because there's no ... it's not that there's no need, there is a need. It's concerning what's going to happen with that time with the kids, [inaudible 00:29:54] had qualifications.

Speaker 1: That's really interesting what you said there in a lot of interesting ways. It shows how job focused the time table tends to be and how really not just in computing and this is somewhere I started and then moved back, so much of it is really focused on employment skills.

I think in computing, actually what I'm finding is that it's certain employment skills. It's certain kind of ... the computing curriculum has been created by and for certain sectors, which aren't necessarily the general sectors that a lot of kids are going into but the ones that have screamed the loudest.

Speaker 2: That's right. I guess, or the ones that never used to get as much of a look as sort of the sectors.

Speaker 1: I think ones where there's been a clear gap, but I still think we have to remember that however many jobs there are in computing and Computer Science, only a small percentage of our pupils will go into those jobs.

Speaker 2: Absolutely.

Speaker 1: They have the interest and because they have the skill and because there are still ... it's still a number.

Speaker 2: Absolutely.

Speaker 1: It's interesting to realise that the less job focused skills in computing are getting much less of an emphasis.

Speaker 2: Definitely.

Speaker 1: The other one that came up today a little bit and I've thought more about cyber security and teaching good cyber security skills. I thought it was interesting and this isn't a criticism at all, but the kids haven't made the connexion between the cryptography lessons and cyber security. They were seeing the cryptography stuff and saying, "How am I going to use this?"

For me as somebody who's emersed in this area, I can say, "It's basic cyber security, understanding cryptography and yes you're right you won't use that."

Speaker 2: The idea is ...

Speaker 1: How it works, but they aren't making that connexion. Again, that's probably one of those things that is a general broad scale. No matter how good your cyber securities systems are, you need ...

Speaker 2: A basic understanding?

Speaker 1: On the human side it's probably the most important.

Do you think that the computing curriculum requires people to be different or think differently in any particular way?

Speaker 2: Not necessarily think differently, I think it's the way they approach the subject. I think they need to be confident in skills that are being applied. Even though they're thinking in a different way they have to break it down, I think it's the kind of being more confident in actually breaking it down. I think for our kids, it's applying the confidence with them.

Speaker 1: Do you think it changes how they approach other things in life at all?

Speaker 2: As they go through, possibly. Again, I think the more they do, the more changes they'll go through with the different skills they've got.

Speaker 1: What do you mean? I think I know what you mean but the more they go through [crosstalk 00:33:14]

Speaker 2: The more they go through school, not necessarily the curriculum, but the more they can certainly see breaking it down is quite the useful thing for this and it's quite the useful thing for that or I've got 15 tasks to do for general work. I can break them all down into human goals.

Speaker 1: Do you think there's any areas where computational logic as far as a form of problem solving is less useful or do you think it tends to be one of the better forms of solving problems?

Speaker 2: I wouldn't say it's less useful, I think it shows them a different way of general thinking. I think it also challenges them to find different ways. Unlike if it's being taught as a mathematical subject, this is the way so do it. It could be, this is your better way of doing it.

I think with the way we do it, with logic. We can do it this way or we could inverse it and do it a different way. An example could be using [inaudible 00:34:14]. We could then ... why don't we use a wild loop and inverse the logic. Why do we inverse the logic? Is that any better? Is that not better?

Then going to the scientist side, let's look at the logic gates on this. Which is cheaper, then we're going to go that way. Then they can suddenly start to see, "Oh yeah." The exactly same thing, just a different way from ...

Speaker 1: A different way. I think that's more a strength in computing [crosstalk 00:34:39]

Speaker 2: The first way I go, "I see" and completely the other way it actually works for you.

Speaker 1: Definitely

Speaker 2: Those different things.

Speaker 1: That's one of those strengths of computing. Again, I wonder whether it's getting [teased 00:34:49] out enough is that in computing, there are generally multiple solutions.

Speaker 2: Yeah.

Speaker 1: I think it makes it a difficult area to teach and also very difficult area to assess. I think from what I've gotten from speaking to people like yourself about those [GCSE's 00:35:06], it's not there yet. Being able to appreciate there are often ... that you could set a task and a kid could come in and probably find a solution that nobody had ever seen before.

Speaker 2: That's right.

Speaker 1: How do you assess the quality of that brand new solution?

Speaker 2: That you got through.

Speaker 1: Also, how do you assess that process in a way that's equivalent to somebody else who's done it the way everybody else has done it the way you're expecting. You can say, "Yes, they've done that and they know that." I think that's one of the challenges of computing.

