(Hello World)

THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS

Issue 17 | Oct 2021 | helloworld.cc

INTERVIEW: MAKER'S ASYLUM

The makers innovating to fight coronavirus in India

DIGITAL DETOX

Running a primary digital detox day

HINTERLAND EXPLORATIONS

Reading around the subject to improve teaching

HEALTH & WELL-BEING

TECHNOLOGY, TIPS, AND TECHNIQUES TO HELP YOU LOOK AFTER YOURSELF AND OTHERS

PLUS

ASTRO PI • SPORTY CODING PROJECTS • BEHAVIOUR MANAGEMENT • PARSON'S PROBLEMS
DISCORD • MINECRAFT PLAYGROUND • PROGRAMMING ANXIETY • CULTURALLY RELEVANT TEACHING
POLLUTION MONITOR PROJECT • PLANNING FOR TEACHER ABSENCE • COMPUTERS AND EYE HEALTH

Raspberry Pi

Learn with the Raspberry Pi Foundation

Free for everyone anywhere in the world

Teaching resources

Discover training, resources, and guidance to help you teach computing with confidence.

teachcomputing.org

Project library

Browse our 200+ online project guides that include step-by-step instructions for learners.

projects.raspberrypi.org

Digital making at home

Check out our code-along videos and take part in Astro Pi Mission Zero from home.

raspberrypi.org/learn

Support for parents

Watch our support tutorials and access engaging resources for your child.

raspberrypi.org/learn

Raspberry Pi Foundation, UK registered charity 1129409

SUBSCRIBE IN PRINT FOR FREE TURN TO PAGE 38

HELLO, WORLD!

Not only is health and well-being an engaging context to bring computing to life for your students, it's also never been more relevant as we emerge from periods of isolation, and pay more attention to personal and public health. The last 18 months has been a time of great uncertainty and upheaval for educators and students; making sure that we don't forget about our emotional and social health, and how we look out for ourselves and others, is vital.

This issue, we hear inspiring stories from a makerspace in India about their efforts to fight the coronavirus, and from researchers who are developing solutions to healthcare issues using computing skills and knowledge. We also hear from Cat Lamin about her efforts to form a virtual space for teachers to meet and support one another, and from Georgie Ford about the importance of the language of emotional recovery in educational settings. We hope that this collection of articles focused on health and well-being helps you to connect the teaching of computer science to its applicability outside the classroom walls, and to support conversations and initiatives around well-being.

If you haven't had a chance to check it out yet, we recently launched the first-ever special edition of Hello World, *The Big Book of Computing Pedagogy*. You can find the free PDF on our website, and we hope it keeps you busy! As always, get in touch with your thoughts about Hello World at **contact@helloworld.cc** and visit **helloworld.cc/writeforus** if you have an article idea — we'd love to hear from you!

Gemma Coleman
Editor

FEATURED THIS ISSUE

MARK TETTEH AZIETAKU

Mark is a senior information technology teacher at New Nation School in Ghana. He shares his research into his pupils' eye health during the pandemic on **page 33**.

SETHI DE CLERCQ

Sethi is the subject leader for STEM and computer science at a British International School in Thailand. On **page 46** he explores the uses and benefits of Discord as a teaching and learning tool.

TONYA COATS

On **page 87**, Tonya, a primary school teacher at Rustic Lane Elementary in California, USA, shares how she creates engaging and collaborative spaces in her classroom in our My Classroom feature.

(HW)

EDITORIAL

Editor
Gemma Coleman

Subeditors
Louise Richmond and Amy Rutter

Subscriptions
Joshua Crossman

Social Media
Neena Patel

DESIGN

criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Ty Logan, Sam Ribbits, Lucy Cowan

Photography
The Raspberry Pi Foundation, Adobe Stock

Graphics
Rob Jervis

Cover
© Emanuel Wiemans, Folio Art

CONTRIBUTORS

James Abela, Ali Alzubaidy, Mac Bowley, Polly Card, Katharine Childs, Beverly Clarke, Tonya Coats, Gemma Coleman, Sethi De Clercq, Diane Dowling, Anna Doyle, Catherine Elliott, Georgie Ford, Rebecca Franks, Ben Garside, Henna Gorsia, Phil Hackett, Tom Hadfield, Ben Hall, Alan Harrison, Eoin Houston, Michael Jones, Stephen Jull, Mustafa Khalid, Diana Kirby, Laura Kirsop, Cat Lamin, Kat Leadbetter, Hayley Leonard, Richmond Mathewson, Siobhán Morgan, Alan O'Donohoe, Shahina Pardhan, John Parkin, Yolanda Payne, Oliver Quinlan, Neil Rickus, Jared Rigby, Chris Roffey, Sue Sentance, Mark Tetteh Azietaku, Gayle Van Tol, Dru Watts

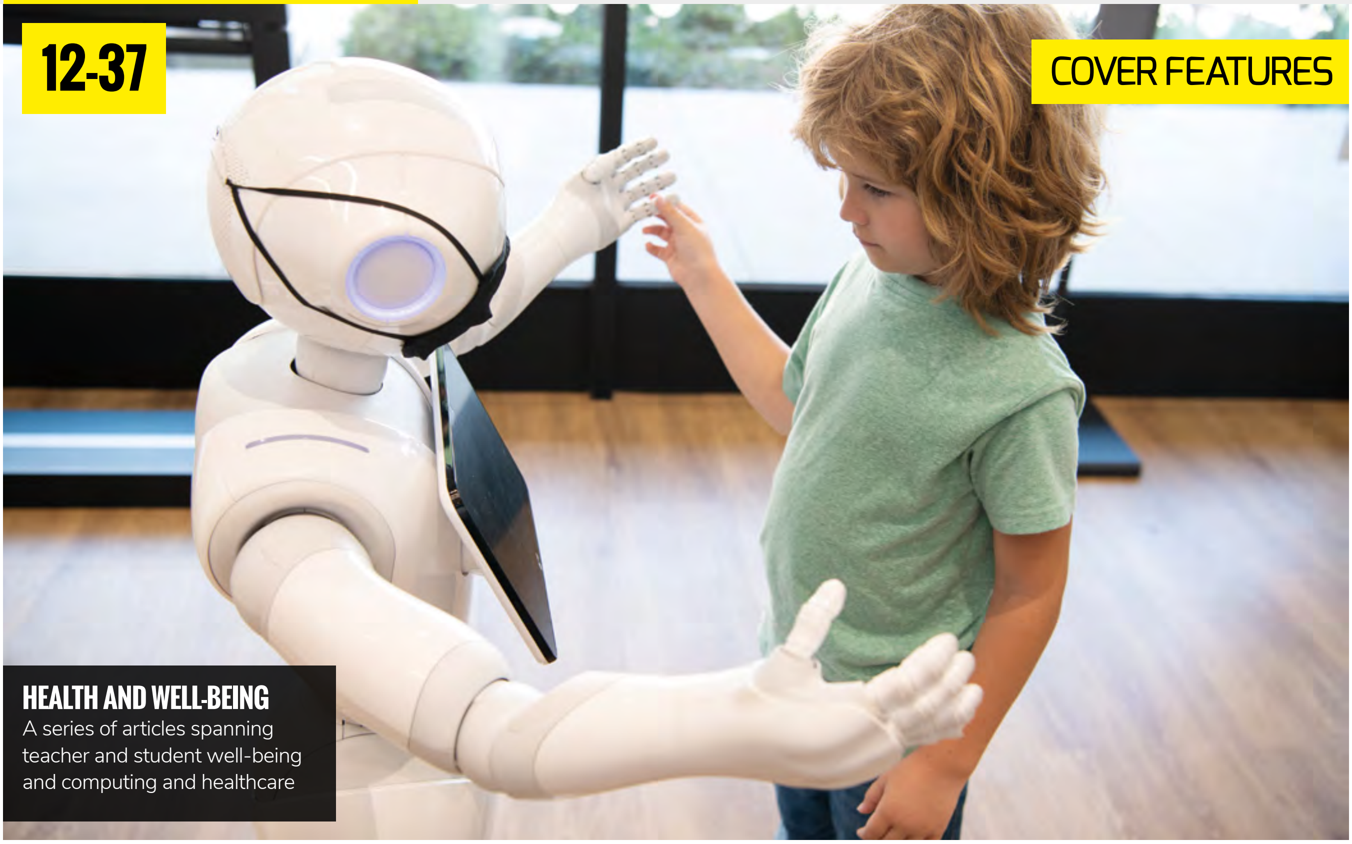
Hello World is a joint collaboration:

Raspberry Pi
bcs The Chartered Institute for IT Academy of Computing
CAS

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

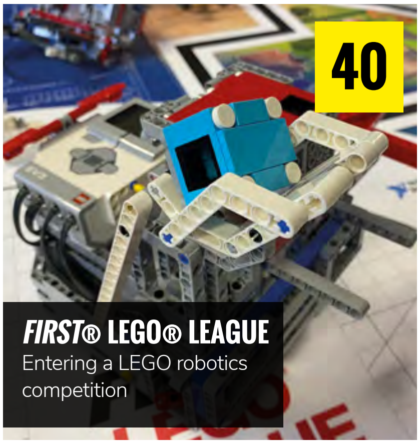
Hello World is published by the Raspberry Pi Foundation, 37 Hills Road, Cambridge, CB2 1NT. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to skills, products, or services referred to in the magazine.

(HW) CONTENTS



© Olena/stock.adobe.com

NEWS, FEATURES, AND OPINION

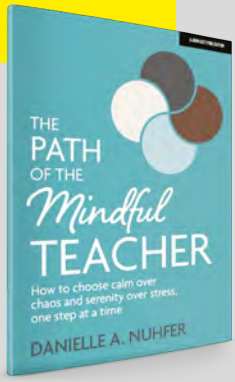


LEARNING

ACTIVITIES & LESSON PLANS

CONVERSATION

REVIEWS


THE PATH OF THE Mindful TEACHER
How to choose calm over chaos and serenity over stress, one step at a time
DANIELLE A. NUHFER

ESA astronaut Thomas Pesquet with the Astro Pi computers on board the ISS

© European Space Agency

WE HAVE LIFT-OFF!

The Raspberry Pi Foundation and ESA Education are sending upgraded Astro Pi computers to the International Space Station to allow young people to get their code into space

Tom Hadfield

Since 2015, the European Astro Pi Challenge has given over 54,000 young people in 26 countries the opportunity to have computer programs they've written run on board the International Space Station (ISS).

Astro Pi, a collaboration between the Raspberry Pi Foundation and the European Space Agency (ESA), began during UK astronaut Tim Peake's Principia mission on board the ISS (**helloworld.cc/principia**). There are currently two Raspberry Pi computers on board the ISS in the Columbus module (the science lab on the ISS). They're in special cases, loaded with sensors and a camera to capture data about being in space. After two years of secret development, two new upgraded Astro Pi computers are ready to be transported to the ISS on a rocket that will launch from Kennedy Space Centre in Florida in December 2021.

They contain the newest model of the Raspberry Pi computer, an upgraded high-quality camera, and an augmented sensor board. This equipment will enable participants to do a wider range of experiments, including experiments using elements of artificial intelligence (advanced machine learning) and higher-resolution photography than ever before. They will be put into service in early 2022 and used to run young people's Mission Space Lab and Mission Zero programs, as part of the European Astro Pi Challenge 2021/22, which launched last month.

Mission Space Lab

Mission Space Lab is designed for young people under 19 with some experience of computer programming and takes place in four phases over the course of eight months. It is particularly popular with secondary school students, mentored by their science or ICT teachers, but one of last year's winning teams featured participants as young as eight — proving that you are never too young to learn a fundamental STEM skill like coding.

In this mission, teams of two to six young people design and program a scientific experiment investigating either life on Earth or life in space, that captures data from the Astro Pi's environmental sensors and/or photographs of the Earth using the Astro Pi's near-infrared camera.

The qualifying experiments are deployed on the ISS for three hours each, with teams later receiving the experiment data they have captured, to analyse and report on their findings. These reports are judged by an expert panel, and the winners receive certificates showing where the ISS was when their programs were run and a range of special prizes, including a Q&A session with an ESA astronaut.

Examples of past winning experiments include studies of the Earth's gravitational field, photographic analyses of the effects of climate change on the Earth's vegetation coverage, and investigations into the potential for viruses and other diseases to spread within the controlled environment of the ISS.

Mission Zero

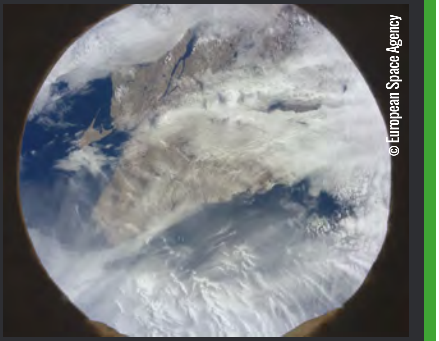
Mission Zero is designed for beginners to computer programming under 19 years of age and can be completed online in around an hour with no prior coding experience or specific hardware. This makes it a popular activity for schools, after-school clubs, and children participating from home, as it only requires a time commitment of a single lesson or session, so can easily fold into their existing syllabus or activity schedule.

The mission is to write a simple Python program to take a humidity reading on board the ISS using one of the Astro Pi computers. The reading is then displayed to the ISS astronauts for 30 seconds, alongside a personalised message.

PREVIOUS MISSION SPACE LAB WINNING ENTRIES

Some of the winning 2020/21 entries:

- Magtrix from The Leys School in the UK analysed whether geographical features of the Earth such as mountains affect the planet's magnetic field
- Atlantes from Niubit Coding Club in Spain used a sonification process to convert data captured by the Astro Pi's sensors into music, inspired by Commander Chris Hadfield's performance of *Space Oddity* on the ISS in 2013
- Mateii from Saint Sava National College in Romania investigated the potential growth of Aspergillus and Penicillium mould on the ISS in comparison to on Earth
- SpaceRad from Centrum Nauki Keplera Planetarium Wenus in Poland investigated albedo (the proportion of light or radiation that is reflected by a surface) on Earth, to evaluate the efficacy of using solar farms to combat climate change
- Spacepi2 from Zanneio High School in Greece investigated urbanisation on Earth by comparing photos captured by the Astro Pi's camera with historical data, using an automated photo classification program they created and NDVI (normalised difference vegetation index) analysis



© European Space Agency

Newfoundland and Labrador in Canada, as captured by the SpaceRad team

> EVEN SOMEONE NEW TO PROGRAMMING HAS A CHANCE TO BE A SPACE EXPLORER

All submissions that follow the set of easy step-by-step instructions and entry guidelines are guaranteed to run on the ISS, meaning that even though a young person may be new to text-based coding, or computer programming in general, they still have the chance to become space explorers.

The upgraded Astro Pi computer

In the 2020/21 challenge, over 14,000 young people aged 14 and under took part in Mission Zero, either individually or as part of a team, and 44.2 percent of them were female — a rarity for computer programming initiatives of this kind.

What's in a name?

To celebrate the launch of the new Astro Pi computers, participants in this year's Mission Zero also have the rare opportunity to help name the new Astro Pi computers and be part of ISS history.

The Astro Pis will be named after two inspirational European scientists. Participants can suggest their own names or pick from a shortlist of suggestions that includes famous names such as Marie Curie and Alan Turing, and less well-known pioneers such as Caroline Herschel, John Edmonstone and Tycho Brahe. A full list of names and more information on how to take part in the naming competition is available at **rpf.io/mission-zero**.

The winning names will be announced by ESA astronaut Samantha Cristoforetti from the ISS in May 2022.

Get involved today!

The European Astro Pi Challenge is free to participate in. For both missions, each participant has to be under 19 years of age, located in an ESA member state (see the full list at **helloworld.cc/astroeligibility**), or in Slovenia, Lithuania, Latvia, Canada, or Malta, and at least one of the following:

- Enrolled full time in a primary or secondary school
- Homeschooled (certified by the national ministry of education or delegated authority)
- A member of a club or after-school group (such as Code Club, CoderDojo, or Scouts)

More information about taking part in the European Astro Pi Challenge 2021/22, including guidance for Mission Space Lab experiment ideas, is available at the Astro Pi website, **astro-pi.org**. (HW)

CODE CLUB WORLD

Code Club World is a new way for young people aged 9 to 13 to learn to code at home

Laura Kirsop & Stephen Jull

Since February this year, we have been part of the team at the Raspberry Pi Foundation creating Code Club World (CCW) — a new free website where young people can learn to make stuff with code.

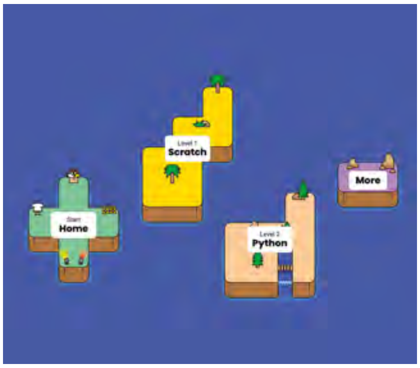
In CCW, learners start their journey by creating a robot avatar with code blocks. With this avatar, learners then explore a map, moving between different islands with cool new challenges. The first island is Home, where they make a character dance, design a T-shirt for their avatar, and compose some music. Along the way, learners earn badges for their progress and can share their work with family, friends, and to a community gallery moderated by the CCW team.

Once they've gained some confidence, learners can use the map to explore other islands and learn to program animations, games, and other exciting applications in Scratch and Python. And we're not stopping there: we're updating the CCW environment every single day in response to user feedback.

> WE WANT EVERYONE TO BE ABLE TO LEARN IN A RICH, ENGAGING ONLINE ENVIRONMENT

Learning at home

When we were speaking to parents and children about learning at home during the pandemic, it became clear to us that they needed educational products that children could master independently, that were as fun and social as the computer games and other apps they love.



Learners explore a map with plenty of coding challenges along the way

What's more, we think this is particularly important for children who are unable to attend coding clubs in person. We want everyone to be able to learn in an online environment that is just as rich and engaging as a face-to-face extracurricular learning experience.

For this reason, we are making sure that CCW learning activities are aligned to our new format for educational projects, developing learner independence alongside their skills and knowledge (**helloworld.cc/projects**). When children are in the CCW environment, they are learning to use computing and digital making to build ideas and address challenges that matter to them.

By providing a structured pathway through the resources, a reward system to engage and motivate learners, and a broad range of projects covering different topics, our objective is to support learners at every stage, while keeping the learning meaningful, fun, and flexible.

We've also been designing these activities to be mobile-friendly, so if a young person's only device is a phone, then they can still make lots of cool things to be proud of. If kids want to make tons of T-shirt designs to stockpile in their personal avatar's wardrobe, or transcribe their favourite music in code to share with friends, CCW is the place to be.

Created with our community

During the time we have been developing CCW, we have been working with a community of over a thousand parents, educators, and children who have been giving us valuable input to shape the direction of the product. We've had some fantastic feedback from them so far:

"I've not coded before, but found this really fun! … I LOVED making the dance. It was so much fun and made me laugh!" (Learner, aged 11)

"I love the concept of having islands to explore in making the journey through learning coding, it is fabulous and eye-catching." (Parent)

The product is still in beta — this means we'd love you to share it with young people in your school and community. Together, we can ensure every child has an equal opportunity to learn to make things that could change the world. Find Code Club World at **beta.codeclubworld.org**. (HW)

GETTING REPRESENTATION IN MOTOR SPORT ON TRACK

The Hamilton Commission concludes that improving representation of Black people in UK motor sport starts with an equitable STEM education

Gemma Coleman

Currently, only 1 percent of the Formula 1 workforce is Black. As the first — and so far, only — Black Formula 1 driver, Sir Lewis Hamilton was all too aware of this lack of diversity in motor sport, from the drivers to the engineers behind the scenes in the garages and factories. In conjunction with the Royal Academy of Engineering, he formed the Hamilton Commission, which has released its findings from ten months of research into this lack of Black representation in the UK motor sport industry.

Underestimating Black students

The Commission concludes that several factors are leading to a lack of Black engineers, starting with practices in school. If a student wishes to pursue a career in engineering, they need to have studied STEM subjects. By the time students take their first formal qualifications in the UK aged 15 to 16 (GCSEs), Black students are already falling behind. Their academic abilities are frequently underestimated by teachers, with a disproportionate number of Black students put into lower streams for STEM subjects, where their GCSE grades are capped. This often has a knock-on effect preventing these students pursuing mathematics and physics at A level, which in turn stops them from following an engineering career pathway.

Behaviour management practices in school also disproportionately affect Black students, including higher rates of suspensions and exclusions, which hamper their educational experience. A lack of Black STEM teachers in schools and higher education further impacts Black students' experiences, limiting the number of positive role models. This, combined with a lack of understanding about the careers that engineering studies can lead to, leaves Black students feeling that STEM and motor sport aren't for them. These factors are further exacerbated by higher levels of deprivation among Black students, and geographical factors meaning opportunities for work experience at places like racing circuit Silverstone are largely out of reach for Black communities.

