Raspberry Pi

The impact of the coronavirus pandemic on the computing classroom

By Dr Sue Sentance, Chief Learning Officer
January 2021

The number of coronavirus cases has surged in the last few weeks. Last-minute announcements by the English government about when and how schools should return after the holiday break have led to more rapidly reorganised work schemes and new plans for remotely delivered lessons. Other countries are affected in exactly the same way. Since the beginning of the pandemic last March, schools and teachers all over the world have had to respond and react to sudden switches between remote and face-to-face teaching. Even face-to-face teaching is very different, with students placed into 'bubbles' and social distancing measures in place.

So how does this affect the actual learning and teaching that takes place? How is teaching a practical topic like computer programming affected? I recently interviewed teachers about their experience, as part of my continuing research into the PRIMM approach to teaching programming. I interviewed 20 teachers in total: 3 in primary and 17 in secondary schools, and all 20 specialist computing teachers experienced in the teaching of programming. From these interviews I've gathered some insights about the impact the pandemic has had on the teaching of programming.

Teaching programming during lockdown

The first set of observations relate to how teachers managed while all schools were closed. Among the small sample of 20 teachers, 14 had tried to teach programming online during school closures. Others had not attempted it at all, choosing to focus on non-programming related aspects of computing.

For some teachers, attempts to teach programming remotely had "crashed and burned": teaching this skill in any meaningful way had been very difficult to achieve, primarily because of a lack of appropriate technology. Teachers reported that learners had been trying to use a range of different devices, from smartphones and tablets to low-spec laptops, and they couldn't make any progress in lessons or with set work. This issue has been widely reported in the media, and so far it persists despite attempts by governments and a range of other organisations (including the Raspberry Pi Foundation) to provide adequate technology in the home. The most significant impact of lack of technology or internet access at home is experienced by those in deprived areas, as reported by the Sutton Trust. One of the teachers I interviewed reported:

> “We started to use [the online programming environment] repl.it, but what we found, with where the school is, the availability of devices that can handle doing two things at once, or even doing something using repl.it at all on a mobile device — it just doesn’t work. So we ended up just doing theory online that can be done on a bit of paper. “ – secondary teacher

Environments such as repl.it, Trinket, and .netfiddle were all mentioned by the teachers who had been successful in teaching some programming during school closures. These environments allowed teachers to share code with students to work on online:

> “I was using Trinket. The first part [of code] at the top might have been an example, so they would then modify that [...] Then towards the bottom of the page, they’d have an opportunity to make their own and to do their own creation of code.” – secondary teacher

Most of the teachers found that working asynchronously was more successful: students had time to work on their code and then came together with the teacher to share their learning:

> "Some of them I think benefited from the fact that there wasn’t that one-hour pressure in the classroom to do something. They can have more time to tinker and to have a go and explore and look at other peoples’ code and read that. I think it’s really important for them to read other people’s code." – secondary teacher

Most teachers reported that they were not permitted or encouraged to use live, synchronous video to talk to or model programming solutions for their students. As a result, some teachers recorded their own videos to set work for students. Although a significant amount of work, this seemed to be successful:

> “I gave them YouTube videos that I’d recorded, and I gave them links to Scratch code that I’d created around the game, and then they had to go and modify my code or remix it and modify it and put comment boxes on it, and then send me the link back.” – primary teacher

By necessity, students learned to communicate differently with their teachers, which seemed to precipitate a shift in the teacher-student relationship:

> "If they really got stuck, they would just message me. And I’d ring them on [Microsoft] Teams and have a one-to-one and take over their screen so that I could see what they were doing." – secondary teacher

Where programming online didn’t work well, some teachers reported that they focused on written exercises to support their programming teaching and formatively assess progress. Teachers reported the use of exercise books and worksheets at home. Alternatively, a number of teachers described using environments such as Microsoft Teams or Google Classroom to get students to take screenshots of their code, annotate it, and share with their teacher.

And back to school…

Teachers reported that, when back in school, they adapted the way they taught programming. Schools are now divided into bubbles, and how timetables and rooming are organised has had to change. Many teachers described using a process of trial and error as the term went on: they would try one teaching approach and then another, at the same time being responsive to any changes in the safety guidelines that affected the way they were allowed to teach. No wonder teachers are worn out!