Speaker 2: Yeah certainly. I think as well, with the way doing the thinking when they start to see it being broken into subs or many functioning programmes that they've made themselves. They can see specific things. That's the part of computational logic and the computational thinking of this program's just doing the specific thing and that is being called from the main [inaudible 00:35:59] to do that so they can slowly start to see how different ways can actually put it through. Again, that's the basics of the specifics of just one thing I guess.

Speaker 1: What do you think is the most ... we have answered this already but, what do you think is the most important thing you teach in the computing lessons? If they took one thing away, what do you think ...

Speaker 2: If they took one thing away, you have to break things down, being able to be confident in how they do things; either programming or ethereal.

Speaker 1: If they can break things down, then that it.

Speaker 2: Yeah, they can see different ways and ...

Speaker 1: And see different solutions.

What part of the computing curriculum do you think eats up the most time?

Speaker 2: Eats up the most time? Probably the programming side.

Speaker 1: Do you think that's right or do you think ...

Speaker 2: In some ways it is. In some ways it's not. I was going back to when I was doing things at Uni and the way they taught it, it was more or less probe and probe and probe and then it went into theory.

You need the programmes to be able to do the experiments in the science side of Computer Science. Unless you're happy at writing small programmes, you're going to struggle with the advanced side of the theory. It [inaudible 00:37:20] the most time but it's one of these strange things, in a qualification it's the least amount of percentage and well. The programme exercises that they do are worth 20% of the final mark. The theory is 80%. The problem solving is 40% of the 80%; they have to be able to do the programme, do the expert to see how the problem solving side is from it.

Speaker 1: They need the programming to do it, but it also takes up a lot of time.

Speaker 2: Right. It's like [inaudible 00:37:50], you're going to get marked on the report but you need the programmes to write the report. You can't do anything until you've actually go the programme. Before you can actually do the actual thing you're getting marked for I guess.

Speaker 1: That makes sense. It's sort of a mixture.

Speaker 2: Yeah.

Speaker 1: If you could change some aspect of the computing curriculum, what would it be?

Speaker 2: To change it? More an emphasis on life skills as well as the Computer Science side.

Speaker 1: Do you make that distinct from ICT?

Speaker 2: Yeah. Life skills, using the computers efficiently, competently and being confident to use a computer.

For some children that could be turning the computer on to do some office space work or office space scenarios. For somebody else it could be using the computer to manage stock control. It could be whatever their [inaudible 00:38:51].

Speaker 1: If you could only teach one aspect of the computing curriculum what would it be?

Speaker 2: Probably the programming side. From what we said, you can't do anything without that.

Speaker 1: Other than the things you've mentioned, is there anything else you'd change about the curriculum?

Speaker 2: I think we need to be more specific in the curriculum about what we're trying to actually do. There's vagueness in there so it gives you a bit of a [inaudible 00:39:33] I guess, but equally teaching, it's a complete different [inaudible 00:39:39] places [crosstalk 00:39:40] consistency of [inaudible 00:39:42] yet.

Speaker 1: Absolutely. That's interesting. I always read that question and think, "Oh no, there's no way there's going to be ..." We've already answered this and there's always something ... I think, "Oh that's why I asked that one."

The next section is about evaluation. Again, like all of it we've covered some of this already. The point of the section is to think about how you feel as a teacher, you're being judged on the computing and how that's changing your practise and impacting on the kids. That's sort of the general idea of it. That's what I'm trying to ...

Speaker 2: Okay.

Speaker 1: That's all right, it just means I've recorded you saying, "You're nipping through." That's all right, don't worry about it.

How do you feel like you're judged and evaluated on your deliver of computing?

Speaker 2: Delivery?

Speaker 1: Yeah. You've already covered a lot of this stuff.

Speaker 2: A lot of it is showing that the kids make progress. What they know before they started the course or topic, what do they know after they completed it.

Speaker 1: Do you think there's any particular areas where that is easier or harder to demonstrate?

Speaker 2: Problem solving is probably the easiest way, we have to write a programme and something else like that. On the theory side, some of the more theoretical components are a little bit harder to be tested by, there's no other way to actually put it through and then it's the kids' interpretation of the questions, which give you the ...

Speaker 1: Do the kids do ongoing tests and quizzes?

Speaker 2: We use things called, "Cahoots." It's like a multiple choice question. Fastest finger first. They get points when they get it right and they'll get points when [crosstalk 00:41:33].

Speaker 1: Are those given every [crosstalk 00:41:35].

Speaker 2: They're given for most of mine, they'll be given every lesson. That would be on a new topic or on a previous topic. What they learned from a previous lesson but it gives me a breakdown as to see who ends up and is right and who's wrong so I can pick a problem out later on to actually work them through.