©Image craft/stock.adobe.com

The road to action

The report identifies three key strands of action that need addressing to begin solving these issues: support and empowerment; accountability and measurement; and inspiration and engagement. The first strand aims to support Black people's progression into engineering careers. For example, it recommends establishing an exclusions innovation fund, and developing accompanying programmes focused on STEM and motor sport that address the factors contributing to higher proportions of Black students being excluded. It also supports piloting new approaches to increase the number of Black STEM teachers who can act as role models for Black students.

The second strand aims to strengthen accountability for those in authority, evidenced through shared data collection. For example, it calls on the UK's Department for Education to enable easier access to data such as student participation, and attainment by diversity characteristics at different learning levels.

The final strand aims to maximise opportunities to inspire and engage young Black people so they can see themselves in these engineering roles. This includes recommending the development of best-practice guidance for STEM inspiration and outreach activities, to enable inclusive engagement with Black school students.

The Commission hopes that its research and recommendations will lead to more Black people in engineering and motor sport, as well as more Black children studying STEM subjects and pursuing fulfilling related careers.

You can read the full report at **helloworld.cc/hamiltoncommission**. (HW)

EQUITABLE TEACHING: THE WAY FORWARD

A new report analyses whether primary and secondary teachers are sufficiently supported to implement culturally relevant computing pedagogical practices

Gemma Coleman

The Computer Science Teachers Association (CSTA) and the Kapor Center have published the results of a 2020 study investigating how well equipped primary and secondary computing teachers in the USA are to provide equitable computing instruction. The study aims to fill a gap in what is known about computing teachers' demographics, credentials, and experiences, as well as the level of support and resources they have access to for implementing culturally relevant pedagogical practices.

The study was commissioned at a vital time. The technology industry is projected to be a leading industry in terms of job growth this year, catalysed by society's shifting needs and ways of working during the coronavirus pandemic. The report recognises that despite technology's positive impacts, it can also be used to spread misinformation and extremism, and that algorithms can both target and discriminate against Black, Latinx, and Indigenous communities. Computing teachers play an essential role in developing students' computational and critical-thinking skills so that the next generation can build a more inclusive technology sector. But how prepared and supported are computing teachers to do this?

Demographics and credentials

The report found that, much like the general teaching population in the USA, the majority of computing teachers were white (75 percent) and were working in high-income, urban, and less racially diverse schools. This is despite the diverse racial and ethnic make-up of the US student body and indeed, of the population as a whole. In terms of credentials and experience, the data revealed that just under a third of computing teachers graduated with a computing degree, and only 16 percent of teachers had eleven years or more experience in computer science classrooms. Despite these statistics, participation rates in computing-specific professional development remain very low.

Challenges

The study investigated the challenges computing teachers currently face in designing and facilitating equitable computing education. The most commonly reported challenges were insufficient resources, limited budgets, a lack of computing content knowledge, and a lack of student engagement. All these challenges require a collective commitment from schools to allocate time for professional development and budget for resources, and to develop a clear strategy and messaging around the importance of computing pathways to students and parents. Despite these needs, just under a third of teachers reported a lack of staff and administration buy-in for computing education implementation.

Culturally relevant teaching practices

There is evidence that, especially for students who have been traditionally underrepresented, culturally relevant pedagogical practices can increase student engagement. Such practices allow teachers to connect computing to students' lived experiences, value students' voices in the classroom, and create opportunities to critically analyse community issues. Currently, there is a gap between computing curricula content and students' interests and experiences. Just over half of teachers were spending time revising their curricula to make them more relevant, but this was reported less by teachers with less classroom experience and by those teaching in less racially diverse schools.

Teachers also reported a lack of confidence in implementing culturally relevant teaching practices. A lack of confidence was particularly prevalent among primary school teachers and those at an early stage in their careers. Of particular concern is that only 59 percent of white teachers, who make up

Implementing equitable teaching practices will help build a workforce equipped for the future

©Robert Kneschke/stock.adobe.com

the majority of the teaching workforce, felt confident using materials highlighting race and culture (compared to 67 percent of Black, Latinx, and Pacific Islander teachers). One teacher shared, “As a white person, I struggle with connecting computer science to diverse cultures. I allow for personalization and encourage kids with the ‘you be you’ mentality, but I still struggle with knowing how they may learn or connect differently.”

> THERE IS EVIDENCE THAT CULTURALLY RELEVANT PEDAGOGICAL PRACTICES CAN INCREASE STUDENT ENGAGEMENT

Recommendations

The CSTA and the Kapor Center concluded that in order to have a workforce trained for a future in which students can participate equitably in computing education, computer science teachers need the necessary equipment, curricula, and training to lead culturally relevant classes. Here are some of the recommendations for policy and practice they have put forward:

1. Develop incentive structures to recruit, retain, and diversify the pool of computing teachers
There is a national shortage of computing teachers in the USA, particularly those who come from traditionally underrepresented racial and ethnic groups. Schools are more likely to fill and diversify computing roles if they develop new approaches to incentivise teachers to both join and stay in the computing education sphere.

2. Build comprehensive teacher training, certification, and endorsement programmes aligned to an equity-focused computing education framework
Greater alignment between teacher training, licensed programmes, credentialling, and an equity-focused computing curriculum can help increase teachers’ confidence and fill their knowledge gaps.

3. Expand access to ongoing professional development
Participation in ongoing professional development is low, meaning teachers are not keeping up to date with appropriate equity practices for the computing classroom. Schools and districts need to invest in time and budget for such training, especially for primary teachers, early-career teachers, and those teaching marginalised groups. Teachers will then be much better prepared to incorporate these practices in their classrooms.

4. Build a district-wide coalition to champion an equitable computing implementation plan
An effective strategic plan to implement culturally relevant pedagogies in classes is required. This is more likely to be successful when school districts make a collective commitment to prioritising computing education.

If school districts and school leaders can take on board these recommendations, they can help effectively prepare students to become part of a more equitable and socially just workforce of the future.

You can access the report at **helloworld.cc/cstareport**. For more about culturally relevant pedagogy, see page 54. (HW)

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).

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**).

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

© GAYSORN/stock.adobe.com

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.

#INSIGHTS

INSPIRING YOUNG PEOPLE WITH CONTEXTS THEY CARE ABOUT

STORY BY Oliver Quinlan

Many young people get into making things with computers through an early interest in the technology itself. There are many others who may not be as fascinated by technology for its own sake, but given the right context, can see it as a powerful tool to make a difference to the things they care about.

One such compelling context is health and well-being. We see just how many children care deeply about issues in this area every year from the projects entered to Coolest Projects, the Raspberry Pi Foundation (RPF) event showcasing young people's digital projects.

Back in 2018, the RPF carried out some research looking at the stories behind the projects children presented at the events in the UK and Ireland. One of the most highly regarded projects at the UK event was Be Healthy, an app to guide people's diets and health habits. The young creator of this app took a holistic view of health and created a single app that could encourage people to live healthy lives in a variety of ways. They focused very much on the design of the app, starting with the goals they wanted to achieve and then exploring how they would realise them with the technology.

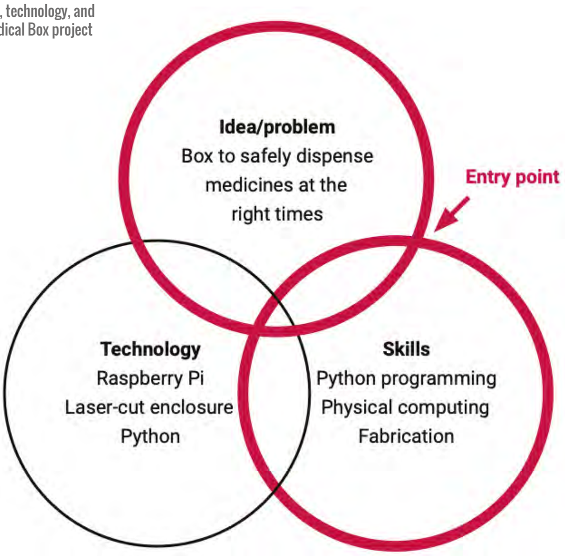
Another was Locking Medical Box, a physical computing project that supported people with mental health issues to manage their medication. The idea for this project came from the interest in health issues of one of the project team and drew on the experience of their parent working in healthcare. The team had worked together before and met through a summer coding camp. For this project, they wanted to take skills they had learnt together and apply them to an area they all cared about. They combined a lot of thoughtful design work with both hardware and software skills to realise their idea as a working prototype.

Ideas, technology, and skills

From our research into these and many other projects, we put together a framework to help us understand the different ways young people approach making digital projects. We found that successful digital projects involve three key areas: ideas, technology, and skills.

What tends to happen is that young people start their projects with an emphasis on one of these areas. Many start with the technology that they have access to and explore its potential uses. This is an important consideration, as a

The interaction of ideas, technology, and skills in the Locking Medical Box project



The Be Healthy app creator at Coolest Projects

lack of access to technology can be very limiting. This approach appeals to young people with a strong interest in technology for its own sake, and allows them to explore the possibilities of particular technologies.

Some young people start with skills they have learnt or want to learn, and create a project to apply these skills. In our research, we found that this approach was less common than the other two. It is a focus that we saw being taken by young people who already had quite a lot of experience with computing and digital making.

Other young people start with an idea in an area they care about and want to make a difference to. This approach can be a challenge for adults to facilitate, as it requires young people to then figure out how they can execute ideas that are often often quite ambitious, using technology they have access to, and skills they can realistically acquire in the course of the project. It is a common tendency to be ambitious with your ideas, but matching them to a comfortably challenging level of skill and the available technology can require some support.

There is evidence from research into formal education that when computing lessons address contexts that young people feel are relevant to them, such as health and well-being, it can help to motivate groups that are usually less engaged with computing, particularly girls (see the 'Further reading' box). This suggests that encouraging young people to approach digital making by focusing on the ideas that matter to them could be an effective way of opening up the activities to a wider range of students.

The RPF is currently working together with Apps for Good (**helloworld.cc/appsforgood**) and the Behavioural Insights Team (**helloworld.cc/thebit**) to explore this sort of approach to computing lessons in schools, as part of the RPF Gender Balance in Computing research project. For more information, and for examples of projects that you could share with young people, see the RPF publication 'How Children Make Digital Projects' at **helloworld.cc/projectsresearch**. (HW)

TIPS FOR TAKING AN 'IDEAS FIRST' APPROACH TO PROJECTS

- Make it clear to young people that the focus isn't on learning about technology for its own sake, but using technology to make a difference to things they care about.
- Share examples of projects linked to topics like health and well-being, and help young people see that they can use digital technology in contexts like this that they may not realise are possible.
- Set aside time away from the technology to talk to young people about the areas that interest them and the issues they care about; understanding this well will help you notice opportunities to build on their interests.
- Try to help learners break down their ambitious ideas into chunks or steps that might be more achievable with the skills, technology, and time available.
- Help learners work out what the minimum viable version of their idea is, so they can get to something functional quickly and feel a sense of achievement; they can always iterate and add to it later on.
- Be prepared for young people to want to do things you might not know how to do yourself. Support them to use forums (if age-appropriate) and online resources. You don't have to know everything, but you can help them figure out how to learn!

FURTHER READING

✓ Hulleman, C. S., & Harackiewicz, J. M. (2009). Promoting Interest and Performance in High School Science Classes. *Science, 326*(5958), 1410-1412. **helloworld.cc/issue17insights1**

✓ Kemp, P.E.J., Berry, M.G., & Wong, B. (2018). *The Roehampton Annual Computing Education Report.* University of Roehampton, London. **helloworld.cc/issue17insights2**

LEARNING ABOUT HEALTH WITH SOCIAL MEDIA

STORY BY Katharine Childs

Opinion is divided on the use of social media as a learning tool in schools. Arguments for the positive benefits of health and well-being apps and digital content are countered by concerns about the risks around inappropriate use by students, and the potential for harm through misinformation. The pandemic has added a new dimension to these considerations as more people turn to social media to support their health and well-being. Some newly published research takes a timely look at how best to use social media for health-related learning in secondary schools.

Young people are increasingly using social media to inform themselves about healthy lifestyles and well-being issues

©akhenatonimages/stock.adobe.com

Teacher and student perspectives

Researchers spoke to students, teachers, and community workers to find out how they thought social media could be used to support informal learning about fitness and healthy eating. Initial opinions were, unsurprisingly, polarised. Students commented on how engaging it was to use social media, and saw it as a valuable source of information and inspiration. On the other hand, teachers highlighted the importance of safeguarding, and observed that uncertainty around appropriate use often led schools to banning all social media interaction in school.

Co-producing learning experiences

After some early conflicting views, the researchers saw ideas emerging about co-producing credible and authentic learning experiences. Students identified their teachers as an important resource for helping them to understand the impact of their own and others' behaviour on social media. Teachers commented that students had expert insights into the variety and type of content available. Their expertise came together to create engaging health and well-being learning activities using social media. For example, one school adapted the This Girl Can sports campaign (**thisgirlcan.co.uk**) and filmed their own students taking part in healthy activities to use as inspiration for other students. In another school, students commented about the value of a talk from a guest speaker who had relevant and authentic experiences to share about healthy eating.

Computing teachers have an important role to play due to their knowledge of the ethical, cultural, and legal aspects of social media. As social media platforms develop and expand, this research points the way to making effective and safe use of digital content to support pupils and teachers to lead healthy lives. (HW)

ACTIVITIES TO TRY

Health and well-being remain high on the agenda for both teachers and students. Here are some ways to use social media and informal learning to support them:

1. Adapt a national health or well-being campaign such as This Girl Can, Mental Health Awareness Week, or #HelloYellow (World Mental Health Day), and make your own social media content for your school account
2. Invite external guest speakers from local charities to discuss ways of using social media to support health and well-being
3. Learn more about leading classroom discussions on the impact of technology with the Raspberry Pi Foundation's free online course (**helloworld.cc/techimpactcourse**)

FURTHER READING

✓ Goodyear, V., & Armour, K. (2021). Young people's health-related learning through social media: What do teachers need to know? *Teaching and Teacher Education, 102.* **helloworld.cc/issue17insights3**

REDUCING ANXIETY IN THE COMPUTING CLASSROOM

STORY BY Henna Gorsia

The coronavirus pandemic has disrupted students' learning for the past 18 months and brought about much stress and uncertainty. It is therefore important to think about ways in which anxiety can be reduced in the computing classroom. Computing programming anxiety can be defined as when students feel afraid and uneasy when writing code.

Earlier this year, five researchers published the results of an investigation into the relationship between academic performance and programming anxiety in first-year IT university students. Their study found that most students avoided programming because it made them feel anxious and uncomfortable, and that those with higher academic achievement were likely to be less anxious than those with lower academic achievement. The researchers went on to suggest that educators should look to create an engaging teaching and learning environment, with increased student participation and interaction.

©hiyusuke9/stock.adobe.com

Many university students avoid programming as it makes them anxious

> LIVE CODING CAN INCREASE INTERACTIVITY AND REDUCE PERFORMANCE ANXIETY

Engaging approaches to coding

One way to achieve this is through live coding. Live coding presents students with a coding lesson in real time. The teacher writes code from scratch in front of students, and the code is presented on a projector for the class to see. The teacher can think aloud as they write their code, and students can code along. This approach enables teachers to ask questions such as, 'What do you think the next line of code should be?' or, 'If I write the code in this way, why will it show an error?' Students can absorb good programming practices and can iterate and test their code during the session. It also enables students to ask questions and increase their interactivity, which can reduce performance anxiety.

Cognitive overload on students' working memory is another factor to consider. Solving problems within 'for' loops can unnerve many students and lead to cognitive overload. Parson's Problems are a technique that can help to reduce this form of anxiety. Parson's Problems are coding challenges in which learners must rearrange a series of lines of code to achieve a specific purpose (see page 70). This allows them to focus on the code structure, rather than syntax. This can reduce cognitive overload, allowing learners to focus on the sequence, and consequently reducing their anxiety. They do not have to worry about missing brackets, colons, or commas and instead can focus on the overall goal they are trying to achieve.

Programming anxiety has been widely documented in research and there are resources available to support teachers to address this challenge. As students adjust to the changes the pandemic has brought, it is important that computing educators look to increase the interactivity and participation of each student. (HW)

FURTHER READING

- ✓ Olipas, C. et al. (2021). The Academic Performance and the Computer Programming Anxiety of BSIT Students: A Basis for Instructional Strategy Improvement. *International Journal of Advanced Engineering, Management and Science,* 7(6), 125-129. **helloworld.cc/issue17insights4**

Some of the M-19 team (Vaibhav Chhabra and Richa Shrivastava top centre)

© Maker's Asylum

MAKER'S ASYLUM

"IT WAS SO MUCH MORE THAN JUST FACE SHIELDS"

Gemma Coleman speaks to Maker's Asylum, a community makerspace in India, to discuss their ethos of open innovation and their effort to create a million face shields

On 23 March 2020, India, like much of the world, went into lockdown in response to the coronavirus pandemic. For a physical makerspace like Maker's Asylum, this could have spelt disaster. "We didn't really want to shut down the Asylum. It was a physical space, we were going through a hard time, and we were all pretty scared," the founder, Vaibhav Chhabra, explained. Instead, they saw an opportunity to continue being a vital social

© Maker's Asylum

Prototypes of the face shields

© Maker's Asylum

Delivering face shields to frontline workers

community space and to respond to the health emergency that was unfolding, "We immediately started looking at ways in which we could contribute and help out … and so we started prototyping".

Like all good tech stories, Maker's Asylum started in a small garage. In those early days, eight years ago, it was a small space in Mumbai for people to share tools and project ideas. "People used to have keys!" managing partner Richa Shrivastava remembers. "Then it grew into a volunteer team and then we started actually working for the organisation. It was very organic." Education has always been part of the makerspace's ethos, Richa explains: "People came to Maker's Asylum to learn about 3D printers, laser cutters, and things like that. It's always been an educational space, but now we've taken it a little beyond that." They now also offer a number of formal programmes for both adults and schoolchildren to learn practical engineering, design, and maker skills.

> " WE WERE ABLE TO SHARE SKILLS AND LEARN TOGETHER AND MAKE, AND THE RESULT WAS BEAUTIFUL

The Innovation School is one such programme for students aged 13 and over. It's split into two parts: skills building and mentorship. Over 54 hours in a one-to-one online format, students learn about programming and flying drones, electronics and robotics, and CAD modelling and 3D design. They can then attend a two-week residency at the Maker's Asylum lab, where students build their own portfolio using the skills they've learnt. "They're really coming up with something of their own, using their own creativity, and we're just helping them out, being there to support them and make it happen!" Vaibhav smiles.

The pandemic has seen some silver linings for these courses, with more programmes moving online and removing the travel barrier to entry. This year, Maker's Asylum has virtually welcomed 200 people from 25 different countries to its workshops to create and make together. "Even digitally, what we're doing is bringing people together, sharing ideas, and building things remotely," Vaibhav explains. If this wasn't enough, the organisation established the M-19 Initiative at the start of the crisis, setting themselves the challenge of designing and creating a thousand face shields for frontline workers.

"We started by prototyping a bunch of designs. We started with paper, then cardboard, and then eventually got to making them with medium-density fibreboard (MDF) on a laser cutter." Vaibhav waves his hand towards the design iterations displayed behind his desk. "Then we showed our designs to a few doctors and got some feedback. We figured out that MDF happens to be a really bad material, because it's porous and we couldn't take that into a hospital. So then we moved to acrylic. And then the aha moment really came when Anool [Anool Mahidharia, Maker's Asylum's co-founder] came up with this amazing material called foam board, which really changed everything for us. It's waterproof, available in every part of the country, super-cheap and laser-cuttable. So it had all the right elements … Plus it was flexible! That was the exciting point: we now had something that worked."

The team showed their new design to more doctors and hospitals, who were excited by what they saw. At first, there were just three or four staff working on bringing the design to life. Then they put out a call for volunteers, and soon, people from all walks of life were showing up to play their part. "There were doctors in the room, psychiatrists, film-makers … All of us had a role, we were a mini factory." For many volunteers, having a sense of purpose and being part of a community was a lifeline during the lockdowns. "A lot of them had no idea how to make anything. And that was the fun part, right? We were able to share skills and learn together and make, and the result was beautiful," Vaibhav explains.

Maker’s Asylum has always been a proponent of open innovation, decentralisation, and openly sharing knowledge. Never has this proved quite as important as during a public health crisis. It wasn’t fast or practical to send face shields around the country to those in need. Instead, by sharing the design and by connecting makerspaces across the country, they were able to bring together makers in 42 cities, towns, and villages and create a million face shields in just 49 days. “India is the land of makers,” Vaibhav exclaims. “The youngest in the collective was 12 years old — he made 300 face shields at home with his family.”

> “ THERE WAS A LARGER PURPOSE, AND THAT WAS WHAT REALLY DROVE EVERYONE AND STILL DRIVES US TODAY

With the same principles, Maker’s Asylum has since gone on to make oxygen concentrators, air-purifying respirators, baby face shields, and rebreathers — units that pump filtered air into a face mask to make breathing easier, particularly for the elderly. “You don’t have access to this level of medical equipment in rural India. So how do you give access to people on that scale so they have the chance of survival?” Richa asks. “Yes, we can buy something, donate it and they will use it. But giving people the knowledge to actually create something and having a more open design philosophy … that was something that resonated with all of us.”

