COLUMN

SUPPORTING WELL-BEING IN THE INCLUSIVE CLASSROOM

Catherine Elliott introduces approaches and techniques to increase computing confidence and self-esteem for students with SEND

Young people with learning difficulties are four and a half times more likely to have a mental health problem than children without a learning disability, according to a study by Emerson and Hatton in 2007 (helloworld.cc/iss17sendstudy). Low self-esteem and anxiety are common among pupils with dyslexia and autism, and social, emotional, and mental health difficulties are identified as a specific area of need in the UK SEND Code of Practice (helloworld.cc/sendcop).

As with any subject, computing has its own frustrations and risks for learners, from an ongoing struggle with the correct syntax in a program, to dealing with setbacks when technology doesn't work. Here are a few ways you can support your students by developing their confidence and reducing the risk of failure in your lessons.

Reducing anxiety

Structure and routine can really help young people with special educational needs and disabilities (SEND) know what to expect in a lesson, and will reduce anxiety, particularly among autistic learners. It may be useful to develop routines for logging on, for handing out equipment and worksheets, and for transition points in the classroom, or to create a familiar structure to lessons. Try to provide enough time for pupils to record any homework and pack away at the end of a lesson.

Reflecting on virtual lessons during lockdown, a number of teachers have talked about students who wouldn't normally answer questions in in-person lessons suddenly offering answers much more frequently via the chat box. These students felt more comfortable in this situation, where the focus wasn't on an individual. There are several strategies that you can use to support and encourage learners with SEND to answer questions in the classroom:

- Provide more time for learners to answer.
- Provide multiple-choice answers to choose from.
- Use technology to allow students to answer anonymously (for example, Plickers or Mentimeter).
- Develop peer instruction to allow for group discussion so that the focus is not on the individual. It is important for the teacher to emphasise that it is the discussion that is key, not necessarily achieving the correct answer (see **helloworld.cc/peerinstruction** for more).

Enabling success

One way to guarantee a certain degree of success for students is by providing templates and working programs for them to modify. For example, you could provide a poster template for learners to add their own text and change the colours, before gradually removing the scaffolding as they become more confident. Similarly, in issue 12 of Hello World, I wrote about how the PRIMM model allows learners to access programming projects in different ways — from predicting what happens, to investigating and talking about a piece of code with a peer, to modifying existing code. This is a great way to involve all students in the learning at a suitable level while reducing the risk of failure.

Quick wins from physical computing devices can also be great for engaging learners, as they see the results of a short piece of code translated into a sensory output such as sound or movement. It is

Catherine is the SEND lead for the Sheffield eLearning Service (**sheffieldclc.net**), and she works on ways to make the subject accessible to all learners. She is a member of the CAS Include working group, and leads the SEND Virtual and the Sheffield and South Yorkshire Secondary CAS Communities (**@catherinelliott**).

important to ensure a level of guaranteed success before developing more complex programs — this could be programming a Bee-Bot to reach a destination using a given program, modifying a program on a micro:bit to display their own text, or controlling a Sphero using an app.

Developing confidence

One of the best things you can do is provide students with the tools and strategies to help themselves when they get stuck. For example, you can provide lots of practice in debugging different projects, showing the most common errors and how to correct them. It is important for learners to do this in a project they are given, as it is much less intimidating and easier to find errors in a project that is not your own. Celebrate debugging, and therefore failure, as an integral part of programming. Ryan Hayes, a teacher at a special school in Wales, makes deliberate errors in the programs he gives to students — they love pointing out and correcting his errors and are more engaged as a result.

Parson's Problems are another great tool for developing confidence while reducing the anxiety of being faced with a blank page. These problems include all the code that is required to make a functioning program, and students have to put it in the correct order to make it work — so they don't need to type in any text, worry about indentation or spelling, or choose the correct blocks from an overwhelming selection. Once again, this provides plenty of scope for differentiation, so that all students can be working on these types of problems at a suitable level. To stretch learners, you can add in distractor code, and to support learners further, you could have some of the code already completed (see **helloworld.cc/parsonsproblems** or page 70 for more).

All these approaches will help to reduce anxiety in the computing classroom, and help learners develop some resilience in dealing with problems. This is incredibly valuable for students with SEND, to help them to engage with the subject and start to develop their confidence. (HW)

IDEAS FOR INCLUSIVE HEALTH AND WELL-BEING COMPUTING PROJECTS

- Create an emotion badge using the micro:bit or CodeBug - how do you feel today? Students can modify a working program if required.
- Design an app or website in groups with a health and well-being focus - this could be a paper prototype without the coding input at primary, or using something like Google Sites. Group work can support the strengths of different learners if managed carefully.
- Learn about selection using 'if-then' commands in PE - for example, if I blow the whistle once, then do ten star jumps.
- Create a Scratch bleep test or reaction game. Have a look at the project at **helloworld.cc/bleeptest**, which you can use for scaffolding.