That gives me an ongoing progress. We ask the kids every week to rate themselves. Have they met what we're after in our learning objectives? They take that to say, "Yes they have" or, "No they haven't." Again, we can see who's met our objective and whose not. It's just finding the evidence from their work to actually prove that they have met it.

Speaker 1: Is that ongoing kind of assessment almost.

Speaker 2: G to C level, it's all down to what they're targets are. That target grade is calculated on [CAD 00:42:24] scores, it's all dependent on us to what those scores are, what their levels of progress should be. It's also based on their key stage two levels as well. [inaudible 00:42:34] what do you use? Do you use the math? Do you use the science? To use the English. That's the hard bit because you tend to want to use the math because it's as close as we can get but that doesn't necessarily mean there'll be a fantastic computer scientist with the theory side. Or you've got the more detailed questions, explain this or explain that which the math then struggle with.

Speaker 1: I saw that when I was doing classroom observations between ... which were at [inaudible 00:43:01] where they are [inaudible 00:43:03] by their math scores. You can see that particularly the difference between the top and the mid-set there is a lot more grey area than you'd think. There are some kids in the top-set who clearly just get it and that's fine. There are other kids there who are struggling and then equally in the mid-set there's kids who are in a computer science sense, operating just as high as the kids in the top-set. You can tell it's not a perfect proxy. You kind of do the best you can.

In terms of evaluation, what do you think are the priorities of your head teacher? What's your interpretation of the head teacher and [inaudible 00:43:54] priorities for computing?

Speaker 2: For what [inaudible 00:43:59], we'll probably use [GCSE 00:44:01] and all the results. That's probably our key thing. It's not how many you've got in A to C or nine to one, it's very much how many kids have made at least expected progress.

Speaker 1: Okay.

Speaker 2: We will get traffic lighted on that using the [OPs 00:44:21] tables. We want to be as close to the red values, which is the top of the scale as possible.

Speaker 1: Expected programmes. Do you think there's anything specific or do you think kind of an evaluation it's really just about our kids progressing and then what they're progressing in? That's really far more down to you?

Speaker 2: [inaudible 00:44:42] yes. The [GCSE 00:44:42] in that qualification, so they must be competent in all areas of the computer science part. Key stage three, it's probably much more of making sure we are following the computing curriculum as close as we can, making sure that it is ...

Speaker 1: In a funny sort of way, especially someone like yourself who has a background in what you're saying. You have a back curriculum and you probably have a very good understanding of it. Do you feel like you have to explain that to other people and say, "I know it looks like it means this, but actually this is what it means."

Speaker 2: Out of the department of three, two of us are pretty competent in Computer Science. Once they've been shown how to do the different parts of the science, they become more confident, actually delivering it. At first, they can be quite negative, "This doesn't look good. It's new."

It's that new skill. Kids do the same thing. We're going to teach you how to programme [inaudible 00:45:46]. The more we do, the more they see what actually works for you.

Speaker 1: It's such a weird proxy to speak to six kids and get a sense of your group. The six kids I spoke to today, you could tell there were some of them that ... there was one lass who said she wanted to be a potter, wanted to be artistic, but you could also tell there was a part of her brain where she said, "Actually, this Computer Science stuff really makes sense to me."

Speaker 2: Yeah.

Speaker 1: I think that's really interesting to see that in ... I think it says a lot about the Computer Science and also how it's being taught that somebody who probably on an initial spot [crosstalk 00:46:25].

Speaker 2: Absolutely.

Speaker 1: This girl said she was actually choosing it as a [crosstalk 00:46:32]. I thought, that says a lot about what's being taught here. Someone like that can see the possibilities where I think it's possible to teach it in a far more constructive way.

If you were ... what's the hardest aspect of teaching ... we've already covered that. We won't cover that one again.

If you were asked to observe somebody else, what would be the key thing you'd be looking for? If you did an observation of another ...

Speaker 2: If I was observing the teacher, if it was on programming, I'd be looking at how they can deeper programmes with the kids. That in itself is a skill.

If it was in theory, I'd be looking at making sure they're using correct terminology. It's very easy in Computer Science to say one thing, use the wrong word and it means a completely different thing.

Speaker 1: What example?

Speaker 2: An example, let's look at processor. Common misconception is that it's the brain of the computer. That's partly true, though that's not entirely correct. The processors job is to process all the instructions, perform and execute cycle; that would be the sort of difference I'd be looking for a teacher to pick up on. The kids go for basic, "It's the brains." It's kind of the brains, let's see if we can get a better definition for it.

Speaker 1: Do you think that subtle terminology is something a lot of teachers get or do you think ...

Speaker 2: I think that makes a difference between a good and an outstanding teacher if that makes sense.