The strapline for the oxygen concentrators embodies this principle: “If you teach a person how to make an oxygen concentrator, they will go on to save a village!” Activating local communities to make oxygen concentrators using locally sourced parts and materials became vital during the second wave of the virus in India. “Most of the oxygen concentrators that were being imported into India started failing, especially around the region that we are in [Goa], because of the climate and the humidity conditions,” Vaibhav explains. “The zeolite inside the concentrators doesn’t last at 95 percent humidity. It becomes like a cake and the concentrator is shot. We were able to see those problems very early on and start coming up with solutions and how we could make them better suited to India’s conditions. The other problem with the concentrators was that they were made for 0–5 LPM [litres per minute], which is a very low flow rate for Covid patients. So the device that we have been building runs at about 15 LPM — this is what we learnt from the hospitals, naval officers, and many more people who were able to contribute and support the collective.”

© Maker’s Asylum

Building an oxygen concentrator

The collective has so far created 30 oxygen concentrators in different parts of the country. “There are groups in Gujarat, Maharashtra, Delhi, Punjab … in every part of India, making them as we speak,” Vaibhav explains. These groups are made up not only of makers, but also of manufacturers such as small shops and factories, and consultants such as hospital workers and zeolite experts. “This has been the beautiful thing about it. Not only have we been able to share what we know with them, but we were able to learn a lot from them about supply chains and scaling our efforts. It was a two-way street.”

So how would Maker’s Asylum sum up the impact of their initiatives during the pandemic? “The learning. All the learning!” Vaibhav laughs. “Not only do we know how to make something new, but it’s the confidence that comes with it. And the impact on maker culture in India — people have a community to share with, get feedback, work together. The power and excitement it gives to each person involved is going to go a long, long, long way! We’ve built a network that is stronger, more resilient, and more confident to create in times of crisis.” The impact on communities is a common theme, with Richa adding, “It gave people a purpose and motivation to get out of that whole mental state of not having the power to change the situation around you. We haven’t felt the pressure of the pandemic as much. There was a larger purpose and that was what really drove everyone and is still driving us today.” (HW)

For more information about the M-19 initiative and other projects, including open-source files, visit: **makersasylum.com**.

COMPUTING AND HEALTHCARE RESEARCH TO INSPIRE STUDENTS

Researchers in the Department of Computer Science and Technology at the University of Cambridge outline their healthcare-related research

Providing students with context about why they are learning something is a valuable engagement tool, especially when that context is as meaningful and widely relevant as healthcare. Here, we hear from researchers in the Department of Computer Science and Technology at the University of Cambridge, who outline their current work at the intersection of computing and healthcare. You can use these profiles to help inspire your students when introducing related topic areas, helping them to understand how the skills they are developing in their computing studies can be built upon and applied by researchers to develop solutions in the healthcare industry. (HW)

PREDICTING MENTAL HEALTH CONDITIONS

NAME: Sarah Morgan

WHAT DOES YOUR RESEARCH INVOLVE: My research sets out to predict and understand mental health conditions, with a particular focus on schizophrenia. Mental health conditions can be extremely debilitating and are still relatively poorly understood. There is also a clinical demand for tools that can predict individual patients' disease trajectories for mental health conditions such as schizophrenia. To help meet these challenges, I apply a range of data science techniques to study both transcribed speech data and magnetic resonance imaging (MRI) brain scans. This includes methods from machine learning, network science, and natural language processing.

A key focus of my research is using brain MRI to study schizophrenia and other mental health conditions. In particular, MRI can be used to investigate brain connectivity by helping us to estimate macroscopic brain networks, where nodes represent large-scale brain regions and edges represent connectivity between brain regions. MRI brain networks from patients with schizophrenia often show altered connectivity patterns compared to healthy volunteers. My research explores both whether we can use these connectivity patterns to predict individual patients' disease trajectories, and what they can teach us about the biological mechanisms underlying schizophrenia.

I also have an interest in using transcribed speech data to predict outcome for schizophrenia. Altered speech is often a core symptom of schizophrenia, and natural language processing markers of speech offer a promising approach to predicting and monitoring disease outcomes over time, in a quantitative way.

WHY IS YOUR RESEARCH IMPORTANT: I hope my research will lead to fresh insights into mental health conditions such as schizophrenia, which have historically received much less attention than other health conditions. Ultimately, I'd like to be able to translate my research to positively impact patients' lives, for example by developing new tools to predict individual disease outcome so that treatments can be better targeted.

COMPUTING TOPIC AREAS TO WHICH TEACHERS CAN LINK THIS RESEARCH:

- Machine learning
- Graph theory/network science
- Natural language processing

© Chinnapong/stock.adobe.com

DIAGNOSING AND TREATING CANCER

NAMES: Helena Andres Terre and Nikola Simidjievski

WHAT DOES YOUR RESEARCH INVOLVE: We develop artificial intelligence (AI) tools and methodologies to help clinicians better diagnose and treat cancer. When a cancer starts to develop, many small signals can be used to detect and potentially assess the severity of its growth based on each patient's personal conditions. These signals or markers are usually identified from different analytics and tests, which are then interpreted by doctors and experts and used for diagnosis and treatment. There are many variables involved in these decisions. These can include the patient's clinical details and history; imaging techniques such as mammograms, MRIs, or ultrasounds to assess the severity and type of cancer; and more recently, genomic information has begun to be used to characterise and predict tumour development.

We are building machine learning models capable of learning from all these different data streams and providing doctors with a detailed and personalised analysis for each patient. In order to do that, we first need to fully understand what is relevant from each data modality. Therefore, we are constantly interacting with experts in each different area to learn and design reliable models. Another important aspect of our research is interpretability. We not only want to build accurate models; but we also need to understand how they make such predictions. So we develop tools that can be interpreted by humans using mathematics and logics, with different approaches such as graphs or rule trees.

WHY IS YOUR RESEARCH IMPORTANT: AI will become a great ally to humans in many aspects of our lives, especially in highly complex issues such as healthcare. With large amounts of data being collected from all over the world, and a healthcare system that is often overwhelmed, the development of reliable and interpretable machine learning tools that help clinicians will change the world of medicine in the next few years.

HELENA: I enjoy developing models that learn and are able to predict real-world problems while helping to tackle one of the most complex current issues in our society. These models need to be precise, interpretable, and safe, and I believe that using well-studied and human-friendly avenues such as mathematics will be crucial for that.

NIKOLA: My main research interests are at the intersection of machine learning and natural sciences (medicine and biology). Addressing data challenges that stem from these domains, and in particular cancer research, inspires and enables the development of novel machine learning approaches and means for analysing complex data. More importantly, however, a main motivation is working on methods that support, assist, and aid clinicians and cancer researchers in the pursuit of better diagnosis, prognosis, treatment, and care, both at the level of whole cohorts, as well as targeted for each patient individually.

COMPUTING TOPIC AREAS TO WHICH TEACHERS CAN LINK THIS RESEARCH:

- Machine learning
- Bioinformatics

ROBOTS FOR WELL-BEING

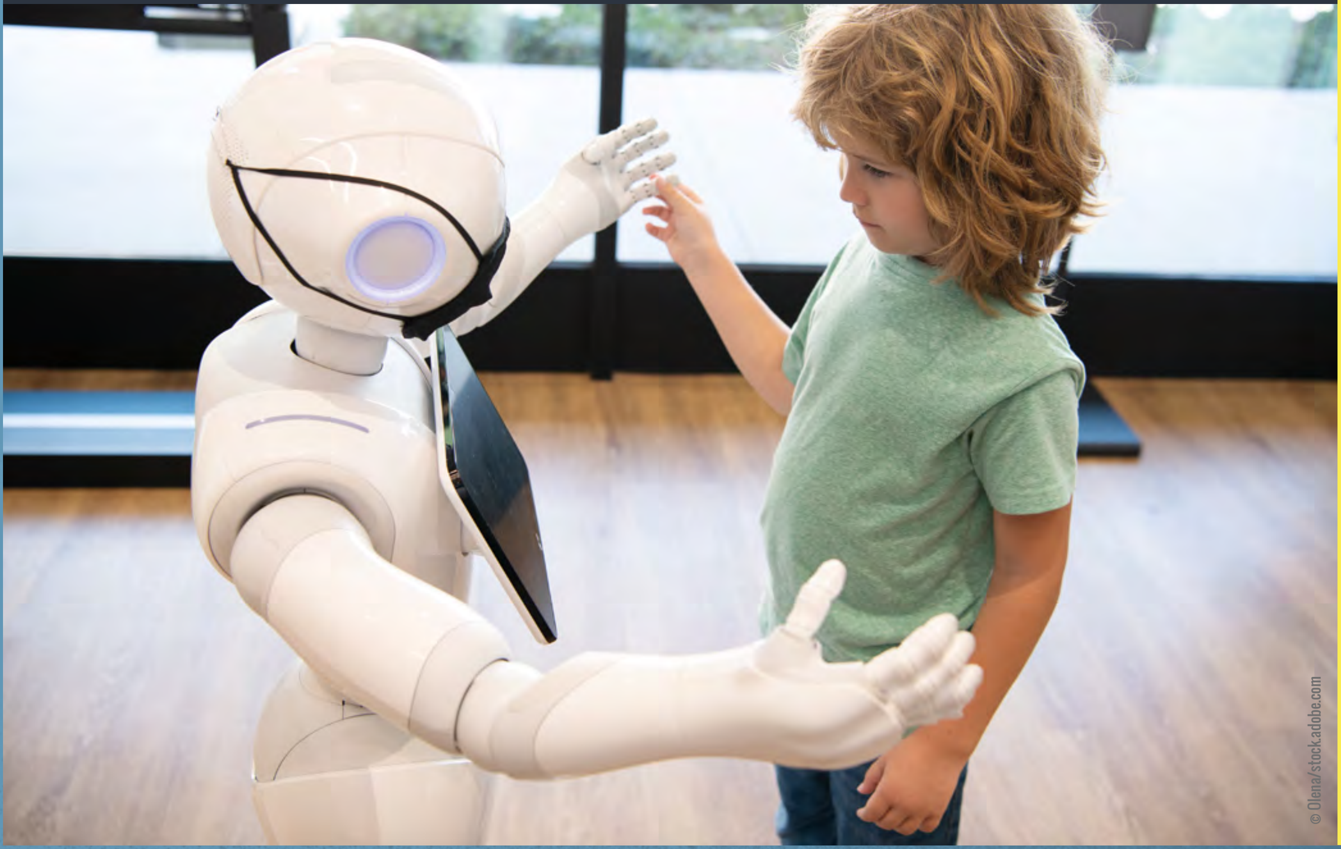
NAME: Indu Prasad Bodala

WHAT DOES YOUR RESEARCH INVOLVE: My work encompasses interdisciplinary themes of human-robot interaction (HRI), social robotics, and psychology. I am working on understanding how human factors such as trust, user perception, and attention affect the way humans interact with and accept autonomous robots. Specifically, I work on developing socially assistive robots (SARs) that can interact with users on a longitudinal basis (that is, with multiple interactions across time) and deliver mental well-being interventions. For this purpose, a teleoperated robotic platform was developed where we replicated the pose and voice of human well-being coaches and projected this onto the Pepper robot (a social, humanoid robot that is able to recognise faces and basic human emotions) to deliver well-being sessions. The data gathered was multimodal in nature, comprising facial affect, body pose, and user ratings, and was used to understand how users interact and perceive a robotic well-being coach from HRI studies. The aim is to study the interactions between users and the robotic coach during the well-being sessions, to understand how users' well-being and perceptions of the robotic coach evolve over time. This understanding will guide the design of efficient robot functionality that enables meaningful user interactions and acceptance.

WHY IS YOUR RESEARCH IMPORTANT: SARs are gaining popularity for use in healthcare-related applications, especially for fostering mental and physical well-being. Despite these developments, there are significant technical challenges in terms of designing robots that meet user expectations and clinical acceptance. My research vision is to develop assistive robots capable of delivering long-term, personalised well-being interventions. The goal of such research is to make social robots even more useful by providing personalised healthcare interventions that allow higher trust and engagement levels for users. Furthermore, these robots will be able to deliver interventions to assist expert human well-being coaches and psychological experts, making the interventions more accessible and affordable. Such accessibility is even more sought-after during periods of lockdowns and social distancing.

COMPUTING TOPIC AREAS TO WHICH TEACHERS CAN LINK THIS RESEARCH:

- Robotics: robot design



© Olena/stock.adobe.com

BUILDING A GLOBAL EDUCATOR FAMILY

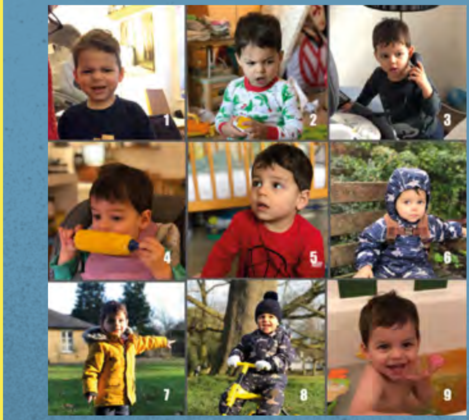
Cat Lamin talks about how building connections and sharing the burden can help make us better educators, even in times of great stress

In March 2020, the world suddenly changed. For educators, we jumped from face-to-face teaching to a stark new landscape, with no idea of how the future would look. As generous teachers pushed out free resources, I felt that I needed to play my part. Suddenly, an idea struck me; in September 2017, I had decided to be brave and submit a talk to PyCon UK to discuss my mental health. Afterwards, several people in the audience shared their own stories with me and let me know that it had helped them just to hear that someone else struggled too. I realised that in times of pressure we need a chance to talk, and we had lost these outlets. In school, we would pop to the staffroom or a friend's classroom for a quick vent, but that wasn't an option anymore. People were feeling isolated, scared, and stressed and didn't have anyone to turn to.

FREDERICK BALLEW, *MINNESOTA, USA*

Staffroom has been the best leap out of my comfort zone that I have ever taken. I have met educators from all over the world and learnt that there are more things that unite us than divide us in this world of education.

Thus, the first Global Google Educator Group Staffroom: Mental Health Matters was launched on 14 March 2020, which coincided with the US government announcing school closures and the UK still waiting anxiously to hear when doors would close. The aim of Staffroom was to give teachers a safe space to talk about how they were feeling under the overwhelming weight of school closures. To say it was a success would be an understatement. Teachers joined the calls from Australia, Malaysia, the USA, Colombia, Mexico, Brazil, Europe, and more!



Which William best represents how you feel?

Which William are you today?

In those early days, we just gave teachers a chance to talk. The format of our meetings was simple: what's your name, where are you from, and then an ice-breaker question such as, 'What colour do you feel like?' or 'What song represents your current mood?' It wasn't long before we hit upon a winning formula by making our own 'Which image are you today?' picture scale (see the 'Which William' image). Using the picture scales allowed people to really express how they felt. Often, someone who had been happily chatting would explain that they were actually struggling to keep their head above water, because a silly image allowed them to be honest.

One of the most important messages from Staffroom was that many people involved with technology in schools were feeling alone. After years of suggesting teachers use technology, they were suddenly being blamed for schools not being properly prepared. They were struggling with not necessarily knowing what to suggest to teachers with technology difficulties, as they were grappling with their own personal lockdown situations. Hearing that other people, all around the world, were experiencing something similar was hugely eye-opening and took a great amount of weight off their shoulders.

We varied the tone of the sessions depending on the attendees' needs. In the first few months, we shared our lockdown situations and our different experiences from across the world. We shared advice and tips, as well as best practice for delivering content, and things that had gone terribly wrong since switching to remote teaching. Or we'd discuss food in different countries around the world (did you know that in Australia, fish and chips is made from shark?) or joke about whether Vegemite was actually an edible product (it's OK; I tried it live on camera during one Staffroom). Other days, we discussed how difficult we were finding teaching, isolation, or life in general during a pandemic.

An honest environment

One of the things that people often mentioned was that Staffroom was a safe place where they felt they could share and be listened to and understood. We made it clear that no one had to speak unless they wanted to. I made a point of always being completely honest about my own mental health. As a person who had suffered from depression and anxiety in the past, it was no surprise to me when I was diagnosed with both near the end of 2020, and I was fortunate enough to get virtual therapy. I shared my story with the group, which allowed attendees to feel more comfortable being open and talking about their own struggles, in some cases leading to their own diagnosis and getting some much-needed support.

People would join Staffroom to share new jobs, engagements, even cross-country moves, but equally they would join after losing a loved one or hearing of a pupil sick in hospital. Staffroom became a safe haven for teachers, coaches, IT directors, and pretty much anyone involved in technology within education. It is a place where we could bond over shared experience, share a joke, ask questions, get ideas, and even plan our futures.

Alongside Staffroom, I also built a website to allow teachers to share their mental health stories and to feel a little less alone (**mentalhealthineducation.com**). I continue to host regular Staffrooms, although less frequently. Just 18 months ago, we needed a chance to talk three times a week, but now we meet two or three times a month instead. You can find current staffroom dates at **globalgeg.org/events**. If you take one thing away from this article, however, it is this: do not underestimate the power of connections, or of sharing your story. (HW)

> IN TIMES OF PRESSURE WE NEED A CHANCE TO TALK, AND IN THE PANDEMIC WE FOUND WE HAD LOST THAT OUTLET

PILY HERNANDEZ, *MONTERREY, MEXICO*

Staffroom for me is a place and time to connect with other teachers from around the world. I remember seeing the calendar invites by mail and I kept thinking I should join but was afraid to do it. The first time I did it, I listened first and it made me realize that my struggles during pandemic online teaching were the same struggles as everywhere else.

CAT LAMIN

Cat is a Raspberry Pi Certified Educator, CAS Master Teacher, and Google Certified Innovator. She works as a freelance trainer and coach, supporting schools with digital strategy and enabling educators to use technology more effectively. Cat also runs a regular mental health Staffroom, for which she was awarded a Mental Health Champion Award from Edufuturists (**@CatLamin**).

ABID PATEL, *LONDON, UK*

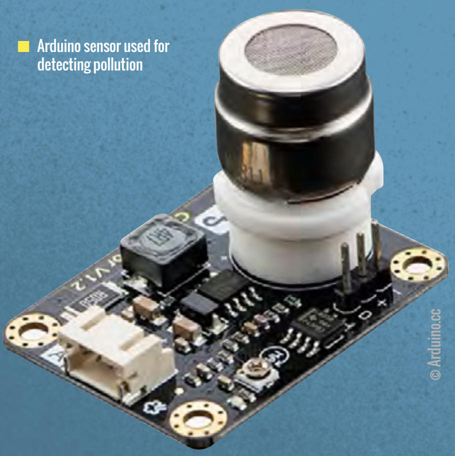
As someone who thrived from having in-person connections and networking opportunities, lockdown hit me hard. Staffroom really helped to keep those connections going and has developed into such a lovely safe space to talk and connect with others.

PROTECTING CHILDREN FROM BREATHING HAZARDOUS AIR

James Abela shares how his computer science students solved the very real problem of predicting air pollution and looks at the classroom environment that made this possible

In 2018, Indonesia burned approximately 529,000 hectares of land. That's an area more than three times the size of Greater London, or almost the size of Brunei. With so much forest being burned, the whole region felt the effects of the pollution. Schools frequently had to ban outdoor play and PE lessons, and on some days schools were closed completely. Many schools in the region had an on-site CO_2 detector to know when pollution was bad, but by the time the message could get out, children had already been breathing in the polluted air for several minutes.

My Year 12 students (aged 16–17) followed the news and weather forecasts intently, and we all started to see how the winds from Singapore and Sumatra were sending pollution to us in Kuala Lumpur. We also realised that if we had measurements from around the city, we might have some visibility as to when pollution was likely to affect our school.


Arduino sensor used for detecting pollution

© Arduino.cc

Making room for student-led projects

I've always encouraged my students to do their own projects, because it gives programming tasks meaning and creates something that they can be genuinely proud of. The other benefit is that it is something to talk about in university essays and interviews, especially as they often need to do extensive research to solve the problems central to their projects.

This project was much more than this: it was a genuine passion project in every sense of the word. Three of my students approached me with the idea of tracking CO_2 to give schools a better idea of when there was pollution and which way it was going. They had had some experience of using Raspberry Pi computers, and knew that it was possible to use them to make weather stations, and that the latest versions had wireless LAN capability that they could use. I agreed to support them during allocated programming time, and to help them reach out to other schools.