One of the most impactful changes to the in-classroom teaching of programming seems to be social distancing. In most schools, teachers were not able to be close to individual students in the classroom to have quiet conversations. The age-old practice that programming teachers use of looking over shoulders, to spot errors and support students who may be stuck and frustrated, was no longer possible. One teacher reported:

> “We are restricted in our movements. I can have a conversation, but it tends to be quite loud. Everybody has to sit quite quietly if somebody needs help. [...] Otherwise you're shouting across the classroom because you can't move to see them.” – secondary teacher

One teacher reported that she was not allowed to let students talk to each other at all, so where she might previously have asked students to work through a program together, she instead encouraged them to use the rubber duck technique to debug their programs by themselves.

Supporting students via classroom monitoring software

Several teachers reported using software on a desktop computer at the front of the class to see all their students’ screens at once and monitor their programming progress. One teacher described being able to see one screen “go red” (with a Python syntax error) and knowing instantly which student needed help:

> “Rather than saying 'Where are you stuck?' and they tell you and try to explain to you, I get it from [the software] and I can see how much they've understood.” – secondary teacher

This is a significant change in pedagogy, as teachers previously might have encouraged students to work through a programming problem to try to explain verbally the cause and location of the error. This dialogue is important in a number of ways, from using correct programming terms, to be able to orally trace through a program. The restrictions may mean that teachers need to just tell the student where the error is, which is much less effective in the longer term.

Where students need to be socially distant, some teachers reported that they used breakout rooms within video conferencing environments rather than physically close groups, so that students could work on tasks and also confer. And despite many difficulties, some teachers said that their students had had to become more independent, particularly the older ones.

> “The way we do things since September has changed, definitely. I would say they're more independent, which is fantastic. Especially the Year 10s seem to be working harder because they know I can see what they're doing all the time as well.” – secondary teacher

Teaching while self-isolating

Teachers get ill. With increasing numbers of infections, many teachers have caught either coronavirus or have to self-isolate while not being ill themselves. Some teachers reported some success teaching their classes when they were self-isolating and the students were in class. This was really successful for one teacher, surprisingly so:

> “I had to isolate for two weeks, and I taught from home into school. And now the children with desktop computers and good internet, I would say we operated at 95% [...] It was really successful. And we used the video conferencing chat as a 'back channel' to ask for help. So I wouldn’t let the children come on that for anything else apart from asking me for hints.” – primary teacher

This teacher explained that a surprising benefit was that the learners could see explanations given to classmates, which they wouldn’t normally be privy to.

Nevertheless, another teacher who taught in this fashion on multiple occasions said that her students were pleased when she returned in person:

> “One of the students turned around and said 'Miss, I'm so pleased you're back. This programming lark is too difficult when you're not in the room.' Even though I can't physically go near her, it’s the reassurance of having me there and the instant questioning, I think.” – secondary teacher

Teaching programming differently

We may all have many more months of these different ways of teaching ahead of us. Summarising the points raised by the teachers I interviewed, we can see that:

- There is less opportunity for pair work, an approach that is very important in learning programming. Working together to share code is invaluable, and we should continue to find effective ways of doing this online while pair programming is not practicable.
- Less classroom talk about programming, both with the teacher and between students, reduces the opportunity to verbalise the way code works, and to practise the vocabulary to help us talk about programs.
- The disruption caused by intermittent school closures means it is difficult for students to maintain the continual practice they need to learn programming concepts.

Each time schools close and students are forced to learn at home, the accessibility of technology and drop-off of participation becomes an issue again; these issues will impact each and every subject in school.

However, the sheer determination of teachers to provide their students with a quality education experience while they are at home means that some benefits are emerging through the effective use of technology:

- Students are sharing and looking at each other's code, which builds up peer support, something we know to be effective for learning
- Teachers are taking a facilitator role more often, with students needing to work more independently
- Older students are using more 'adult' ways of working with teachers through technology (e.g. Microsoft Teams etc.) and thus become increasingly independent

Adapting in this way can be difficult for students who need more support in the classroom, and it may favour young people who have strong support networks and are already resilient, thus widening the attainment divide.

As we move ever more into remotely delivered education, we need to understand better what classroom and online environments support effective teaching and learning for all students, what types of devices can be used in the teaching of programming, and how to break down material into small enough chunks so that we can maintain students' confidence as well as their progress.