Speaker 1: It makes a lot of sense.

Speaker 2: Or having that competency of they know what it says in the text book but can they actually push that? Can these real world examples of something happening and then the big one, can they take the most technical and can they actually get it down to a level that the children can understand?

We've got a lot of people that are in teaching that are very technical and that's fantastic but if they can't explain it in a different way, then we've not got a chance to actually put it through. Think about translators; what does a translator do for a lot of the language? Can we get that now to be understood by the kids? You look at how a translator works between someone speaking French and English, how does that happen from one [inaudible 00:48:58] to the other and then we can start to see what he's doing now. Then they can put the definition [inaudible 00:49:04].

Speaker 1: That's the difference between a very good teacher?

Speaker 2: Yeah, that's what I would say.

Speaker 1: And having that good sense of that theory.

Speaker 2: Equally I think, the teacher's also got a thing [inaudible 00:49:19] the answer, you can't make it up. Not in Computer Science. So many try to go technical or try just trying to [inaudible 00:49:25] the kids off and the kids don't appreciate it at all. The kids are a lot more happy if you just go, "I don't know about that", or, "I've not tried that."

Speaker 1: Again, something I think is a skill that computing teachers need to work at is the sign posting. I think someone like yourself has a real advantage but other teachers who don't have a background in computing. You have to learn how to sign post.

When a kid comes to you and says, "I've written this in [Ruby 00:49:52], you don't say ...

Speaker 2: What's Ruby?

Speaker 1: "What's that? I've never heard of that." You have to find a way of kind of sign posting them so that they're still getting support but they acknowledge that you're not supporting them directly but you're sign posting them to kind of continue to be inspired in that area. It's not just languages where that is, but languages are a particular place that you [crosstalk 00:50:20], because there's so many languages available. Entry level for some computer languages, like Java, is so low that ...

Speaker 2: Yeah.

Speaker 1: It's not unreasonable for 15-year-old to decide they want to learn Java and come back after the Christmas holidays and say, "I've learned this. Here's what I've done sir."

Speaker 2: I think most schools have gone down the Python route because there's so many resources out there with Python. If you're not competent, you've got stuff to start you off. That's fine to an extent but then they struggle with that a little bit because you need a bit more than Python, you need stronger ... I guess the problem with the Java one is if you start to teach and you're not sure of teaching it yourself, you could get into a very big mess very quickly.

Speaker 1: Yeah, definitely.

Speaker 2: So many universities actually have said, "We'd rather you don't teach Java at all so we can teach it our way." She can understand this way, it isn't an easy language to actually go into.

Speaker 1: Absolutely. What do we mean by easy language? I think that's another thing where many languages are easy when you start and get very complicated, very quickly. I was talking to one of the tutors at Lancaster who said, "We've now moved C to a second year thing because actually it gets complicated and it's hard."

One of the really interesting things that's very ... and you'll appreciate this probably more, St. [Adens 00:51:55] where it is and here, up here, geographically you look at them and they're very similar schools. I think that you have very different kinds of kids. You have a far more, in some ways, rural economy down there. You have fewer of the kids who are necessarily getting support at home or coming in with new languages. There are some but I think there's fewer.

Whereas up here, from Wendy was saying, you've got a lot of kids who have stem parents.

Speaker 2: Because of the area that we're in.

Speaker 1: Yeah. [crosstalk 00:5:30], necessarily on a map guess that.

Speaker 2: I guess the options you've got, you must do the Computer Science or the ICT because it is the scientist side and you can understand that quite a lot.

Speaker 1: The parents are giving that emphasis, not the kids?

Speaker 2: All of the kids want to work in the power plants and they are after the robotics and things like that.

Speaker 1: I think that's such an interesting difference and not necessarily ... I think you'd expect it if one school was rural. Geographically, they're very ... they're not distant schools. Size-wise they're not dissimilar either but actually because of that context and what's going on around you.

Speaker 2: It's a little different.

Speaker 1: Very different kinds and different challenges for teaching.

Speaker 2: Very much so.

Speaker 1: There was one girl down at School A who said, "I've learned from computing that I never want to have [crosstalk 00:53:19]. I want to be a farmer. I'm going to be a farmer and I know I can get computers in my tractor but I don't want them. GPS and that's it."

Speaker 2: That's fine because they know what they're after.

Speaker 1: It was so clear. She had decided that she didn't get it and she was going to avoid it.

Speaker 2: I think the only thing the kids have to be aware of now is before, four years ago, you had to do some sort of technology qualification. That was the curriculum. It's still there in the new curriculum but it's hard to interpret that, whether every child does it or it's just offered for children to do it. That's the difference. I think now we've got to make the kids aware when they're doing their options of, if you're going for a job and there's somebody four years older than you, they will have some sort of technology qualification that you don't have.