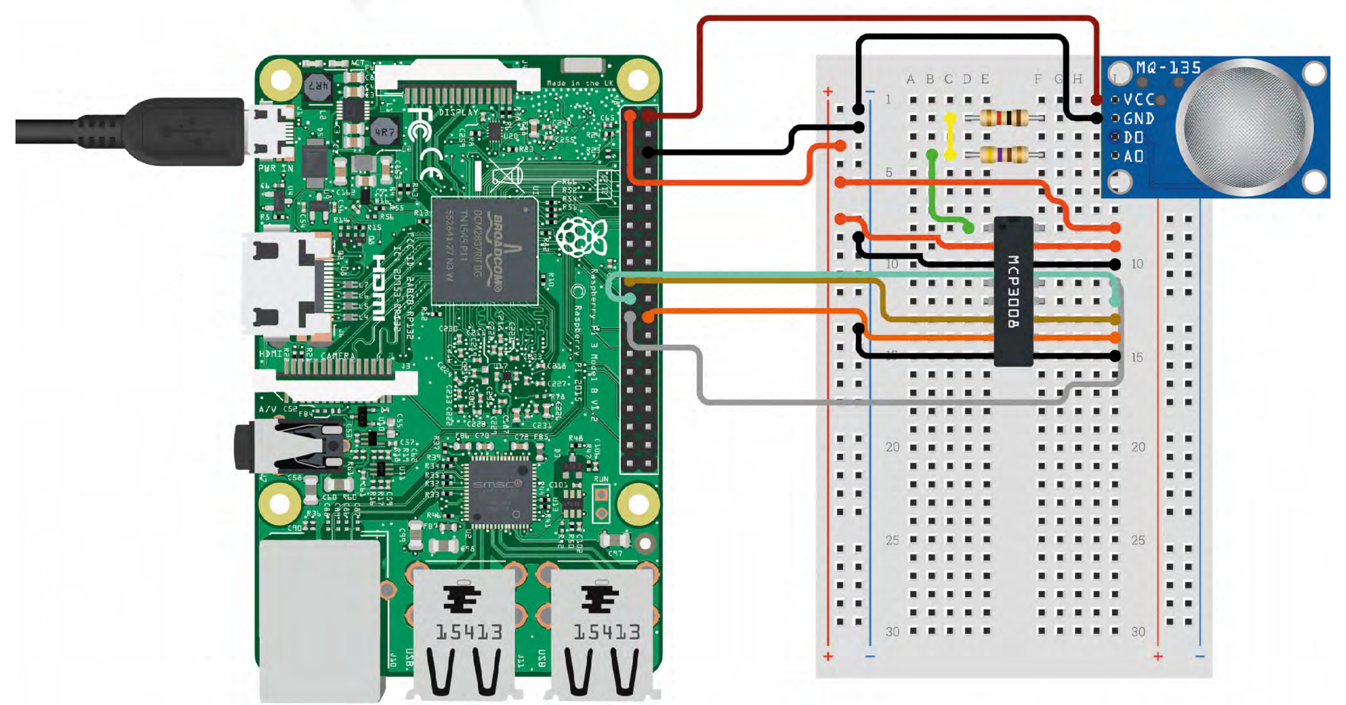
I was able to offer students support with this project because I flip quite a lot of the theory in my class. Flipped learning is a teaching approach in which some direct instruction, for example reading articles or watching specific videos, is done at home. This enables more class time to be used to answer questions, work through higher-order tasks, or do group work, and it creates more supervised coding time.

I initially started doing this because when I set coding challenges for homework, I often had students who confessed they spent all night trying to solve it, only for me to glance at the code and notice a missing colon or indentation issue. I began flipping the less difficult theory for students to do as homework, to create more programming time in class where we could resolve issues more quickly. This then evolved into a system where students could work much more at their own pace and eventually led to a point at which older students could, in effect, learn through their own projects, such as the pollution monitor.

Building the pollution monitor

The students started by looking at existing weather station projects — for example, there is an excellent tutorial at **helloworld.cc/weatherstation**. Students discovered that wind data is relatively easy to get over a larger area, but the key component would be something to measure CO_2.

We found a sensor (the CCS811 sensor module) on a Malaysian site called Lazada. It was designed to work with an Arduino, and so we connected our Raspberry Pi via its USB port to an Arduino and so to the CO_2 sensor. You could also order a more accurate sensor directly from the Arduino store (the MG-811 sensor) or order a variety of sensors from eBay, such



Circuit design of the CO_2 sensor using just Raspberry Pi *Designed on circuito.io*

as the MQ-135 hazardous gas sensor. We then used the 'Get started with Arduino' guide (**helloworld.cc/getstartedarduino**) to help us connect the two together. The advantage for us was that the CO_2 sensor module we bought was designed to interface with the Arduino, so it was easier to install. It is also possible to connect Raspberry Pi directly to such a sensor using a breadboard, an analogue-to-digital converter (MCP3008 will work), a 1K resistor, and a 470 ohm resistor (see the circuit design image above).

We were very pleased to see that data started to come through showing us the CO_2 levels. Our plan was to run the Raspberry Pis headless and export this data to Google Sheets. We found an excellent way to do this in Python using the Google Sheets API (**helloworld.cc/sheetsapi**). This meant that our spreadsheet was automatically loaded with real data and from there, we could make a visualisation to show the CO_2 data as it was being generated. We also contacted other schools around Kuala Lumpur to see if they would be interested in putting a device on their roof, and most were interested in the idea.

Beaten to the punch

We were not the only ones with such an idea and around this time, a company called IQAir began selling the AirVisual Pro, which did almost exactly what we hoped to do and did it incredibly well. Schools were already very receptive to the idea and quickly invested in the technology. It is still very impressive to think that three Year 12 students came up with an idea that solved the very real concern of pollution visibility and were only fractionally behind a commercial solution.

This project really helped these students to decide whether they enjoyed the hardware side of computing, and solving real-world issues really encouraged them to see computing as a practical subject. This is a message that has really resonated with other students, and we've since doubled the number of students taking A level computer science. Since doing this project, I've encouraged students to take on the Extended Project Qualification (**helloworld.cc/extendedproject**). This will give them time to really explore concepts and allow them to put their programming to good use, tackling problems that interest them and that the world needs solving. (HW)

> IT WAS A GENUINE PASSION PROJECT IN EVERY SENSE

JAMES ABELA

James is the head of computing at Garden International School in Kuala Lumpur, Malaysia. He is an RPi Certified Educator, founder of the South East Asian Computer Science Teachers Association, and author of *The Gamified Classroom* (**@eslweb**).

READ, WRITE, TALK, LISTEN, AND MOVE

A computer science teacher shares how using this simple approach to lesson design can positively impact students' social, emotional, and physical health

"I've been staring at a screen for the last 90 minutes; I'm too tired to make it through history today." I overheard that comment as a student left my classroom for her next lesson, and I knew I needed to adjust my class format. Our secondary school had recently moved to 90-minute periods, and although everyone was engaged and seemed to be enjoying their selected projects, I realised I needed to implement more breaks and balance into my screen-dominated class time. I needed to do this not only to improve the health of my students in my classroom, but also to encourage healthy habits outside the classroom. Whether it's eye strain, obesity, or social withdrawal, prolonged screen time and a lack of movement can become an unhealthy habit for students, and can turn other students away from computer science.

LESSON EXAMPLE

TOPIC: Computing ethics

TIMING: 90 minutes

- **READ:** Read one of four news articles about cybercrime
- **WRITE:** Summarise article details and reflect on motivations/societal impacts
- **TALK:** Share responses and discuss in small groups and then as a class
- **LISTEN:** Listen to the discussion and lecture
- **MOVE:** Initially, students move to a different corner of the room to read one of four articles and discuss in small groups. Then, students move back to their seats for the whole-class discussion and lecture. Midway through the class, students take a walk outside.

I needed a simple strategy for lesson design that would help me to provide more balance in my classroom. Read, Write, Talk, Listen, and Move became that strategy. This approach allows students time to do all those things in a single lesson: Reading to help students gain new information and hear another point of view. Writing, to learn and to express themselves. Talking and Listening, to both hear and to be heard. And finally, Moving, to keep the blood flowing and energy levels high.

© Bouncybands

Bouncybands can help students work off excess energy

© Rafael Ben-Ari/stock.adobe.com

Students can be given a discussion prompt and invited to take a walk midway through their class

Balance in the computing classroom

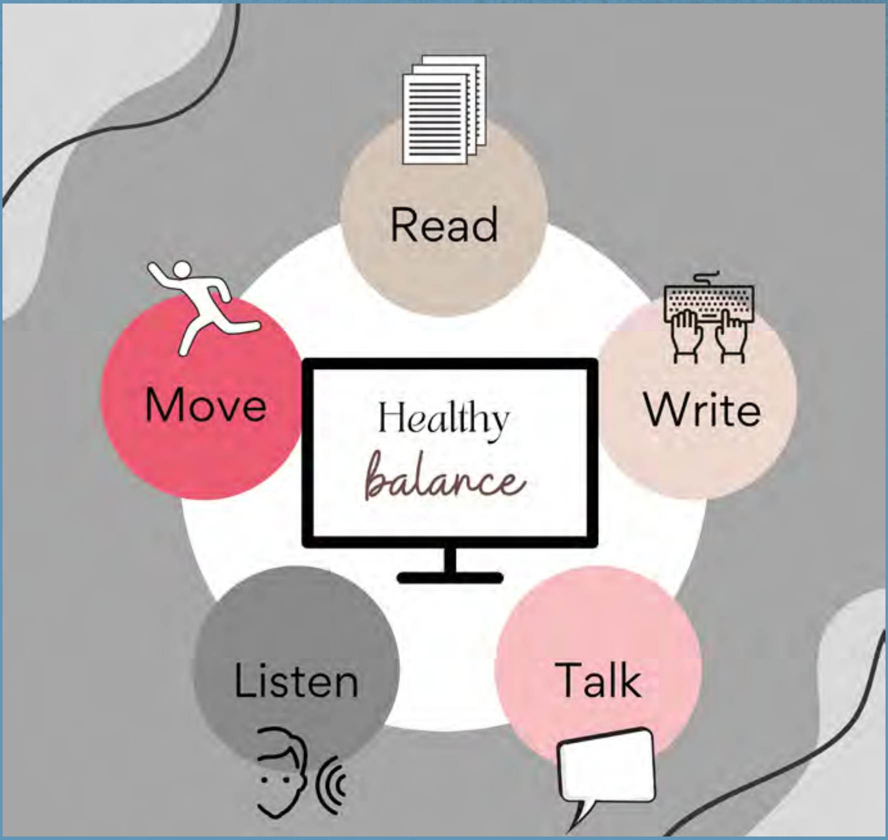
My primary-secondary school had recently been promoting this strategy as a way of improving engagement, but I found that it also promotes physical, emotional, and social health. Reading and writing often take place at the start of my lessons: students are asked to read and then respond to some information that sets the theme for the day. This can also naturally lead into talking and listening, as pairs share their responses. As students move into coding work, on their own or in pairs,

LESSON EXAMPLE

TOPIC: Coding loops

TIMING: 90 minutes

- **READ:** Read prompts in an online coding platform such as Code.org
- **WRITE:** Summarise the benefits of using loops
- **TALK:** Partners talk to each other as they pair-program
- **LISTEN:** Partners listen to each other as they work together in their pair programming task
- **MOVE:** Unplugged dance activity demonstrating the efficiency of using loops in a set of directions (clap, clap, clap, versus clap 3x)



Credit: Gayle Van Tol

Incorporating these elements in computer science lessons can be an effective strategy for encouraging health and balance

> THE READ, WRITE, TALK, LISTEN, AND MOVE STRATEGY ALLOWS STUDENTS TIME TO DO ALL THOSE THINGS IN EACH LESSON

they are given more opportunities to reflect on and share their solutions and points of pride in their work.

My classroom is a traditional computer lab, with rows of computers tethered by wires to a specific spot and fixed login locations that keep students at the same desk. However, I've found that I can incorporate movement with gallery walks, seat stretches, and by attaching stretchy bands (Bouncybands) across the legs of students' seats for their feet to move, to channel excess energy. In addition, one of the most important movement opportunities I use is a walk outside midway through class. Using the same short route, students are sometimes given a reflective or social/emotional prompt to discuss with a classmate as the fresh air and sunlight hit their bodies. Most students are eager for the refreshment, and with an adequate heads-up, those few students who are hesitant to leave their screens come to appreciate the break.

Incorporating the Read, Write, Talk, Listen, and Move framework has helped promote physical, social, and emotional health in my classroom. Try it for yourself and see if you can achieve a healthier and more balanced computing classroom. (HW)

GAYLE VAN TOL

Gayle is a computer science teacher at Peak to Peak Charter School in Lafayette, Colorado, USA. Originally a life science teacher, she transitioned to teaching computing to pupils aged 11 to 18. She focuses on student choice and agency in the courses and curriculum she has designed (**@MrsVanTol**).

ACTIVE CODING WITH THE SUMMER OF SPORT

Kat Leadbetter shares how young people across the world challenged themselves to create sporty coding projects as part of the Code Club Summer of Sport

How can sport and coding interact? That's exactly what young people across the world, from Scotland to Libya, contemplated this summer when they took part in the Summer of Sport: a free, fun activity linking coding and getting active.

With Euro 2020, the Tour de France, the Olympics, Paralympics, and many more pandemic-delayed events taking place this summer, sporting fever was everywhere. Physical exercise and being active are essential aspects of well-being and an important part of recovering from the impact of the pandemic. As a programme supporting young people in schools and educational spaces, Code Club looked to see how we could help them grow their coding skills and think about being active, while connecting with their peers in the spirit of the sporting calendar.

Code Clubs help young people aged 9–13 learn to code through free, informal, educator-led coding sessions. At club sessions, young people use step-by-step instructions to create games, animations, and web pages, developing their coding skills in a range of different programming languages. As well as teaching young people the fundamentals of programming, Code Clubs can help develop soft skills like problem-solving and confidence, and build creativity as children adapt and personalise the things they make.

The Summer of Sport was designed as a way to extend these ideas by connecting young people with their hobbies and interests, asking them to use Scratch to share about sports they enjoy, tell us about an athlete they admire, or to create a sport or activity-themed game, animation, or project.

Feedback from educators and volunteers has consistently shown the value of linking coding topics to relevant themes in curricula and real-world scenarios that impact young people's lives. Embedding coding within real-world contexts and topics can support educators and young people alike to have meaningful coding experiences.

Low barrier to entry

Code Clubs often pause their sessions during the school holidays. With many spaces only recently returning to in-person learning, the holidays also meant a pause in supporting catch-up activities for young people, developing learning that has been disrupted over the past year. For clubs choosing to extend their sessions into the break, or those who wanted to offer children an activity to try at home, the Summer of Sport was an easy way to keep coding.

The Summer of Sport was open to all young people across the world and used existing coding projects that many Code Club members were already familiar with. To take part, participants simply needed to select a Scratch project from a suggested list and get to work. Beginners could follow instructions to complete a simple project, while those with more experience could use a project as a starting point and add their own adaptations, or code something entirely new to share with us. The only rule was that finished projects had to have something to do with sport. Every participant could download a certificate to celebrate taking part and was invited to add their project to the Summer of Sport Scratch studio. You can take a look at the studio at **helloworld.cc/sportstudio**.

Global entries

From basketball games to penalty shoot-outs, we saw amazing entries from young people across the world. In one entry, players helped the Scratch Cat navigate ski slopes while avoiding an onslaught of rocks and multicoloured snowmen (**helloworld.cc/skicat**). We loved how this project took the original 'Scratch Cat goes skiing' project and added some creative changes. Haya, aged 13, from Baghdad in Iraq, thought carefully about why she chose to create a skiing game: "I chose this design because I love skiing. This sport is fun because it is a competitive game for everyone. It made me think of ways to combine sports with programming."

In another shared game, players were asked to use their mouse to catch as many baseballs as they could in 30 seconds (**helloworld.cc/baseball**). Young coder Hawra, aged eleven, from Najaf in Iraq, followed her passion for basketball. She created a basketball game because she loves to play the sport in her spare time (**helloworld.cc/basketball**). She told us, "Sports are very useful, especially basketball, because we have to jump and run, and this makes the body active. When I exercised, I thought about how to design a game about this sport in an interesting way."

As well as games, the Summer of Sport received plenty of Scratch animations. One such animation showed a scene of two friends horseback riding together, using simple code and changing costumes to simulate movement, while another showed a diver jumping from the high diving board.

> FEEDBACK FROM EDUCATORS HAS SHOWN THE VALUE OF LINKING CODING TOPICS TO REAL-WORLD CONTEXTS THAT IMPACT PEOPLE'S LIVES

One project included a Scratch animation of two friends horse riding together

For club leader Ali Alzubaidy, the Summer of Sport provided a chance to connect with his club members after the pandemic: "We have missed the positive energy of such workshops, as we are only contacting our club via online meetings. The Summer of Sport activity was a brilliant idea which kept the active energy of our club members, encouraging them to think about their health and keep doing sports at home [with] the aid of coding magic."

Themed coding collections

In the coming months, Code Club will be sharing more collections of themed projects. For World Space Week (4–10 October) we've curated a collection of projects themed around all things out of this world. Our next project collection will be based on the environment, to mark the United Nations Climate Change Conference (COP26) from 31 October to 12 November. Each project collection is designed for all levels of coder; you can find out more at **blog.codeclub.org**.

To find out how you can start a club in your school, head to **codeclub.org**. (HW)

SUMMER OF SPORT PROJECT IDEAS

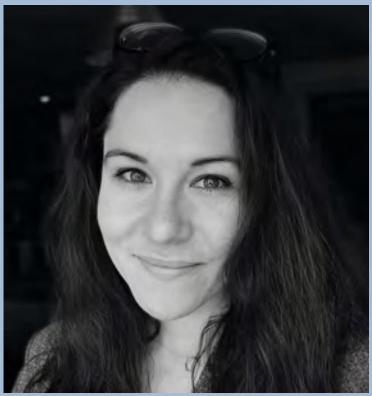
Beat the goalie: Create a two-player football game (**helloworld.cc/goalie**)

Sprint: Code a sprinting game, then race to the finish line (**helloworld.cc/sprint**)

Synchronised swimming: Code a synchronised swimming routine using loops and clones (**helloworld.cc/swim**)

Chatbot: Tell us about your favourite sport (**helloworld.cc/chatbot**)

Relax and stretch: 'Cool down' by creating a program to demonstrate mindful, relaxing exercises (**helloworld.cc/relax**)



KAT LEADBETTER

Kat is a Code Club community manager, supporting educators and volunteers to run free coding clubs for young people aged 9-13. She works to help grow the community, promotes exciting opportunities and resources for clubs, and runs online workshops for the community.

THE LANGUAGE OF EMOTIONAL RECOVERY

As educators, we need to know how to discuss and support emotional experiences; **Georgie Ford** introduces a new framework to aid post-pandemic emotional recovery

Resilience. Recovery. Building back better. Reconnection. These are common words in the current landscape, and as we return to educational settings and workplaces from various lockdowns, there must be some consideration for how the language used to brand this return can significantly impact the well-being of students and staff. Institutions with well-being at their heart should define a meaning of recovery that is individual to them and their authentic well-being purpose. Weston College in Somerset, UK, where I work, has developed an emotional recovery framework and accompanying training to do just this. It has been driven by our transformational principal and chief executive, Dr Paul Phillips, who possesses a genuine belief in well-being mechanisms for every person within his organisation.

The framework

The framework is adaptable to your organisation or educational establishment. It is made up of four aspects: emotion differentiation; individualised consideration; the emotional cycle of stress; and false positivity.

Emotion differentiation promotes the importance of being able to distinguish between emotions and label them. If humans are unable to do this (and it is quite a skill), then we are unlikely to solve and support emotional problems effectively. We should encourage exploration of emotions and break down broader experiences such as 'I'm stressed' or 'I'm angry'. Breaking these down further allows for accurate problem-solving as we can start to understand the precise cause of our feelings. Many emotional literacy frameworks begin with the art of accurately recognising what we are feeling because there are significant differences between relationship stress and work stress, for example, and they require different support solutions.

The second aspect of the framework equips people with the skills to individualise their communication about emotions. It encourages the consideration of different people's emotional experiences and schemas that will dictate how we hear communication. For example, a seemingly simple question such as 'How are you?' requires the responder to connect to their inner emotions, which we aren't often great at, and they may respond with a perhaps inaccurate 'I'm fine.' Think about the words you're using and what they trigger in others. Here, for example, it may be best to use, 'How has your day been?' or a question more specific to the person you're asking.

The emotional cycle aspect of the framework helps participants to understand the emotional aspects of stress, burnout, and the chronic invisible stressors of Covid-19. It is often not enough to remove the external stressor, as we are emotional beings with cognition. The emotional experience that accompanies stressful events requires processing; we must give ourselves the time and space to process our emotions from beginning, to middle, to end.

The framework culminates with an exploratory journey through the critical skill of empathy and the ability to accept and allow authentic emotional experience within our environments. As humans, we often deploy mechanisms such as humour and deflection when we feel uncomfortable and wish to 'fix' our own or others' experiences. In some situations, like the pandemic, it may not be possible to silver-line. And by doing this, we actually risk suppressing and undermining emotional experience and the opportunity for true resilience to flourish.

Learning how to label and talk about our emotions accurately is vital to enable effective problem-solving and to support mental and emotional health. The framework was launched in September as a Weston College training initiative alongside the SEND Centre for Excellence, and includes live training, environmental support sessions, and networks (**weston.ac.uk/mentalhealth**). Please contact **georgie.ford@weston.ac.uk** for more information. (HW)

GEORGIE FORD

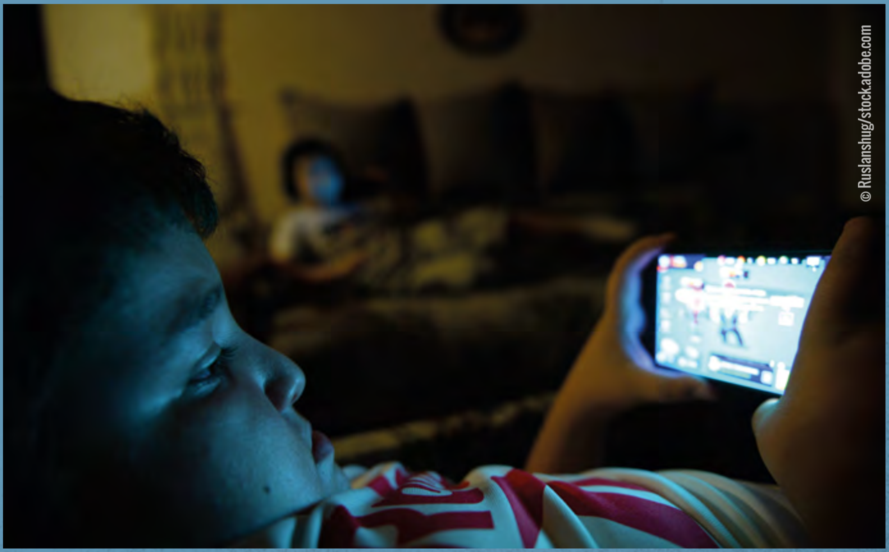
Georgie is the lead specialist practitioner in mental health at Weston College, UK (**@westoncollege**).

EYE HEALTH AND THE PANDEMIC

Mark Tetteh Azietaku shares his findings from research into eye problems among his students during the pandemic

The upsurge in the use of computers and the resulting health effects for children during the pandemic have been a major concern to many teachers and parents. The use of computers has risen exponentially, according to Common Sense Media, a non-profit organisation helping parents and schools navigate media, with teenagers in the USA spending an average of seven to eight hours a day online. According to the American Optometric Association (AOA), computer vision syndrome (CVS), or digital eye strain, is a group of eye- and vision-related problems resulting from the prolonged use of computers and other digital screens. The American Academy of Pediatrics (AAP) recommends that children aged two to five should spend a maximum of an hour per day on an electronic gadget for non-educational purposes, while those aged six and above should be encouraged to have healthy screen habits.

With these stats in mind, I started to make my own observations about my students. I observed that before the pandemic, about a fifth of my students wore glasses. Returning to school after various lockdowns, I noted that this had jumped up to more than half of my students. Midway through this year, I started a survey to uncover whether students were suffering from CVS and investigate the strategies they were using to prevent it.



© Ruslanshug/stock.adobe.com

Children should be encouraged to have healthy screen habits, including limited non-educational screen time

Survey results

Two hundred students aged 16–18 completed the survey, representing 70 percent of the upper-secondary student population. The population sample size was fair and gave a true representation of our students, who were mostly attending their classes online. The research asked students questions about their estimated average screen time over a 24-hour period; whether they were wearing glasses before the pandemic versus during the pandemic; whether they visited the optometrist during the pandemic; and what strategies students were using to prevent CVS.

Screen time, defined as the total number of hours students spent using their computers, was very important to the research. This was used to estimate the time students were online or using their gadgets to learn or do homework. It also determined whether the usage was prolonged or normal. From the survey, it

MARK TETTEH AZIETAKU

Mark is a senior information technology teacher and a system analyst consultant for New Nation School, a Cambridge affiliate school in Ghana. He is a Plus member of the Computer Science Teachers Association.

emerged that 80 percent of students spent more than eight hours on their electronic gadgets over 24 hours, and 15 percent spent between five to eight hours on devices. The remaining five percent spent less than five hours using their screens. These results were not a surprise, because from the daily school timetable we can see that students are expected to be in five different one-hour classes, and that's before considering homework and research work. The answers we got from this survey question suggest that students spent most of their time using computers and electronic gadgets for online classes and homework. These very long hours in front of a screen suggested that there was a very real possibility of this negatively affecting students' eyes.

Eye problems

Building upon the first research question, the condition of students' eyes before and during the pandemic was considered. Prolonged screen time skyrocketed during Covid-19, and knowledge of the health of the eye before and during the pandemic gives us a detailed understanding of what changed over the affected period. This research question also tells us which of the students already had eye problems before the pandemic and which have problems that emerged during. Only 9 percent of students answered that they had an eye problem before the pandemic. Of the 91 percent of students who had no problem with their eyes before the pandemic, 18 percent confirmed that they have since been diagnosed with CVS and are wearing glasses now, while the remainder do not have any traces or signs of CVS.

During the pandemic, many schools across the world stipulated health and safety guidelines for students when using computers (for example, **helloworld.cc/iboscreens**). We included a research question about what strategies students were using to prevent CVS. The almighty 'taking breaks' was included in their strategies, as it was the popular one that most organisations were using and that school authorities had included in their guidelines. Out of the 73 percent who had no eye health issues, 73 percent rated taking a two-hour break as their top strategy pre-pandemic, but this wasn't always viable during Covid-19. During the pandemic, 53 percent of those students with no eye issues decided to reduce the brightness of their monitors, 12 percent used anti-glare glasses or screens, and the remaining 8 percent continued to take a two-hour break during their computer usage. Most of them were of the view that taking a long break disrupted their classes and that anti-glare glasses were too expensive. Their only viable option was to reduce the brightness of their monitors.

© Hedgehog94/stock.adobe.com

It's important to make the most of outdoor time away from screens!

Recommendations and conclusions

Long periods of screen time for students were inevitable because most schools during the pandemic were holding classes online. A combination of strategies needs to be employed to prevent CVS, rather than emphasising only one. This will give

© Prostock-studio/stock.adobe.com

> 80% SPENT MORE THAN EIGHT HOURS A DAY ON THEIR DEVICES

students the flexibility to choose any of the strategies based on the situation they find themselves in. It would be beneficial if teachers were trained to pause online lessons at suitable times for students to take a break, as well as schools creating a shared calendar with set break times that all students and teachers can observe. Guidelines could also be posted on virtual waiting rooms so that students can adjust their settings before joining a class.

The survey aimed to build a foundation for examining computer usage and its related health risk. A further study to determine the correlation between computer use and CVS will be conducted in the near future to deepen our knowledge in this field of study.

If you'd like to explore more about reducing screen time, check out the next article about running a digital detox day. (HW)

TOP TIPS FOR LOOKING AFTER YOUR EYES

Hello World caught up with Professor Shahina Pardhan, director of the Vision and Eye Research Institute at Anglia Ruskin University in the UK, to hear her recommendations for protecting your eyes when using digital devices.

GOOD EYE HABITS:

- Any uncorrected vision problems should be adequately dealt with by an optometrist. It is important to have the correct spectacle prescription for your screen distance, and vision anomalies should be addressed by exercises, glasses, and other optical corrections. This will help to reduce ocular symptoms of digital eye strain.
- Blinking exercises can be helpful for people who suffer from dry eyes. You can set a reminder on the computer to blink ten times once an hour.
- The UK's Health and Safety Executive advises more regular, short breaks from the screen rather than longer ones less often: five to ten minutes every hour is better than 20 minutes every two hours. A 20-20-20 rule is also useful: take a 20-second break every 20 minutes, to gaze a distance at least 20 feet away. Frequent, short breaks can relax accommodative and binocular vision.

SMART USE OF DIGITAL SCREEN TIME BY USERS AND SCHOOLS:

- Ensure that time spent on digital devices is maximised for learning and that less digital time is used for recreational activities.
- Working within government rules during the pandemic, fully utilise time outdoors at school. Increased physical activity and outdoor play have been shown to lead to longer sleep in preschoolers and also a reduced risk of myopia.
- Parents can reduce the detrimental effects of digital screen time by monitoring recreational usage and setting reminders to disconnect. A daily schedule for when devices can be used would be beneficial. The continuous use of digital devices for non-educational purposes should be limited to less than 15 minutes per day and a cumulative duration of less than one hour a day.

DIGITAL BEHAVIOUR CHANGES INTERVENTIONS:

- A cluster randomised controlled trial in China found that children who were exposed to a digital behaviour change intervention reported reduced levels of eye strain during Covid-19 online schooling.
- The intervention group reported a reduction in screen time and eye strain at the end of the two-week trial, suggesting that digital behaviour change interventions can reduce eye strain in children undergoing digital learning.

RUNNING A DIGITAL DETOX WITH YOUNGER PUPILS

John Parkin shares how he ran a digital detox day with a class of reception children and some tips on how to do one with your class

We are constantly hearing in the media that many people, adults, and children alike, are having too much screen time, and that we need to have a healthier relationship with technology — in fact, you only have to read the previous article! Guidance released by the World Health Organization in 2019 recommended that children aged three to four have a maximum of one hour of screen time a day. Research during the pandemic, however, has shown that the amount of screen time for children during this period has increased significantly in a number of countries. In light of concern about the impact of the internet on physical health and mental well-being, the UK Council for Internet Safety published the Education for a Connected World framework to help schools plan a curriculum to help children use technology in a responsible way (**helloworld.cc/connectedworld**). Consequently, I wanted to run a digital detox day with my reception class of children aged four to five, to help them develop healthy technology habits.

Preparing for the day

As I was teaching young children, I wanted to make sure that I explained what I meant by a digital detox. To help me with this, I found two excellent picture books that would help pupils understand the concept. *Unplugged* by Steve Antony is about a robot who spends all day plugged into a computer, but after a power cut realises the joy of going outside. Similarly, *Dot.*, by Randi Zuckerberg, is about a girl who discovers the fun she can have outside and away from technology.

I read *Dot.* to my class a few days before the digital detox day so they could understand what would be happening and why. Through reading a story, the children could identify how a character in a book used technology a lot, and how she could also enjoy other things. As a class, it helped us to discuss whether it would be good (or indeed fun) to use computers all the time. I also discussed with pupils what types of technology we had in the classroom which we should not be using on the day.

> CAN YOU TEMPT YOUR PUPILS TO HAVE AN EVENING AWAY FROM ONLINE GAMING AND SOCIAL MEDIA?

The children identified that the interactive whiteboard, desktop computer, iPads and Bee-Bots needed to be off limits. We also discussed whether the digital detox applied to just the children, or to me as well: the children thought I should have a digital-free day, too! We also brainstormed all the things we could do instead, such as playing with construction toys inside, playing football, and making magic potions in the mud kitchen outside.

How the day went

When the day arrived, the pupils were excited to see what it would be like. We began the day by reading *Unplugged* to remind ourselves of what we were doing. Then, we went off to do activities away from technology. Some children wanted to practise their writing and produced signs for the equipment we were not using, to remind people not to turn them on. The children enjoyed a range of activities across the day, such as making a model from old boxes of the robot from *Unplugged*, as well as reading a variety of books in the book corner. In the outside area, children made treasure maps to direct a friend to a hidden toy, which also helped develop their computational thinking skills. Pupils also worked together to build a den in the outside area.

For me, I enjoyed not using technology for the day, including my mobile phone over lunch! The main challenge I found was not using the interactive whiteboard during whole-class sessions, so I had to plan things differently, such as using real objects for mathematics inputs.

I also wanted to tie the day in with sustainability issues and trying to reduce electricity consumption. I planned to have our digital-free day just before Earth Hour, which is held on the last Saturday in March. During this annual event organised by

You can brainstorm activities to do away from devices, such as playing with construction toys

© DenisProduction.com/stock.adobe.com

the World Wide Fund for Nature, people around the world are encouraged to not use electricity for an hour, to show their commitment to sustainability. During our digital detox, we discussed Earth Hour as a class and talked about how we were also saving electricity.

At the end of the day, we discussed the digital-free day. Overall, the children enjoyed the day, but a few had missed using technology to support their learning. However, we all agreed that we could still learn and have fun without the internet.

A whole-school event

While I only did my digital detox day with a single class, you could try and do it across your school. Many of the things I did with my reception class could potentially work with other year groups as well. Here are some thoughts on what I could do next time:

- The idea of a digital-free day could be shared with students in a whole-school assembly, exploring why it is beneficial to have a healthy relationship with technology. You could discuss how pupils currently use technology and whether there is a need to reduce how much they use it.
- Everybody in the school could be invited to join in, including the head teacher and those working in the school office. While it might not be possible for all staff to be offline all day, they could reduce how much they use computers. I am sure they will appreciate the break!
- Remember that computing lessons can still carry on! Instead of using technology for computer science lessons, the focus could move to unplugged activities to help students further develop those computational thinking skills. The Barefoot Computing website is jam-packed with activities away from a screen (**barefootcomputing.org**).
- Students could be challenged to carry on having a digital-free day once they return home. Can you tempt your pupils to have an evening free from online gaming and social media? Parents will surely enjoy their children having some internet-free time.
- Linked to this, it might be useful to let parents know about the digital detox day in advance, so they can have conversations at home about how to have a healthy relationship with technology.

Hopefully some of these suggestions have whetted your appetite to have a digital detox day with your class or in your school! (HW)

JOHN PARKIN

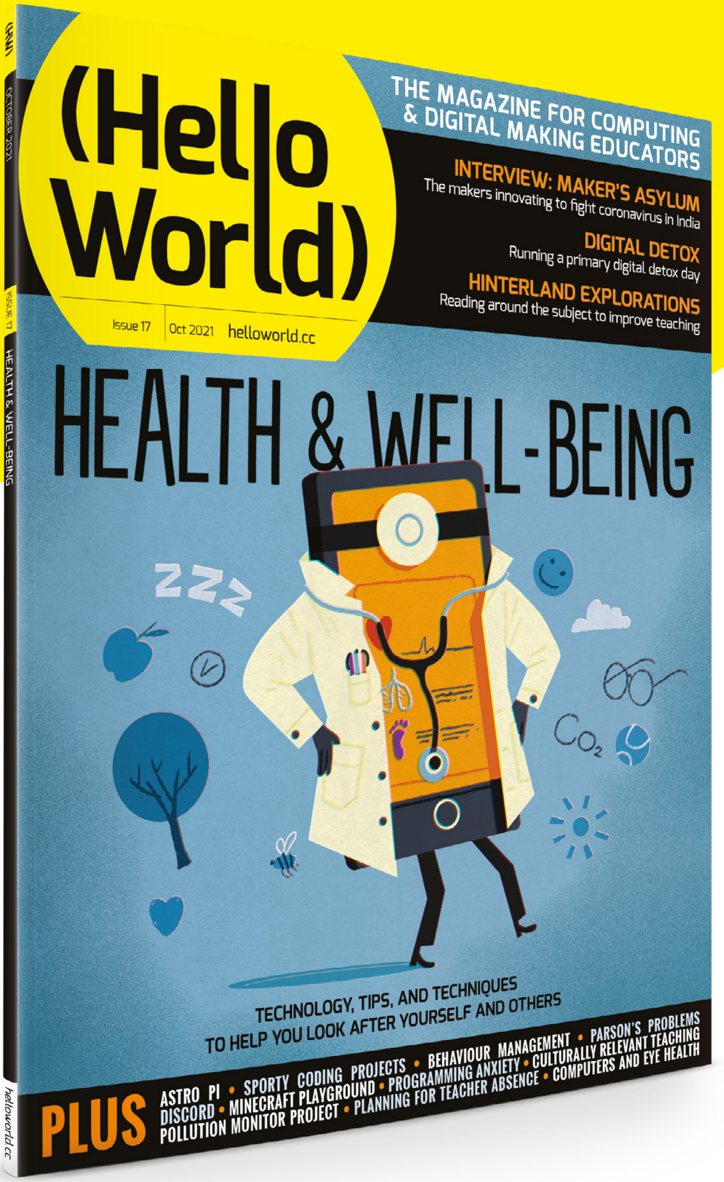
John is a senior lecturer in education and course leader for the BA Accelerated Primary Education Studies course at Anglia Ruskin University in Cambridge, UK. Previously, he was a primary teacher for 17 years, with most of his career as a reception class teacher and computing leader. He is also a Raspberry Pi Certified Educator.

(HW)

SUBSCRIBE TODAY

FREE IN PRINT

for UK-based educators


(Hello World)
THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS
INTERVIEW: MAKER'S ASYLUM
The makers innovating to fight coronavirus in India
DIGITAL DETOX
Running a primary digital detox day
HINTERLAND EXPLORATIONS
Reading around the subject to improve teaching
Issue 17
Oct 2021
helloworld.cc
HEALTH & WELL-BEING
TECHNOLOGY, TIPS, AND TECHNIQUES TO HELP YOU LOOK AFTER YOURSELF AND OTHERS
PLUS
ASTRO PI • SPORTY CODING PROJECTS • BEHAVIOUR MANAGEMENT • PARSON'S PROBLEMS
DISCORD • MINECRAFT PLAYGROUND • PROGRAMMING ANXIETY • CULTURALLY RELEVANT TEACHING
POLLUTION MONITOR PROJECT • PLANNING FOR TEACHER ABSENCE • COMPUTERS AND EYE HEALTH
OCTOBER 2021
ISSUE 17
HEALTH & WELL-BEING
helloworld.cc

Why subscribe?

- Teaching resources and ideas used by over 90 percent of our readers
- Exclusive news, research findings, and in-depth features
- Delivered to your door three times a year

FEATURE

BUILDING A GLOBAL EDUCATOR FAMILY

Cat Lamin talks about how building connections and sharing the burden can help make us better educators, even in times of great stress

FREDERICK BALLEW, MINNESOTA, USA

PILY HERNANDEZ, MONTERREY, MEXICO

ABID PATEL, LONDON, UK

CAT LAMIN

IN TIMES OF PRESSURE WE NEED A CHANCE TO TALK, AND IN THE PANDEMIC WE FOUND WE HAD LOST THAT OUTLET

TO SUBSCRIBE VISIT:

helloworld.cc/subscribe

THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS
(Hello World)
INTERVIEW: MAKER'S ASYLUM
The makers innovating to fight coronavirus in India
DIGITAL DETOX
Running a primary digital detox day
HINTERLAND EXPLORATIONS
Reading around the subject to improve teaching
Issue 17 | Oct 2021 | helloworld.cc
HEALTH & WELL-BEING
TECHNOLOGY, TIPS, AND TECHNIQUES TO HELP YOU LOOK AFTER YOURSELF AND OTHERS
PLUS ASTRO PI • SPORTY CODING PROJECTS • BEHAVIOUR MANAGEMENT • DISCORD • MINECRAFT PLAYGROUND • PROGRAMMING ANXIETY • CULTURALLY POLLUTION MONITOR PROJECT • PLANNING FOR TEACHER ABSENCE • COMPUT

Not a UK-based educator?

- Subscribe to receive the free PDF on the day it is released
- Read features and news at **helloworld.cc**

FREE PDF

for anyone, anywhere

HANDS-ON LEARNING WITH LEGO

Dru Watts shares her love of LEGO® robotics and how she implemented the *FIRST*® LEGO League

Around eight years ago, a lower-secondary student mentioned in computing class that he had just got a programmable LEGO set. With a school open evening coming up, I encouraged him to bring it to the computing department so that we could see what it could do. We were impressed and ordered a LEGO MINDSTORMS® set for our computing club, which proved very popular. We then found out we could enter a competition to compete internationally with our LEGO robots — and so our *FIRST* LEGO League team was born!

The *FIRST* LEGO League is an annual STEM competition that runs worldwide for students aged 4 to 16. There are three divisions: Discover for ages 4 to 6, Explore for ages 6 to 9, and Challenge for ages 9 to 16. Each year, the competition has a different theme, from cities, to water, to animals. As well as building a robot, there is an innovation project to complete, challenging students to solve a real-world problem based on the competition's theme. For example, our team has previously created an online game for saving water. Competitors are judged on their team's core values and robot design skills. There are awards at a regional, national, and international level for these individual sections, as well as overall tournament champions.

DRU WATTS

Dru is a computer science teacher from Suffolk in the UK. She is a Google Certified Innovator, Trainer, and Educator, and an RPi Certified Educator. She has a particular interest in getting girls and other underrepresented groups interested in computing (**@DruWatts**, **@TeamBergholt**).

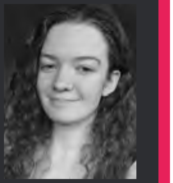
How it works

Our team is a voluntary after-school club, usually meeting twice a week. During competition time, students spend every extra minute working on their robot and projects. In the off season, we still meet regularly to practise, sort equipment, and take part in other activities and competitions.

There is usually a high demand to join, so when we have been oversubscribed, we have created additional training teams that enter other robotics competitions before a place in the *FIRST* LEGO League team is available.

There are various ways to organise a team, such as by year group, by gender, or by school class. Our team has evolved over the years, and it is now a mixed-gender group of students from different year groups. This makes it unique among our school clubs, and students say this is one of the things they love best about being in the team — they even socialise outside of the league, something they said they would never have done otherwise! Another benefit of having a mixture of year groups is that we're never starting the club from scratch: new team members are always introduced and trained by existing team members when the oldest finish secondary school — so the team virtually runs itself now!