Speaker 1: The kids I was speaking to earlier today said that. Some of them said, two of them who both had said they want to be primary school teachers. They said, "Actually it'd be better" and this says, "Everyone's going to need computing skills." It'd be better if it was required, if it was compulsory. Currently I'm not going to choose it because it'll take one of my choices away. I know it's important and probably it should be compulsory because everyone's going to need it.

Speaker 2: Some sort of training whether it's ...

Speaker 1: Someone else said, "If it was that important, it would be compulsory."

Speaker 2: Absolutely. It's a vicious circle isn't it.

Speaker 1: Maybe.

Speaker 2: I tend to do that. I tend to show them the, "Did you know" videos that are dated. It give facts and figures and the discussion they made from that, not by me, is amazing. We can actually ... I always leave it, "It's your choice. You've got to make your decisions." Look at the top ten jobs last year all have something to do with the computer. That starts opening their eyes a little bit further as to ...

Speaker 1: The jobs as well, I think this is one of the ... in some ways it's not yet a question in school, but definitely a question more broadly, what jobs aren't going to require a computer? Some knowledge. What are those skills that you'll need using a computer? If you're a plumber and you're using a computer as a device, what skills do you need to be able to use those correctly?

If you're a carpenter, probably you use some sort of digital manufacturing. What skills do you need to do that? It will be, I think, in ten years it will be difficult to find jobs that aren't enhanced by ...

Speaker 2: There certainly will be.

Speaker 1: ... in some way by digital technology. That really swoops nicely into the next question, which is a first time. To what extent do you think that the computing curriculum relates to how pupils in your school use computers in their everyday lives?

Speaker 2: Everyday lives?

Speaker 1: Yeah.

Speaker 2: In key stage three is quite a good resemblance. We're looking at what's actually being used in the household. When we're writing programme, we might send them homework to go and find objects that have a computer on board so they can actually see a computer is not just what's in front of them. They're all over the place.

When we get a little bit older into key stage four, we're looking at how technology's evolving, using [inaudible 00:56:36] technology for example to see the news. We're closely looking at the [CES 00:56:41] Conference that's been on recently as to what's been going off on that. How the web of technology is starting to become quite important in their every day lives. How it actually changes so the kids can then start to see the direct resemblance of a programmes has made us watch. A program's got this to link through that. What would I do without it? I'd cry, what not. Therefore this why it's important as to how you do it and this is how they're doing it using the part of the visual basics they're going through.

Speaker 1: What sort of homework? Do you set homework?

Speaker 2: Key stage three, when we set the homework it's either question-based; it's self-marked or computer marked set of questions, or it's to go out and find out about fine me 15 objects that have an embedded computer on board. Or find me three pieces of technology that have been recently in the news or something like that so that can actually start to see what's actually happening.

Speaker 1: I think that's one of the challenges I've encountered is the difficulty of setting homework in computing.

Speaker 2: It's not the setting of the homework, it's the marking the homework.

Speaker 1: Right.

Speaker 2: [inaudible 00:57:52] of an average week I'm probably seeing between 240 to 260 kids. How do I get the time and my future time to mark all of those in a weeks' turnaround? That's quite a big ... that's why we're looking at a computer marked one, they can give us a bit of an idea.

Speaker 1: Do you worry at all if you did set any programming assignments to do at home, whether they would struggle to have access to the ...

Speaker 2: Some do, some don't. When we set it, we've always got a computer open somewhere at lunch times and before and after school. Worst case scenario is they can go into those. Most kids, if they're interested in it will go off and download it and have a go at it at home. We always say it's never a requirement to the course. We always can make sure [crosstalk 00:58:45] .

Speaker 1: Do you find some kids do?

Speaker 2: Yeah, some will want to go off and download the thing. We always try to make sure everything is free and the problem with Visual Basic is if it's got Mac, you've got to use the old real basic ... very few people have only Mac, most have got Windows based.

Speaker 1: That's funny. That's interesting. I wonder if that's another ... I should ask kids that. I ask them ...

Speaker 2: Who's got Windows, Mac computers?

Speaker 1: I ask how many kids ... I've started asking how many computers they have in their homes and whether they have their own computer. I haven't asked if it's Mac or PC or Lennox.

Speaker 2: Absolutely. Lennox is out there as well.

Speaker 1: The thing about my own household, my wife and I both use Macs but the computer our daughter uses, she's 12, it has Lennox because it was an old, spare laptop that was creaking under its Windows.

Speaker 2: The things on there.