WHAT OUR ALUMNI SAY: HARRIET



Harriet joined Dru's team when she was aged twelve and is now a computer science undergraduate at the University of Southampton, UK.

Why did you join the club?

I joined the club because I'd previously enjoyed experimenting with coding games in Scratch, and thought this seemed like something else I could do where you could actually see the results of your coding in real life, in the form of a robot.

What did you enjoy about the club?

I really enjoyed the camaraderie of being in a team: going to competitions together and all being really excited to test our robot against the other teams, and having a good time whether we did or didn't do so well!

The benefits of taking part

Students benefit from taking part in the competition in several ways. The league is unique in the way it encourages students to get involved in every aspect of a project, rather than requiring them to have a specific role. All participants

Students Harry and Annika with team alumni and mentor Morgan Levison

> THE FIRST LEGO LEAGUE IS UNIQUE IN THAT IT ENCOURAGES STUDENTS TO GET INVOLVED IN EVERY ASPECT OF A PROJECT

need to code, engineer, present to judges, and be gracious with other teams when competing — sometimes all at the same time — and our students have really risen to these demands when under pressure. You get to see your students flourish, and to observe the quietest students finding their voice and confidently talking about the ideas and passions that they have discovered and created.

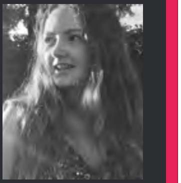
I am passionate about getting girls into computing, and the *FIRST* LEGO League is a great way to do this. The club has definitely been the most popular club for girls in our school, and I think the variety of events within the competition, as well as the teamwork and social elements, really appeal.

We have also taken part in research investigating the impact of the *FIRST* LEGO League programme. This research found that in our school, 100 percent of our team felt that they had improved their STEM skills and knowledge, compared to 77 percent of non-team members. Furthermore, 100 percent of our team reported that they felt they had learnt more about STEM careers, compared to 53 percent of non-team members. This is certainly reflected around the team's HQ in school at competition time. As a school governor once said, "Computing is **the** place to be."

How you can get involved

You can register to attend a regional tournament, or ask for a class pack so you can run your own tournament in school. If you would like to find out more about any of the *FIRST* LEGO League divisions or sign up for the next season, you can visit **helloworld.cc/legoleague**. (HW)

WHAT OUR ALUMNI SAY: CHARLOTTE



Charlotte also joined the team aged twelve, and is now in the second year of her maths degree at the University of Warwick, UK.

Has taking part influenced your career choices or given you skills you still use?

The club helped me realise that working with technology can be fun. The problem-solving in the competitions probably helped me realise that I enjoy logical thinking, leading to me taking mostly STEM A level subjects and then deciding to study a STEM degree subject.

Would you recommend that other schools start a team?

Yes, because it helps get people interested in STEM, which we need for the future. I also really enjoyed everything about it and being part of the team.

©CodeYourFuture

CYF trainees in class

FIGHTING INEQUALITY ONE LINE OF CODE AT A TIME

Harnessing the talent and passion of people from disadvantaged and refugee backgrounds to create a more diverse tech workforce, CodeYourFuture should be on everyone's radar

Triggered by the refugee crisis of 2015, Germán Bencci founded CodeYourFuture (CYF) in 2016 to train refugees and disadvantaged people to become web developers and support them with finding work in the tech industry. Drawing upon his own experience as an immigrant who has a successful career in tech, he saw a route to deliver positive futures for refugees while simultaneously addressing the dire need for more diverse talent across the industry. With the tech industry also having the lowest barrier to entry of any professional career — as a degree is not required — CYF seemed to be a no-brainer.

CYF now has hubs across the UK, as well as in Italy and South Africa. It is led by a small team and works with a vast and diverse network of volunteers. The focus now extends beyond refugees to reach people from all underrepresented backgrounds, including the long-term unemployed, ex-offenders, and women. CYF provides them with a warm and welcoming environment, delivering rigorous free courses in programming and employability.

The courses

There are three courses available, and coding is the base on which everything is built. The vast majority of trainees embark on the CYF programme with no previous programming experience. It really starts at the beginning, and there are no shortcuts. Participants begin with an introduction to coding, a self-paced programme with online support and workshops, in which they learn the basics of coding, build web projects, and discover relevant opportunities in the local tech industry. They can then move onto a fundamentals course, learning about the building blocks of coding and soft skills over a month. Finally, trainees can participate in an eight-month programme, learning everything they need to get a job as a full-stack web developer. It teaches a technology stack to build anything from a simple website to a powerful database-driven web application, and covers HTML, CSS, JavaScript, and front- and back-end development. Strong programming practices such as agile development, pair programming, and collaboration are introduced from day one.

The course syllabuses are a huge asset, and they are constantly evolving. The CYF team of volunteers across the country collaborates to update and improve them as the programme grows. The content is freely available for anyone to use as a teaching and learning resource, and we welcome feedback from educators and industry professionals.

Class cohorts currently run to 30 trainees, and each class has around three mentors at any time. There are career mentors to help trainees prepare for the working world, personal development mentors to help them improve their confidence and other soft skills, and a technical mentor for the coding. Since its founding, 165 people have graduated from CYF, and many have since gone on to work at the BBC, Capgemini, and the UK charity Comic Relief. The result of CYF's work, though, has been much more than creating a new pool of software developers.

The power of community

Listening to volunteers and trainees, the power of the community shines through. It is a welcoming and inclusive environment where everyone is always learning, and 'building confidence' is a key phrase that's repeated by both volunteers and students. Volunteers and graduates emerge having learnt something new, made new friends, and made a positive change in their lives. What's more, graduates of CYF often come back to be teachers later on.

In the words of CYF graduate Madiha, who went from juggling many unfulfilling jobs to a career with Capgemini, "CYF was a boost of confidence for many of us. During my time at CYF, where I now volunteer, one of the most important things I learnt is that there will be times when life will bring you down, but you just need to [get] back up."

Ahmed, a Syrian refugee, was on the very first CYF programme. He had been accepted to study economics at university, but civil war forced him to flee his home country and seek sanctuary in the UK. He spent the next few years simply surviving and unable to follow his ambitions. He applied to CYF in August 2016, stating, "I don't see myself better at anything except being a developer." He completed the course in six months and within a few weeks of graduation was recruited by tech start-up We Got POP as a junior developer. Ahmed was the first person to complete the CYF circle, finishing the course and getting a job.

With so many incredible people to meet and so many ways to get involved, volunteers also gain huge benefits from the programme. Beth, a volunteer on the personal development team in London, shared: "Volunteering with CYF has massively increased my confidence with both public speaking and leadership skills." Volunteers can develop IT and presentation skills, gain mentoring and leadership experience, and learn to be a teacher.

With such innovative and dynamic programmes for both volunteers and trainees, CYF is a great place to work or learn, and is helping to make a big difference to society: fighting inequality, one line of code at a time. (HW)

©CodeYourFuture

GET INVOLVED

Around 60 volunteers are involved per cohort, and there is always demand for more volunteers. Whether you want to teach code, be an ambassador, provide personal development and mentoring, lead on tech projects, or advise on the CYF syllabus, there is a place for you.

- Check out volunteer experiences here: **codeyourfuture.io/blog**
- Find out how to get involved here: **codeyourfuture.io/volunteers**
- Follow **@CodeYourFuture** on Twitter/ **@codeyourfuture_** on Instagram

ANNA DOYLE

Anna is the liaison officer for the Software Engineering Graduate Apprenticeship at the University of Glasgow in Scotland. She is a board member for the Ada Scotland Festival and a volunteer at CodeYourFuture. Anna also sits on the BIMA Scotland Council (**@UofGGAs**).

CYF TRAINEES IN NUMBERS

- 91 percent have no prior coding experience
- 78 percent are from minority ethnic backgrounds
- 40 percent are female
- 75 percent are from households below the UK poverty line

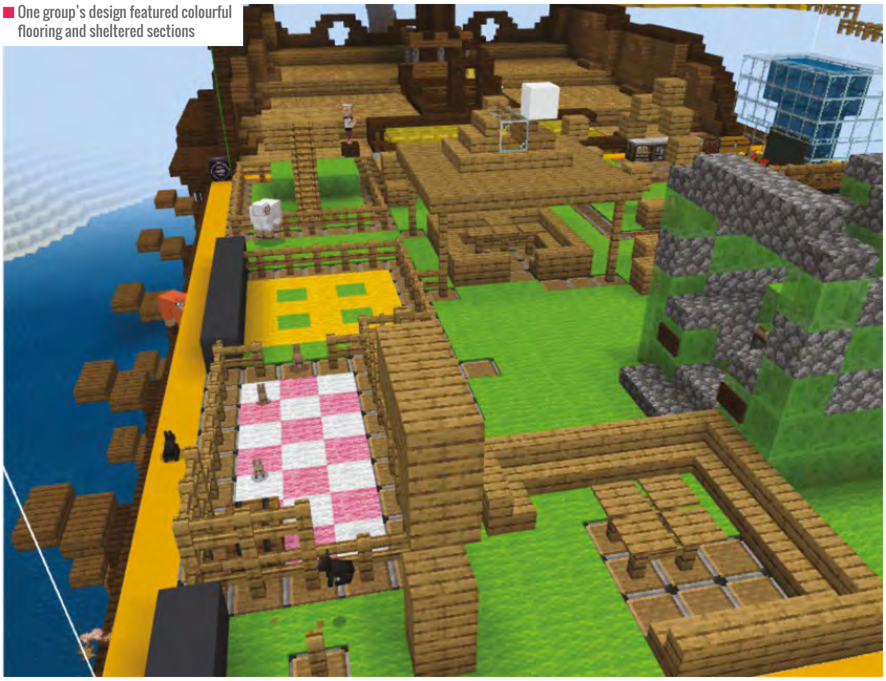
USING MINECRAFT TO DESIGN A NEW SCHOOL PLAYGROUND

Siobhán Morgan shares how her school's pupils used Minecraft to help design their new school playground

When they returned from the UK's winter lockdown in March, many of our pupils had not seen each other since before Christmas. An esports challenge to solve a real-life problem provided an opportunity for pupils to experience something new and work collaboratively, while remaining safely in their bubbles (small groups).

Our school was in the process of redeveloping the outdoor play areas for our junior school pupils. We needed areas that would be suitable for children between the ages of seven and eleven, so that they could enjoy playing outside with their friends — something many of them had lost out on over the past year. It was important to the school that we considered what our pupils wanted from this space, and so the pupil voice was a key part of the planning process.

We introduced pupils to the idea of their new playground in an assembly. They were able to share their ideas there, and we actively encouraged them to keep coming forward with more ideas. As all of our pupils are familiar with how to use Minecraft, it made perfect sense for them to share their design ideas through this medium. One of the benefits of Minecraft is how easy it is to use. Another is the ability for pupils to create designs using an inventory of blocks and items to visually represent their imagined playground. Minecraft doesn't require pupils to be able to draw, so everyone would be able to represent their ideas equally within the virtual world.



One group's design featured colourful flooring and sheltered sections

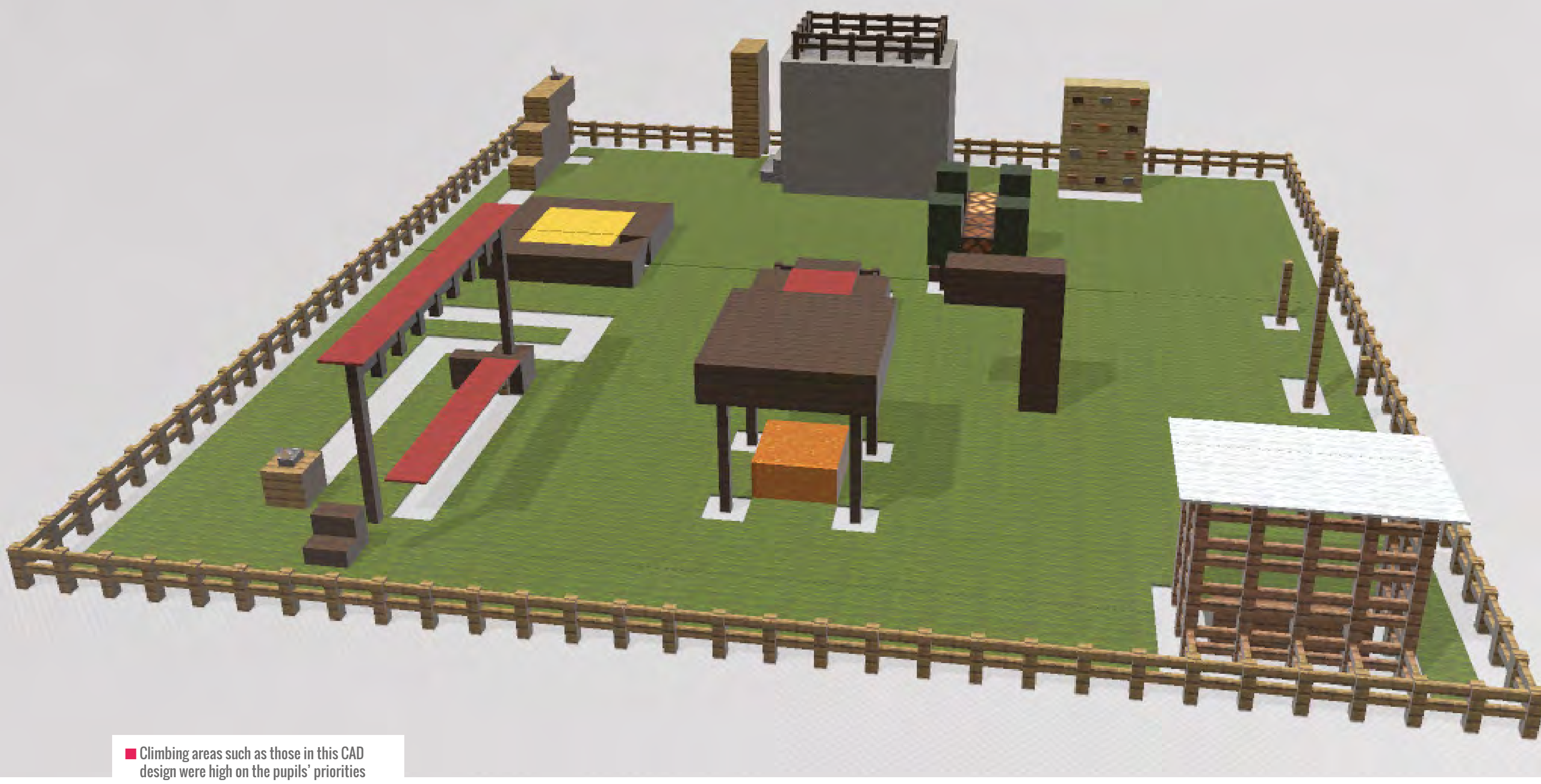
Running the project

Rather than asking all pupils to create their own playground design, which would have produced nearly 200 separate designs, we decided to set up an inter-class challenge. Each class was split into two teams, each of which would work together to create a playground design. To make this more fun, we decided to use one of the Minecraft esports worlds (**helloworld.cc/minecraftesports**) to do this through online competitive and collaborative gaming.

We selected the Pirate Cove option, which enabled two teams to build their designs on two giant pirate ships. This meant that each team had a clearly marked-out design area and couldn't accidentally encroach on the other team's design. Playing within the esports world, instead of a standard empty Minecraft world, meant that the activity was timed, and that each team was trapped within their game zone for its duration.

We ran the challenge for three weeks during tutor times at the start or end of each day. Each class arrived at the computer room in their time slot, knowing who was in their team. When the game began, the time available was displayed with a countdown on the giant sails of the pirate ship, adding extra excitement to the proceedings.

Framing the activity as an esports challenge was effective and generated a great deal of excitement among pupils.



Climbing areas such as those in this CAD design were high on the pupils' priorities

While all of them had previously used Minecraft in lessons and at lunchtime clubs, they had never used the esports worlds, so it was a learning curve for all involved. They had to learn to be patient: normally, in the virtual world, pupils quickly run off exploring. In the esports world, you have to stand still on your team's coloured zone to be teleported onto the ship. This was hard! We had to introduce 'no flying' and 'no potions' rules in the virtual world, and a 'hand on heads' rule in the real world, to avoid anyone being trapped outside of the gaming area due to overexcitement!

As the challenge took place, my avatar flew around the Pirate Cove with a camera and a portfolio. The camera allowed me to take photos of the progress of the pupils' designs, and the portfolio saved my photos in a virtual book that I could download as a PDF at the end of each session. When the timer ended, the pupils were able to visit the other team's pirate ship and explore their design.

We then got together as a class to choose which design we liked best. We chose not to use the built-in score-based voting feature in the esports worlds, and instead had class discussions in which we focused on what we liked from each of the designs. After all, it wasn't a competition, and our main focus was to gather pupil voice to support the decisions we needed to make about our new playground.

Another great feature of the esports worlds is that they are set up with a structure block that allows you to export the models instantly at the end of the session. This enables you to open the design in CAD software, so you can edit the designs further, or even use the files to produce 3D prints of the designs built in Minecraft.

What we learnt

Collecting pupils' opinions about the playground through Minecraft gave them the creativity to express their ideas more fully than may have been possible through other methods. We learnt that pupils wanted sheltered spaces and climbing features, and these are featured in the school's final playground design. With the timing of returning to school after a period of separation from friends, our focus was on enjoyment, rather than competition. The esports world was an effective tool to allow us to do this while gathering pupil voice.

I would be keen to introduce more Minecraft esports activities in school in the future, and they can be both competitive and collaborative. For example, one lower-primary class recently built Celtic roundhouses and farms in an esports world, linked to their history class. With esports proving popular in secondary schools, developing similar opportunities for meaningful learning through gaming for primary pupils seems too good an opportunity to miss. (HW)

SIOBHÁN MORGAN

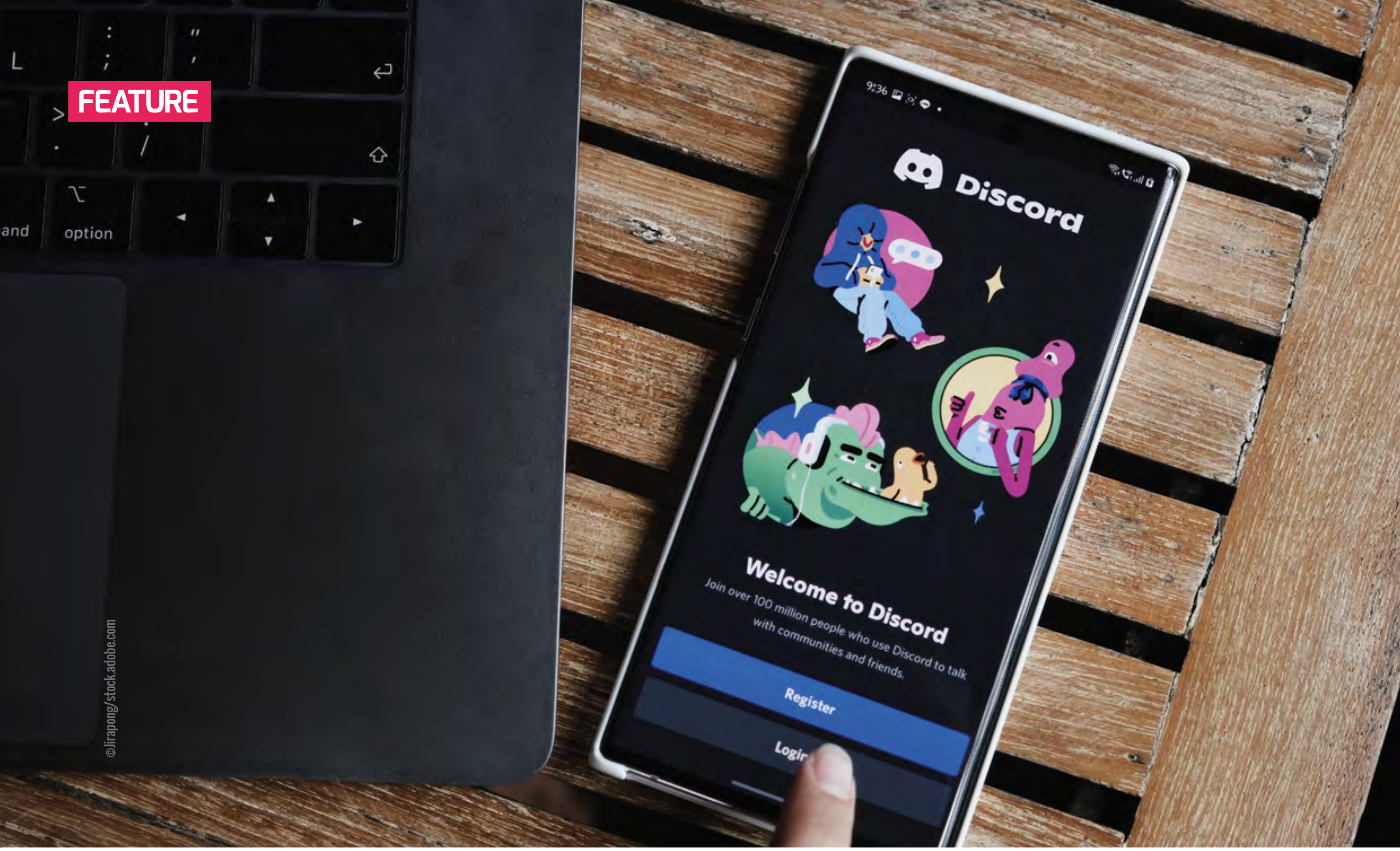
Siobhán leads computer science and design technology at Exeter Junior School in the UK and is completing a PhD in technology-enhanced learning. Her interests are programming with Kodu and Crumble, and implementing Minecraft across the curriculum (**@koduclassroom**).

TRY IT FOR YOURSELF!

Learn more about esports challenges at **helloworld.cc/minecraftesports**.

Take the Microsoft esports educator course at **helloworld.cc/esportscourse**.

Explore Exeter Junior School's designs at **helloworld.cc/exeterschool**.


Discord
Welcome to Discord
Join over 100 million people who use Discord to talk with communities and friends.
Register
Login
option

©Jirapong/stock.adobe.com

DISCORD: THE HUB OF COMMUNITY BUILDING

Sethi De Clercq explores how Discord has become a platform of choice and looks at its merits in the computing classroom

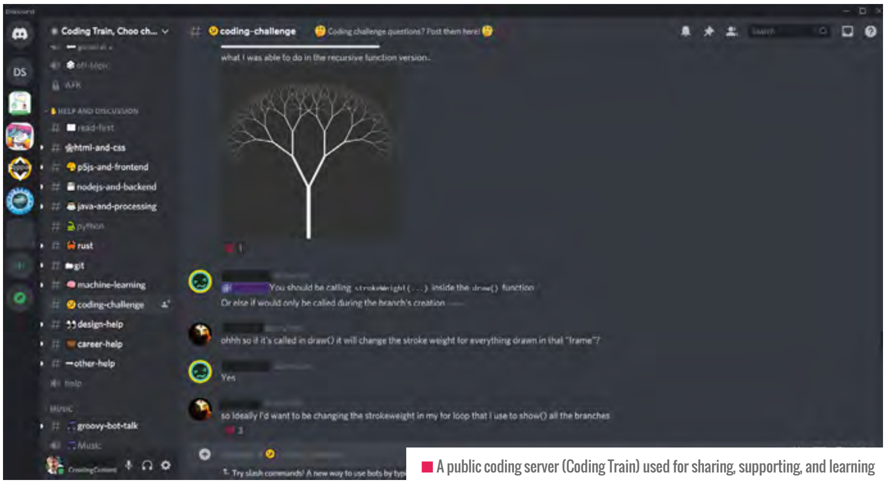
Effective, clear communication is key to meeting the goals you've set out with your students. Achieving that communication is an opportunity to develop new skills and grow your competencies as a leader, learner, teacher, friend, or colleague. Over the years, digital communication has taken many forms. Some platforms are strictly professional, while others have blurred the line between work and leisure. We've seen companies come and go, and seen the various tools we now take for granted, such as email services, message boards, forums, and IRC chat rooms, evolve and develop. Video conferencing has been added to the seemingly never-ending stream of emails and text messages. We communicate with strangers using the comment sections on social media, and leave feedback on documents and shared slideshows. Teachers and schools have quickly learnt the importance of timely and clear communication, and are quick to try out new ways of communicating subject matter, or of helping students become the best versions of themselves.

Our students, of course, have their own ways of communicating with each other. We've seen Snapchat rise in popularity, Instagram DMs become standard, and many other social media messengers become dominant forms of communication. Social audio apps such as Clubhouse and Twitter Spaces are exploding onto the scene, and Discord has quickly become a staple means of communication for the modern student.

What is Discord?

Discord (**discord.com**) is a platform offering a free voice over Internet Protocol (VoIP) service, and is designed to help you create your own online communities. It was originally popular in the world of gaming. Users communicate using voice calls, video calls, text, media, and files in private chats, or as part of these created communities, known as servers.

Over the past year, Discord has evolved into a place for creatives of all kinds. Students have created their own communities to help them connect during the various lockdowns, and large coding and design communities have emerged where students and coders connect, ask questions, share their solutions, and



what I was able to do in the recursive function version..

@ You should be calling `strokeWeight(...)` inside the `draw()` function
Or else it would only be called during the branch's creation

ohhh so if it's called in draw() it will change the stroke weight for everything drawn in that "frame"?

Yes

so Ideally I'd want to be changing the strokeweight in my for loop that I use to show() all the branches

Try slash commands! A new way to use bots by typ

A public coding server (Coding Train) used for sharing, supporting, and learning

“ DISCORD HAS GIVEN STUDENTS AND TEACHERS A PLATFORM THAT ENABLES EFFORTLESS COMMUNITY BUILDING

support those learning the craft. A recent search showed at least 80 public servers tagged with the term 'computer science', and that number is rising daily.

Discord has provided students and teachers with a platform that enables effortless community building and bridges the divide between online and in-person learning. Even before our students faced the prospect of distance learning, they were discussing lessons, homework tasks, and everyday experiences on this platform.

Does it have a place in education?

It might already. Discord's reach has risen from 56 million active monthly users in 2019, to 140 million in 2021. It's a platform our students know and love. As educators, we can leverage this to break down some of the hesitation our students may have about collaborating and supporting peers in an online community.

The server settings (remember, a server is a single community on Discord) are uniquely designed to allow collaboration through the roles and permissions that are given to your users. Servers can be completely private and work on an invite-only basis, or can be opened up to the wider community. Students can be given additional moderation roles or responsibilities within their own channels.

Discord has been adopted by many schools, colleges, and universities (over 200 at the time of writing) for all sorts of communities, including as after-school club programmes, and as a way for students to find study buddies.

What about computing?

Discord has become more and more popular in the gaming and developer communities because of a simple, yet powerful advantage it has over many other online community platforms: Discord bots. This feature has opened up worlds of possibilities for the computing classroom.

In Discord, bots are a staple of most communities, and moderate communication within the servers. Discord bots can help you do everything from automating mundane tasks to playing music across your server, integrating with other platforms such as YouTube and Twitter, or enforcing server rules and banning inappropriate language.

So what does this have to do with the computer science classroom? Well, anyone can write and deploy these Discord bots using Python and JavaScript. That's right! We've just given our more reluctant learners another reason to dive into the real-world application of these programming languages. They can even take these bots to their own servers, or make them available to others.

SETHI DE CLERCQ

Sethi is the subject leader for STEM and computer science at a British International School in Thailand. He leads professional development on the use of educational technology, and shares his experiences on YouTube (Flipped Classroom Tutorials) and Twitter (**@SethiDeClercq**).

I have seen some very innovative uses of the Discord platform by many amazing computer science teachers to run boot camps, hackathons, coding challenges, and more. And the biggest winners in all of them are the students! They get to use a platform they are familiar with, see real-life applications of what they build, and do it all on a platform that is so very 2021.

If you wanted to make a comparison with the business world, you'd be likely to spot some similarities with team-based productivity apps such as Slack. The enterprise world has long loved Slack, and there are three main reasons why so many use it as their application of choice: communication, collaboration, and notifications. It lets you chat with your team, share files, and, for more advanced users, write bots to answer questions and give help automatically. It helps you stay up to date with notifications, and to funnel your communication away from your email inbox.

All the reasons given above are exactly why our students love to use Discord. So why not give it a try and see if it adds a special spark to your lessons and class discussions, reaches that one student who seemed reluctant to ask questions, or helps you to engage with your students in a way that is part of their everyday communication? (HW)

RESEARCHING VIRTUAL PAIR PROGRAMMING IN A PANDEMIC

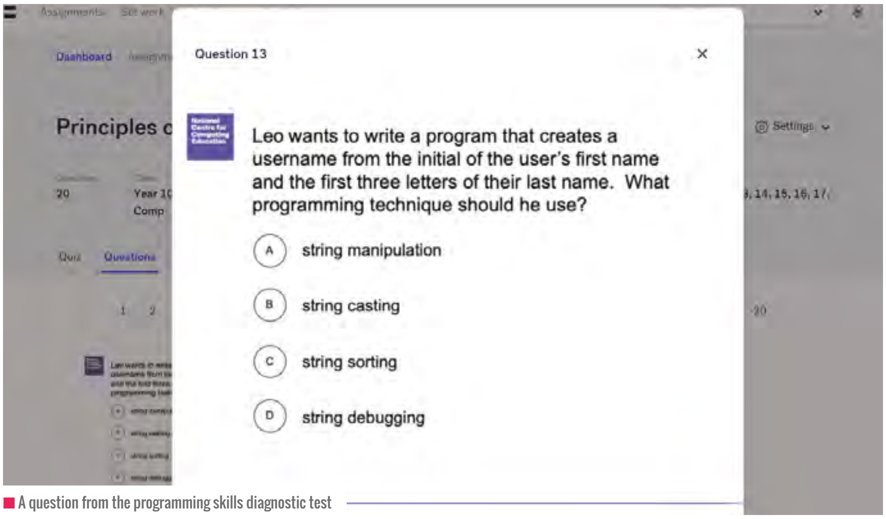
Phil Hackett shares his research into pair programming with GCSE students, exploring innovative technology, discovering different pairing types, and identifying hidden benefits

When I started my final-year research project as part of an MSc in computing, I wanted to investigate the benefits of pair programming and identify the best approach to adopt when using this technique. Pair programming usually involves two people working on the same computer, solving the same problem in driver and navigator roles. The driver controls the keyboard and mouse and writes the code, and the navigator focuses on the wider problem-solving aims of the task. Unfortunately, my risk assessment didn't consider that pupils would not be allowed to use the same computer due to a pandemic! This led to some innovative adaptations and interesting results.

Starting the research

I carried out the research over a six-week period with one class of students aged 13–14 and another class aged 14–15. Social distancing restrictions in the UK had relaxed enough to allow pupils to sit near each other, but they still could not use the same computer. This meant that I couldn't investigate traditional pair programming, and instead, I investigated through the lens of virtual pair programming. This involves two people working on their own computer, but using a shared integrated development environment (IDE). Both people can type at the same time, so the driver and navigator roles are blurred.

At the start of the research, I measured pupils' programming skills and programming confidence levels. I would measure these again at the end of the research, to understand the benefits of the pairing activities and the best combinations of pupil pairings. I assessed pupils' programming skills using Eedi's Diagnostic Questions website and England's National Centre for Computing Education's question sets 'Principles of programming', 'Design and development', and 'Algorithmic thinking'. I also carried out a survey of programming confidence levels using Google Forms.


Question 13

Leo wants to write a program that creates a username from the initial of the user's first name and the first three letters of their last name. What programming technique should he use?

- (A) string manipulation
- (B) string casting
- (C) string sorting
- (D) string debugging

A question from the programming skills diagnostic test

Each pupil's predicted grade in computer science also formed part of the initial data that helped to identify the different pairings. Pupils were categorised as having one of a range of abilities in programming, from low to very high, this sometimes differed from their predicted GCSE grade, so a higher-ability pupil may have a low programming ability based on the initial assessment. I allowed pupils to work solo, or with another person of their choice, each week. This provided lots of different combinations of pairs and solo programmers for the research.

Innovative technologies

Innovations in online technology accelerated while I was completing my master's. One such innovation was the multiplayer mode in the Replit IDE, which the older group of pupils used. This mode allows different people to work on the same program at the same time, from different locations. During the research, I would introduce a programming topic, and then pupils would complete the programming challenges on Replit, either as a pair or solo. Pairs could sit near each other so they could communicate face to face, as well as using the IDE's built-in comment and chat facilities.

In the younger group of pupils, pairs and solo programmers used **blog.withcode.uk**, which introduces a topic and provides a self-marking programming challenge. Those collaborating as a pair discussed a solution and completed the challenges on their own computers.

Pair programming can improve social and communication skills

> PAIR PROGRAMMING APPEARS TO OFFER PUPILS HIDDEN BENEFITS, SUCH AS IMPROVED SOCIAL AND COMMUNICATION SKILLS

Results from the research

It was evident that pair programming can have a positive impact on programming proficiency, although there are other factors affecting performance. One key indicator of performance is confidence in programming, which appears to have a bigger impact on performance than predicted grades, particularly for pupils with less confidence.

The results show that solo programmers performed better in the weekly challenges than pairs did, both in terms of number of activities completed and percentage score on the activities completed successfully. Bear in mind that the solo programmers included more confident pupils. Furthermore, the use of a virtual — rather than traditional — pair programming technique, with the driver and navigator roles blurred, may explain the drop in speed for pairs.

The pair combinations used did not appear to have any impact on the outcome of the weekly activities. In some weeks, a low-ability pairing performed the same as a medium-ability pairing. Also, in the final practical activity, a solo low-ability pupil outperformed a pairing of pupils with medium and very high abilities. What was clear from the research, though, is that pairing pupils with lower confidence did not improve their final assessment score or confidence level. Pupils who were able to experience a mixture of solo and pair programming did, however, see an improvement in their confidence and achieved a higher score in the final assessment.

Recommendations

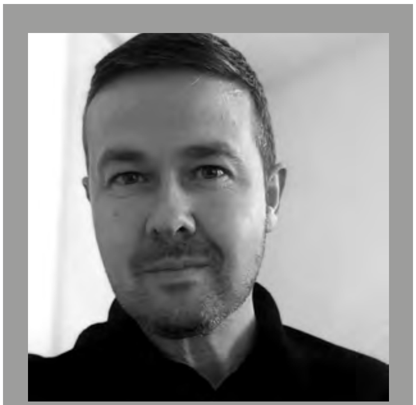
There are several factors to consider when using pair programming in the classroom, including the pairing of pupils and the hidden benefits. If using pair programming, there should also be an opportunity for solo programming; pupils should avoid becoming overly reliant on others. The pairing of pupils should not necessarily be based on predicted grades or programming proficiency alone. Social and communication factors are also important. My research recommends that teachers measure their pupils' programming confidence to avoid pairing low-confidence pupils together.

LINKS TO RESOURCES

- Assessment platform for computing: **diagnosticquestions.com**
- Self-marking programming challenges: **blog.withcode.uk**
- Collaborative IDE for many languages: **replit.com**

As a teaching technique, pair programming appears to offer some hidden benefits, such as improved social and communication skills. Although pairs took longer than solo programmers to complete every activity, speed is not necessarily a measure of success, and if the process of finding a solution through collaboration takes longer, that is perfectly fine.

Through this research, I recognise that the focus of programming lessons should be on confidence-building activities. Activities that are too difficult can lead to pupils giving up. When pupils did complete the challenging but achievable activities during this research, I could hear their excitement and sense of achievement. I heard lots of "I did it!" and "Yes, it's working!", and this is surely what a computing education is all about. (HW)



PHIL HACKETT

Phil is a staff tutor and associate lecturer at the Open University in the UK, as well as an NCCE CPD facilitator and an RPi Certified Educator. Phil has over ten years' experience of teaching computer science to students aged 11-18, and ten years' experience as a network manager and IT consultant (**@missedabracket**).

THE GAMEFUL CLASSROOM

What ideas can be borrowed from the world of game design as we plan units and assessments?

Gameful learning is a pedagogical approach that identifies the design decisions that drive people to complete good games, and applies that to the design of learning environments. By making some changes to assessment structures, gameful learning aims to build more motivating classroom experiences.

What does gameful mean?

The first thing to clarify is what gameful education is and is not, as there are three similarly named areas of research which can overlap to some degree: game-based learning, gamification, and gameful learning.

Game-based learning is the process of having students play purpose-made educational games in order to learn new skills or concepts (see Neil Rickus' article in Hello World, issue 16). An example of game-based learning could be having students build chemical compounds in Minecraft: Education Edition. By contrast, gamification is the idea of borrowing simple elements from games, such as leader boards, avatars, and achievement badges, to motivate users to complete a task. Platforms like Quizlet Live include elements such as teammates and live performance graphs to make a standard multiple-choice quiz more engaging for students.

Gameful education aims to make the entire classroom experience itself more like a game. Like gamification, it also takes inspiration from game design elements as tools for learning, but goes one step further by operating within a self-deterministic framework. The key to building a gameful classroom is combining a handful of elements from game design with the recommendations self-determination theory makes about increasing levels of intrinsic motivation, such as:

Autonomy: Allowing students to make meaningful choices about what they do in the classroom
Competency: Providing students with an adequate level of challenge, but one at which they feel they can succeed
Belonging: Ensuring that students feel connected to those around them

Self-determination theory has been around for a few decades now, but it was not until much more recently that the ideas of autonomy, competence, and relatedness have been more thoroughly examined in an educational context. In the last five years, with support from Microsoft Education, the University of Michigan has been championing gameful pedagogy at their Center for Research on Learning and Teaching. We will now run through the principles they have developed for running a gameful classroom.

The key ideas

1. Build up points from zero: When video game players start a new game, they are expected to start with a score of zero and build up points as they progress. This exact same idea is applied in a gameful classroom: a student starts with zero points, and then each piece of work they submit can be assigned a number of points which helps to build up their total score. Scoring periods can be separated by unit or term, or even continue for an entire academic year.

2. Offer student choice: In a game, there are usually multiple routes to success. A gameful classroom similarly offers students a small amount of choice over their learning environment, helping them to take ownership of their learning. For example, if a student is expected to produce a portfolio of work worth 1,000 points in total to receive an A grade, the teacher can set 500 points worth of assignments everyone must complete. For the remaining 500 points, students can then select from a menu of quizzes, presentations, blog posts, group projects, etc.

USEFUL RESOURCES

- **gamefulpedagogy.com:** Learn more about the key principles of gameful pedagogy
- **helloworld.cc/gamefulcourse:** A free online course from the University of Michigan and Microsoft Education to help support gameful learning in schools

© Highwaystarz/stock.adobe.com

Gameful education identifies the design decisions driving people to complete games and applies them to an educational context

> THE AIM OF GAMEFUL EDUCATION IS TO MAKE THE ENTIRE CLASSROOM EXPERIENCE FEEL MORE LIKE A GAME

3. Provide immediate feedback: Something that makes games engaging is the immediate feedback given in response to a player's actions. Once a student completes a task, a teacher in a gameful classroom ensures that they receive fast and thorough feedback wherever possible. Suitable methods include providing students with rubrics, auto-grading, peer feedback, or having students complete reflections.

4. Allow freedom to fail: When something goes wrong in a video game, it is not the end of the world. Players can restart levels, experiment with what's possible, and try new approaches until they are ultimately successful. The same is true in gameful education. Students should be encouraged to try things outside their comfort zone, which will ultimately expand their skill set. Failure should be seen as a learning opportunity for a student and a low grade in an assignment should not seriously damage a student's final grade. Ways to incorporate this concept could include allowing students to resubmit an assignment multiple times, or creating multiple pathways of assessments to complete a unit within a course.

5. Make the course requirements transparent: As gameful education is designed around student choice and agency, it is important that all available assignments are defined clearly at the start of a term or unit. This will help students to plan what assignments they will complete to achieve their desired results. Moreover, the grading criteria should be clear, so students know exactly what is expected of them to succeed, and teachers should provide actionable feedback if they are not performing as they should be.

Practical applications

Personally, I've found that this approach can be applied effectively in both theoretical and practical topics throughout the computer science syllabus that I teach. For example, when working through a unit about computational thinking and problem-solving with my A-level students, I used a number of gameful techniques when preparing the scheme of work.

For instance, there were a few core assignments that all students were

GAME DESIGN FEATURES

Here are ten features of well-designed games to apply in your classroom:

- Clear learning goals
- Identity play (for example, have students pretend to be members of an R&D department)
- Embedded assessment
- Intrinsic and extrinsic motivators
- Supporting student autonomy
- Encouraging belonging and community
- Supporting competence in a new skill
- Making failures productive
- Encouraging exploration
- Practice and reinforcement

© Studio13lights/stock.adobe.com

Gameful education identifies the design decisions driving people to complete games and applies them to an educational context

> I SAW AN INCREASED LEVEL OF LEARNER RESILIENCE, WITH STUDENTS WILLING TO STICK WITH A CHALLENGE FOR LONGER

expected to complete, such as programming tasks and an end-of-unit written test. However, this only made up a percentage of their classwork grade for the unit. The remaining proportion of their grade was based on the submission of a couple of additional pieces of work that students could select from a menu of tasks to further demonstrate their learning. The optional assignments included tasks such as a group video project in which students explained an algorithm in depth; writing a blog post that explained how to implement a data structure in their given programming language; or even providing evidence of completing all gameboards related to a topic on the Isaac Computer Science platform.

With regard to allowing freedom for failure, I tried to implement this in the core programming assignments. Before the students had written a single line of code, I provided them all with access to the unit test files I would be using to grade their work through GitHub Education. Students were allowed to submit their work an unlimited number of times before the deadline, and for each submission of their work, they would be sent an automatic response within minutes. The message would explain which tests they were currently passing and where they still needed to improve. During the submission period, I was able to watch in real time as students incrementally went from scoring 0/10 in their tests, to 2/10, to 5/10, and to 7/10. They were learning with each new iteration of their project without being penalised for missing edge cases and off-by-one errors at the first attempt.