Speaker 1: Right, we'll do this and get a few more years out of this terrible computer.

How would you describe the degree to which the pupils find the curriculum relevant to their own lives?

Speaker 2: To their own lives? First, the kids go, "Why are we doing this? Why are we programming?" Then when you start talking about computer games or [inaudible 01:00:06] objects, somebody's had to programme a microwave to do things or someone's had to programme Uber or something like that. They start to understand the difference and importance of it. Especially when you start to talk about video games, how many people have programmed this. We start to tell them it's hundreds of people and 1000 people, depending on what it is. They start to get really, "Wow, that's a big, big ..." They ask about the development of things and then they start to see interesting sides of it.

The algorithmic side, we start to see it when we use real world examples. If we're doing basic teachings, we might start to get into instructions for [inaudible 01:00:41], you see that straight away. What we're doing now is looking at real world things.

We use things like flow alter actually. Help us that, we're getting real world objects programming a Ferris wheel or programme a robot or programme a lighthouse. Something that everybody's seen or programme a traffic light so they can actually see the sequence. They don't realise they've actually been taught Computer Science behind it whilst they're actually looking at a real world object.

Some of the programming we do as well, when we start introducing the programme, we might say, "That's where the calculators ..." Something they've used in the real world, their every day lives. They can actually then start to see just how incisive that actually works as well. It's been quite useful. Or [inaudible 01:01:27] or [inaudible 01:01:29] could be high or low again.

Speaker 1: There's a guy called M– who's on a lot of the computing stuff and he often recommends that one of the best ways to teach programming is to programme things across the curriculum. If you're doing art programmes and it draws a picture. Programme [inaudible 01:01:52].

When you're doing math, write a programme ... it embeds those concepts ...

Speaker 2: Of different things they've got.

Speaker 1: ... of whatever it is you're doing while at the same time teaching the programming in another context. I've seen a great presentation ... it's one of those things where I think there's arguments both ways. When you see something very compelling presented ...

Speaker 2: With those things.

Speaker 1: ... you can see, "Oh yeah, that makes sense." I think something like ... it'd be very interesting to say, "You can use a calculator in math if it's the one you programmed."

Speaker 2: Absolutely.

Speaker 1: Any calculator you programmed yourself, you can use in class. I'm sure they would make sure it was a good as it could be.

Speaker 2: Absolutely. It's fun giving them those ideas though. We've got abstract skills here that we can then build on.

Speaker 1: And they're useful on that you can build tools. I think one of the things that is very difficult to teach at the moment is the idea that in computing, you can build tools that are yours. You can build the tool that does the thing you want it to. I think that's quite a big shift for some kids to think about that. Tools are so often off the shelf now.

Speaker 2: Yeah. Again, I think it's just getting them used to it's computing, not the IT side of the [inaudible 01:03:16]. Again, I think as a primary school, I'm more confident in the curriculum that will disappear and what I would eventually ... [inaudible 01:03:24] what we're teaching in year seven as compares to what they should be doing in year five, year six level and what we're doing at year eight should be going down to year seven, nine and to eight and then the year nine stuff we can then start doing some more of the advanced [crosstalk 01:03:37].

Speaker 1: How many feeder schools do you have?

Speaker 2: Loads.

Speaker 1: I think that's a struggle for schools like this where there's so many feeder schools and across [Cambry 01:03:46] it's all the same. So many small primary schools that every school is different in it's delivery and resources.

Speaker 2: I've done some work with a school in Lancaster, just outside and it's a two class school. That's key stage one and one class key stage two in the other class. The teacher there's been going, "How do you do Python programming for year three, four, five and six?"

Speaker 1: All at the same time.

Speaker 2: The school has been getting her to develop something they're confident with using proper programme because the way they've been told for that primary school is to use the espresso stuff which is drag and drop Python. That's no good. It's good for the threes and fours, but for five and six it'd be more of a challenge. It's been looking at ...

Speaker 1: In some ways that key stage one gap is even more difficult because ...

Speaker 2: Absolutely.

Speaker 1: ... I've done a little bit of work on this as well because I think the developmental changes from year one to year three are huge. Starting with using a keyboard and [inaudible 01:04:51]. It's very difficult. You know your key stage three kids are pretty confident in the scratch and probably could do some text based stuff. But your year ones can't type very well.

Speaker 2: That's why it's more of the drag and drop. Five and six is they need an actual challenge. What I found quite useful is over the course of a term they've had one lesson a week. They've gone from no programming whatsoever to pretty confident in the basics. The basics in Python would have been print, import, FC lifts and four loops. Leave it at that. It's amazing they could go off and write Christmas quiz that had all those [inaudible 01:05:35] in there.

The challenge was to get the teacher happy. Tell the teacher what to look out for. I know a lot of people say, "You should be bringing that down to a key stage two." I think in some ways we should be challenging the upper levels.

Speaker 1: Absolutely. I think the other thing that the teacher's aren't ready for but the kids are [crosstalk 01:05:55] is doing things like [ardwinos 01:05:57] and physical computing. In some ways I think the microbit is way below year sevens.

Speaker 2: It shouldn't be.

Speaker 1: It sounds like the microbit should be introduced at year three. And then by the time they get to year seven they should be on a Python.

Speaker 2: Yeah, things like that.

Speaker 1: An [ardwino 01:06:15], something far more ... I don't want to say complicated but more complicated and [crosstalk 01:06:19].

Speaker 2: I guess the problem with doing in an hour a week with the [inaudible 01:06:25] getting all this stuff set up the proper way, re-cabled. Unless you've got indicated space it's a nightmare.

Speaker 1: [Pi's 01:06:35] are funny. I don't know if they're as good as they think they are.

Speaker 2: [Cano's 01:06:39]. Have you seen the [Cano's 01:06:40].

Speaker 1: I have seen the [Cano's 01:06:41].

Speaker 2: They've been quite useful. I've worked with a child here, an actual curricular using a [Cano 01:06:48].

Speaker 1: The other one I'm interested in, I might even ... there's something called the Pi Top, which is like a laptop case.

Speaker 2: Okay.

Speaker 1: I'm really tempted to get one to play with it but also as a thing for schools to look at because I think there are some real challenges for using Pi's in schools. I also think ... I have this theory that they breed in closets at schools. Put 40 of them away and I'll end up with 60. Raspberry Pi; every school I think has a cupboard of raspberry Pi's that they don't know what to do with yet.

Speaker 2: We've got them. We use them with ESNs and 11s and 12s. We had one for projects. It's funny how we can actually put that into the proper curriculum of the mainstream.

Speaker 1: The other problem I think with microbits to a certain extent and things like [ardwino's 01:07:50] far more-so is that they don't fit in very well. They're a bit of design technology and they're a bit of [crosstalk 01:07:59].

Speaker 2: They are, so who takes the responsibility for them?

Speaker 1: If you're actually doing red boarding with electronics but you also have to programme it, who is doing it?

Speaker 2: They make it the DT and they come to us to do the programming.

Speaker 1: Then you have to move these things around with it.

Speaker 2: Yeah. It's one of those things, who takes control of it?

Speaker 1: I think computing demonstrates how some of the disciplinary distinctions in the curriculum are creaking a bit. Because computing is new, it shows that some of those don't work as well as they should.

Speaker 2: Absolutely.

Speaker 1: Last two questions I promise. Do you think that learning computing has effected people's decisions and choices about their future?

Speaker 2: I think it has. I think much more now going to the programming side, the kids have gone to a lot more thinking of, "I can do something with the computers to see where I want to go." It's not, "I want to do this, I want to do that." It's, "I want to develop this, I want to develop that."

If more kids are wanting to go down or are going off to do Computer Science for a degree, which has been quite positive ...

Speaker 1: Has the type of kids ... has there been a change in the type of kids that are interested in Computer Science as a degree do you think?

Speaker 2: I think what we're getting now are kids that actually have the passion for it. Not they're good on the computer so they should take it. I think that's what used to happen. You're on the computer a lot, you should do it. You should do computing.

We've gone away from that now. We've got very much into a, "I want to actually learn about it. I really want to see that passion." Again, they've got passionate teachers, they know that trust there as well. They know exactly what, who and where it goes through.

Speaker 1: Also understanding that it's far better to have kids that are really interested and making sure they have that introduction into Computer Science earlier on and see how that feature is possible.

Speaker 2: I think it's thinking a lot more as to where they want to go ...

Speaker 1: Do you think it's restricted their choices in any way?

Speaker 2: Restrict? I think the way the curriculum is for most schools, it's a choice option. IN the restriction, I'd say more and more kids are now saying, "I need to do some sort of qualification that's either IT based or the computing base. That's my option gone." It could restrict them into the other parts of it. Equally, it's one of those competencies that we can then show we've got. It's that type of two sources that might lose another option from something a bit more creative or thought that they've got a life skill or a competency that they're going to have to show for a job interview. Much more than that. It's a round about I guess.

Speaker 1: You think having the computing ... overall having the computing [crosstalk 01:11:21] gives them more options?

Speaker 2: What we're seeing a lot less of are people doing both IT and the Computer Science qualification.

Speaker 1: Interesting.

Speaker 2: We're getting much more of the quite technical, quite hands on, "I want to, I'm good at math so I want to do Computer Science." "I'm more creative, I want to develop product right." You're going down more the ICT area. We can see that more.

You still get the odd couple .. a couple of years ago it would have been people wanting to do both and that is going quite a lot. For some that's good because it is a different subscale but in some ways it's bad as well because you're only getting one specific subscale.

Speaker 1: This kind of skips back, what do you think is the biggest if anything, the biggest loss either for you as a teacher or for the kids from ICT? The biggest thing that they don't get anymore and you think they should be, if anything?

Speaker 2: The biggest loss from the ICT skill that we taught, probably spreadsheet skills I think they've lost the most. So many of the subjects depend on that. That and presentation skills. Most people, job interviews you have to do, you have to develop some sort of presentation. That group work ... we still do group work with programming but it's a different way of being able to present and do it to a good standard.

Speaker 1: Those were all within ICT?

Speaker 2: Yeah, they were the ICT strands. We don't do lessons on how to [inaudible 01:12:50] anymore or how to do spreadsheets. We still use spreadsheets, but very much as a tool, not as a developer.

Speaker 1: Less kind of discrete teaching. This is a spreadsheet and this is how you use it. This is how you can link ...

Speaker 2: When the kids go to do a science experiment, make a graph, "How do I do that on the computer?" Same as geography. "Make me a pie chart" and [crosstalk 01:13:11].

Speaker 1: Do you think some ... I imagine some kids can figure it out?

Speaker 2: Yeah, some can but it's amazed the other subject teachers, "Why can't they do that anymore?" Then they go, "Oh, how do I teach that quickly?"

Speaker 1: The science teachers have gotten very reliant on you guys teaching spreadsheets?

Speaker 2: Absolutely and now we've stopped that.

Speaker 1: Now they're saying, "Oh, I didn't know I had to teach them how to use a spreadsheet."

Speaker 2: That's the next thing, they then go applying that to their lessons.

Speaker 1: We're doing graphs, so we're going to have to spend less on how to make a graph in Excel.

Last question and this is probably one of the hardest ones. If I ask the pupils in five years, when they're 19 or 20, about learning computing at key stage three/four, what aspects do you think will have had the most impact and what do you hope or think they'll say?

Speaker 2: I would hope with where we are now that they'd be able to understand why we're doing the curriculum we did five, six years ago. I think if they've done some programming they might have come back to programme, especially if they're doing science subjects. Even if you're doing math, you probably still have to write a programme to do things. It's between that transferrable skill that is quite useful for the subjects.

The actual insight into computer, that's more of an awareness and the passion I guess. The literacy side, the safety side, hopefully they'd be able to understand how and why they're kept safe or, "I got that spam email and I remember that lesson we did and I knew it was a spam and I knew how to ... it was a fishy email", what not, "And I knew how to get rid of it."

Speaker 1: It's interesting, a lot of kids say ... I asked them ... I'll show you the questions I asked them, but one of the things I asked them was whether they're part of online groups at all? Often somebody, when I do these interviews will say, "You speak to people you don't know but as long as you're [inaudible 01:15:12], it's fine. They do have this sense of, "I can speak to whoever I want but I don't give away my name, my personal details." They all seem to have that message, which is really interesting and more or less be making friends across the world, which is really interesting to see as well.

Speaker 2: They understand it's a valuable tool, which also is a risky tool that they've got to be aware of.

Speaker 1: They have that sense of where the risks are in an interesting way. I think it's instinctual rather than what they're taught necessarily. They see ... different kids have different senses but they do have that sense of how to use this tool.

Great. That literally is the last question.

Speaker 2: No problem.

Speaker 1: How will they use it in the future. I think ti's always an interesting one.

Speaker 2: Yeah.

Speaker 1: What would you like them to remember?

Speaker 2: Yeah.

Speaker 1: Again, it's always to ask the kids. I ask them that too and they say different things. Sometimes they say nothing. Today they said, "We spent a long time on programming. I think I'll remember the programming." Okay. That's it.

Speaker 2: Excellent.

Speaker 1: Do you have any questions for me? Is there anything I haven't asked you?

Speaker 2: You've covered most things to be honest, which is good. You've covered some of the questions ...

Speaker 1: Don't feel like that's it. If you think of anything, feel free to email me.

Speaker 2: Just let you know? Okay.

Speaker 1: I'll probably ... at some point I may need to get a permission slip that you've signed as long as you know you've verbally given permission, that's fine.

Speaker 2: That's fine.

Speaker 1: I really appreciate it.

Speaker 2: No problem.

Speaker 1: I will, at some point, transcribe this.

Speaker 2: You've got all that to look forward to.