Reflections

While I have not transitioned to a 100 percent gameful learning environment, I have made a number of personal observations after trialling the approach. Allowing my students to self-select certain assignments has led to better-quality submissions, due to each learner's greater sense of ownership of their work. After moving to unlimited attempts for the programming tasks, I saw an increased level of learner resilience; they were willing to stick with a challenge for much longer periods of time. Moreover, during these programming tasks, I observed the organic creation of mini code-review groups: students were happy to explain their design approach when completing a certain task, to help their classmates get on the right track.

In the coming academic year, I aim to trial gameful aspects with some lower-secondary classes. I would like to investigate whether my younger students are able to handle having a higher level of responsibility for their learning, due to the self-selected submissions. Watch this space! (HW)

JARED RIGBY

Jared is a learning technology coach and computer science teacher working at Dulwich College Beijing. He is a Microsoft Innovative Educator Expert and was the first GitHub campus expert based in mainland China (**@jazibobs**).

RICHMOND MATHEWSON EFL SCHOOL DIRECTOR

LIVECODE: THE IDEAL STARTER LANGUAGE

Richmond Mathewson explains why LiveCode is the ideal starter programming language and an excellent alternative to block-based programming

In the 1980s, it became clear to a number of people that computers would not be a niche product, and that programming languages such as BBC BASIC and FORTRAN were too complex for ordinary users to learn quickly. Out of that realisation came HyperCard, the first hypermedia programming environment, predating the World Wide Web. While HyperCard was confined to Macintosh computers, it still suddenly empowered a considerable number of people who were able to build bespoke software to serve their needs. LiveCode is its direct descendent, with considerably extended possibilities and the ability to run on Windows, macOS, and Linux.

An ideal starter language

While traditional programming languages present the programmer with a bewildering page of unlikely-looking text, LiveCode offers programmers a WYSIWYG (What You See Is What You Get) interface. This type of interface means that code is delivered not as a continuous list, but as discrete snippets in constituent predefined objects. Unlike in other languages, compiling code in LiveCode is performed on the fly, meaning you can run and edit programs constantly and simultaneously. The software can be repeatedly run and edited with minimum fuss, saving a lot of time vis-à-vis the traditional run, edit, compile, and debug cycle of other languages, and everything is almost entirely transparent.

As LiveCode is presented on a computer embedded in its own integrated development environment (IDE), coding and results can be seen almost simultaneously. This contrasts sharply with both block-based environments such as Scratch and more traditional languages such as Python. In Scratch, the underlying code is not exposed and is written using the block metaphor. In Python, while the underlying code is exposed, the look and feel of the result is hidden. As most of the children I teach are under eleven years old, they have not reached Piaget's formal operational stage and, as such, find it extremely difficult to think logically about things without the presence of a physical model (as with Scratch). The advantage of LiveCode is that the physical mode and the code can be present on-screen at the same time as its WYSIWYG interface, offering something that neither Scratch nor Python (or other traditional text-based languages) can.

LiveCode is the ideal starter language: it's continuously being developed, easy to read, quicker to edit and run, and gives young students the opportunity to use a real-world language used by corporations. Why not give it a go? (HW)

GET STARTED WITH LIVECODE

LiveCode provides an extensive number of step-by-step lessons, code samples, cheat sheets, beginner's guides, and learning courses: **helloworld.cc/livecoderesources**

Richmond maintains Facebook pages with slightly simpler support resources: **helloworld.cc/livecodesupport1** and **helloworld.cc/livecodesupport2**

RICHMOND MATHEWSON
Richmond has been running a private English language and programming school in Bulgaria for 18 years. He has previously taught at all levels of education from primary schools through to university. He holds a BA from Durham, an MA from SIU Carbondale, and an MSc from Abertay, Dundee.

COMPUTING FOR ALL: DESIGNING A CULTURALLY RELEVANT CURRICULUM

Researchers at the Raspberry Pi Foundation present guidelines they have developed to support computing teachers in delivering a culturally relevant curriculum

Computing is often seen as a neutral subject in which there are clear rules and programs to follow, with the person behind the code having little influence. However, the person's cultural identity, which is made up of a number of factors such as their age, gender, where they live, their family income, and their religious beliefs, all affect the way they think about and understand the world. This has an impact on decisions they make during the design and development of computing systems. Having limited diversity among computer scientists can lead to bias when making these decisions, towards what is valued or understood by the dominant groups in society.

Making computing culturally relevant means that learners with a range of cultural identities will be able to identify with the examples chosen to illustrate different concepts, engage effectively with the teaching methods used, and feel empowered to use computing to address problems that are meaningful to them and their communities. This may help a more diverse group of learners feel that they belong in computing and choose to continue with it academically and professionally.

In the USA, different frameworks have already been developed to demonstrate how education can be made more equitable and relevant for diverse learners. The first, Ladson-Billings' culturally relevant pedagogy (**helloworld.cc/crp**), emphasises the importance of incorporating and valuing all learners' knowledge, ways of learning, and heritage. The second, Gay's culturally responsive teaching (**helloworld.cc/crt**), builds on the previous framework to identify a range of teaching practices that can be implemented in the classroom, including drawing on learners' personal experiences and cultural identities, and encouraging them to choose projects that are meaningful to them.

At the Raspberry Pi Foundation, we have drawn on these frameworks to develop a set of guidelines for UK-based computing teachers, to support the integration of more culturally relevant and responsive approaches into their practice (**helloworld.cc/crpguidelines**). This article discusses how these guidelines were created, how they can be used, and how teachers can develop their knowledge further.

KEY TERMS

Term	Definition
Culture	A person's knowledge, beliefs, and understanding of the world. It is affected by multiple personal characteristics, as well as social and economic factors.
Culturally relevant pedagogy	A framework for teaching that emphasises the importance of incorporating and valuing all learners' knowledge, ways of learning and heritage, and promotes critical consciousness in teachers and learners.
Culturally responsive teaching	A range of teaching practices that draw on learners' personal experiences and cultural identities to make learning more relevant to them, and support the development of critical consciousness.

Culturally relevant teaching may help a more diverse group of learners feel that they belong in computing

© jovannig/stock.adobe.com

> OUR AIM WAS TO BROADEN PARTICIPATION IN COMPUTING AND ADDRESS THE NEEDS OF A DIVERSE SET OF LEARNERS

Developing the guidelines

In 2021, we were awarded an ACM SIGCSE Special Project Grant for a project called 'Developing criteria for K-12 learning resources in computer science that challenge stereotypes and promote diversity'. Our overarching aim for this project, as with all our work at the Raspberry Pi Foundation, was to broaden participation in computing and address the needs of diverse learners.

We started by recruiting a working group of teachers and researchers to work with us to develop the guidelines. We invited four advisors to take part, one from the USA, two from the UK, and one from Canada (see 'Our advisors' box) and we put out an open call for UK teachers to apply to join the working group. After assessing all the applications, we selected seven teachers to join the group.

The group then worked together over the course of three months to develop the guidelines. The work involved two meetings and an ongoing process of discussion and collaboration as we gradually iterated the guidelines from an initial set of draft criteria into the final product.

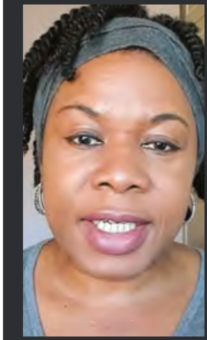
At the first meeting, we ran a series of whiteboard activities and both small-group and whole-group discussions. It quickly became clear that the focus of our initial draft criteria was too narrowly focused on learning materials, with the teachers in the group in particular identifying the need for a broader perspective on curriculum and pedagogy.

The guidelines were therefore split into three main focus areas, which we likened to the structure of a tree: the curriculum forms the roots of the approach, and the branches represent a number of different teaching approaches you can take to deliver the curriculum. The leaves represent the learning materials you use in your computing lessons. Beginning with the curriculum and working your way up will give you the strongest basis from which to implement culturally relevant pedagogy in your classroom.

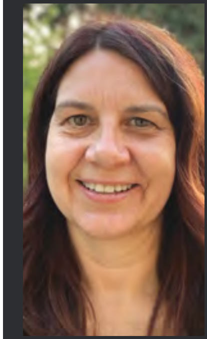
There are key elements within each of these three focus areas. Within the curriculum, it is important to think about the contexts in which computing concepts are taught, and how connections are made with issues that are meaningful to your learners. Advisor Lynda Chinaka sums this up thus: "Children need to be presented with curricula topics that stimulate their interest, reflect their life experiences and their heritage. Presenting them with role models of different ethnicities and backgrounds who have advanced work in computing, such as Joy Buolamwini, who is a computer scientist and founder of the Algorithmic Justice League, is equally important if children are to be given the message that this field is potentially open to them."

Equitable teaching approaches such as open-ended, inquiry-led activities and discussion-based collaborative tasks are key to providing opportunities for all learners to express their ideas and their identities through computing. Finally, inclusive representations of a range of cultures, and the accessibility of the learning materials, are of great importance to ensure that all learners feel that computing is relevant to them.

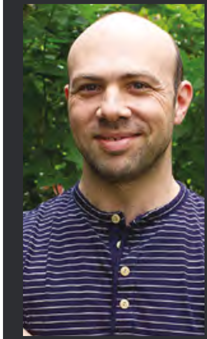
OUR ADVISORS



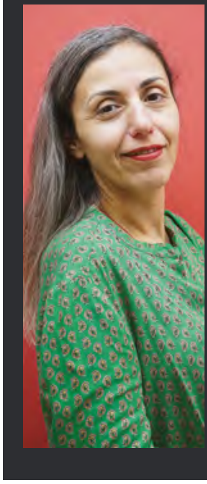
Lynda Chinaka is a senior lecturer in computing in education at the University of Roehampton, UK. A key goal of hers is to ensure that there is equity and creativity in the computing taught in schools.



Joanna Goode is the Sommerville Knight Professor of Education at the University of Oregon, USA. Her research focuses on equity, inclusion, and anti-racism in computer science education.



Mike Deutsch is director of learning services at the Canadian education charity Kids Code Jeunesse, and a master's student at McGill University. His work focuses on pedagogy in computing education.



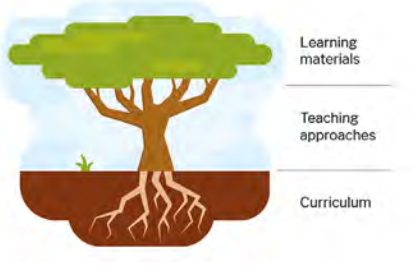
Yota Dimitriadi is an associate professor of technology-enhanced learning and computing education at the University of Reading, UK. She has been awarded the National Teaching Fellowship Award by Advance HE in recognition of her work in inclusion and diversity.

> COMPUTER SCIENCE EDUCATION NEEDS TO BE INCLUSIVE, USING IMAGERY AND METAPHORS FROM DIFFERENT CULTURES

Teachers' perspectives

Joe Arday and Alain Ndabala were two of the teachers who ran activities with some of their learners aged 13–14 and 16–18. They both originally worked in the tech sector and have been teaching computing for ten years. We asked them about their experience of being part of the project and how they plan to use the guidelines in their own practice.

Joe: "*I have been able to reflect on how to further improve my teaching practice and pedagogy to ensure that the curriculum taught is culturally diverse and caters for all learners that I teach. I plan to review the computing curriculum taught in my computing department and sit down with my colleagues to work on how we can implement the guidelines in our units of work for Key Stages 3 to 5. I am also planning to deliver CPD sessions to colleagues inside and outside my school. The guidelines will also help my department to work towards one of my school's aims to encourage an [anti-racism] community and curriculum in my school.*"



The three main focus areas of the guidelines are like a tree

Alain: "*Our classrooms are a microcosm of the local population; it is therefore important for teachers to learn about culturally relevant pedagogy because it enables them to adopt a responsive pedagogy that expands the knowledge of young people that single-perspective computing lessons have not been able to. I have developed an awareness of the academic value of a culturally relevant computing curriculum and learned that computer science education needs to be inclusive, using imagery and metaphors from different cultures. I will use the guidelines to review and enrich our computer science curriculum and include role models from different cultures and topics relevant to young people in Britain today.*"

Learners' perspectives

As part of the project, we asked the teachers in the working group to consult with their learners to understand their perspectives on computing and look at how schools can engage more diverse groups of students in computing qualifications. The responses to the questions below came from Joe and Alain's students.

Q. How did learners feel about computing?

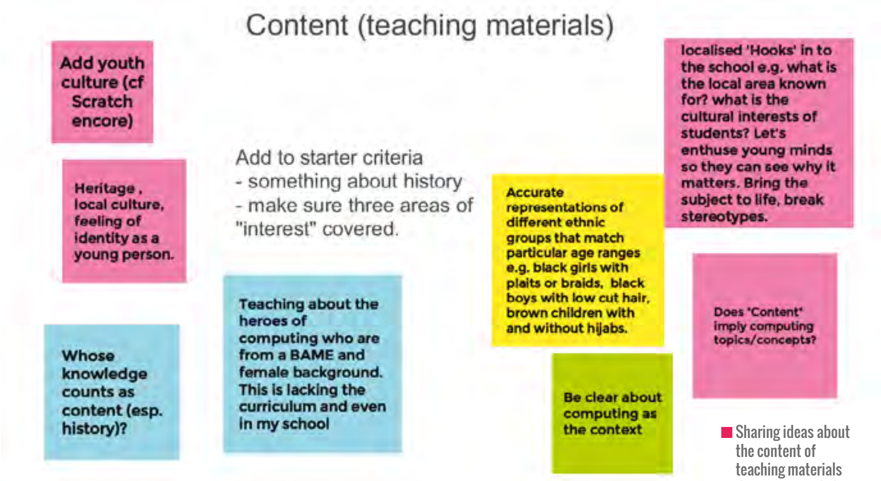
While many of the learners thought that computing was relevant for everyone because of its importance to society and a range of different jobs, several commented that they weren't interested in it or that it would not be relevant to their future careers. The main reasons that learners reported they were put off computing were due to "complex" or "boring" lessons of coding, with a focus on theory rather than practical outcomes. Many reported being inspired by tasks such as producing their own games.

This feedback fits really well with the guidelines we have developed, especially thinking about using a range of teaching methods and highlighting connections between computing and the real world. Allowing learners some choice in projects gives them ownership of their work and gives them a purpose for coding, which they may otherwise see as meaningless or too complex or boring.

Q. What do learners think teachers and schools should do to engage more diverse groups in computing?

Learners had some really great ideas for ways to attract more diverse students to computing. As well as making tasks fun, and engaging learners from a young age, some learners identified the importance of making links with a range of different individual interests, and ensuring everyone knows the benefits, impacts, and job options for computing. One student shared, "I think the exam board and schools need to take into consideration more of the students' reasons for why they would enjoy computing and technology in general. Making the curriculum more personal to students might encourage some interest in the subject."

Another key point that learners highlighted was the importance of organising activities with role models in computing, to inspire people from different groups: "I think we should hold workshops that will bring in people from those industries that are from the communities that are lacking in those industries. This will be mainly for people from different ethnic backgrounds to talk about the lack of diversity in certain industries/ courses in school."



Content (teaching materials)

Add youth culture (cf Scratch encore)

Heritage , local culture, feeling of identity as a young person.

Whose knowledge counts as content (esp. history)?

Add to starter criteria
- something about history
- make sure three areas of "interest" covered.

Teaching about the heroes of computing who are from a BAME and female background. This is lacking the curriculum and even in my school

Accurate representations of different ethnic groups that match particular age ranges e.g. black girls with plaits or braids, black boys with low cut hair, brown children with and without hijabs.

Be clear about computing as the context

localised 'Hooks' in to the school e.g. what is the local area known for? what is the cultural interests of students? Let's enthuse young minds so they can see why it matters. Bring the subject to life, break stereotypes.

Does "Content" imply computing topics/concepts?

Sharing ideas about the content of teaching materials during one workshop

> A PERSON'S CULTURAL IDENTITY AFFECTS THE WAY THEY UNDERSTAND THE WORLD

This feedback again sits well with the guidelines we have developed, identifying a need for inclusive representations, and for a range of different contexts and connections to be employed when delivering the computing curriculum.

What's next?

You can download our guidelines on culturally relevant pedagogy for computing teachers now and begin to explore the resources provided in them (**helloworld.cc/crpguidelines**). They include links to different curricula, and we have highlighted units and lessons in the Teach Computing Curriculum (**helloworld.cc/tcc**) that promote key aspects of the approach. There are also links to academic papers and books, as well as videos and courses that can be used for professional development.

While this is the end of the current project, we're excited about developing it further and plan to work with more computing teachers and their learners to incorporate culturally relevant pedagogy into computing lessons. Watch this space to hear about future projects and opportunities to get involved!

We would like to thank our working group of teachers and academics, as well as the learners who contributed their time and ideas to this project: Joseph Arday, Lynda Chinaka, Mike Deutsch, Yota Dimitriadi, Amir Fakhoury, Dr Samuel George, Joanna Goode, Alain Ndabala, Vanessa Olsen-Dry, Rohini Shah, and Neelu Vasishth. (HW)

HAYLEY LEONARD, DIANA KIRBY, AND SUE SENTANCE

The authors are all affiliated with the Raspberry Pi Computing Education Research Centre at the University of Cambridge (**helloworld.cc/rpiresearchcentre**). Hayley is a research scientist focusing on diversity and inclusion in computing. Diana is the research team programme coordinator. Sue is the director of the research centre and chief learning officer at the Raspberry Pi Foundation.

FEATURE

©Sashkin/stock.adobe.com

GREEN INFORMATION TECHNOLOGY

The global IT industry generates as much CO_2 as the aviation industry. **Beverly Clarke** considers the hidden impact of our IT use and the changes we can make

With the onset of the pandemic, the world seemed to shut down. Flights were grounded, fewer people were commuting, and companies and individuals increased their use of technology for work and communication. On the surface, this seemed like a positive time for the environment. However, I soon found myself wondering about the impact that this increased use of technology would have on our planet, in particular the increases in energy consumption and e-waste. This is a major social, moral, and ethical issue that is hiding in plain sight — green IT is big news.

Energy and data centres

Thinking that online is always better for the planet is not always as straightforward as it seems. If we choose to meet via conference call rather than travelling to a meeting, there are hidden environmental impacts to consider. If there are 50 people on a call from across the globe, all of the data generated is being routed around the world through data centres, and a lot of energy is being used. If all of those people are also using video, that is even more energy than audio only. Not only is the amount of energy being used a concern, but we must also ask ourselves how these data centres are being powered. Is the energy they are using coming from a renewable source? If not, we may be replacing one environmental problem with another.

What about other areas of our lives, such as taking photos or filming videos? These two activities have probably increased as we have been separated from family and friends. They use energy, especially when the image or video is then shared with others around the world and consequently routed through data centres. A large amount of energy is being used, and more is used the further the image travels. Similarly, consider social media and the number of posts individuals and companies make on a daily basis. All of these are travelling through data centres and using energy, yet for the most part this is not visible to the user.

E-waste

E-waste is another green IT issue, and one that will only get worse as we rely on electronic devices more. As well as the potential eyesore of mountains of e-waste, there is also the impact upon the planet of mining the precious metals used in these electronics, such as gold, copper, aluminium, and steel. The processes used to mine these metals lead to pollution, and we should also consider that some of the precious metals used in our devices could run out, as there is not an endless supply in the Earth's surface.

It is also problematic that a lot of e-waste is sent to developing countries with limited recycling plants, and so much

USEFUL RESOURCES

- **helloworld.cc/bcsgreenit:** BCS (British Computer Society, The Charted Institute of IT) has a Green IT specialist group, a community of IT professionals and enthusiasts who are passionate about green IT
- **helloworld.cc/dispatches:** Dispatches documentary *Online Habits Killing the Planet* (only available to UK and ROI residents)

> GREEN ICT IS A MAJOR SOCIAL, MORAL, AND ETHICAL ISSUE THAT IS HIDING IN PLAIN SIGHT

of the e-waste ends up in landfill. This can lead to toxic substances being leaked into the Earth's surface.

First steps towards action

With my reflective hat on, I started to think about discussions we could have with pupils around this topic, and came up with the following:

- Help learners to talk about the cloud and where it is located. We can remind them that the cloud is a physical entity. Show them images of data centres to help make this real, and allow them to appreciate where the data we generate every day goes.
- Ask learners how many photos and videos they have on their devices, and where they think those items are stored. This can be extended to a year group or whole-school exercise so they can really appreciate the sheer amount of data being used and sent across the cloud, and how data centres fit with that energy consumption. I did this activity and found that I had 7,163 photos and 304 videos on my phone — that's using a lot of energy!

Conference calls have a higher environmental impact than we may realise

©Jacob Lund/stock.adobe.com

- Ask learners to research any local data centres and find out how many data centres there are in the world. You could then develop this into a discussion, including language related to data centres such as sensors, storage devices, cabling, and infrastructure. This helps learners to connect the theory to real-world examples.
- Ask learners to reflect upon how many devices they use that are connected to the Internet of Things.
- Consider for ourselves and ask parents, family, and friends how our online usage has changed since before the pandemic.
- Consider what happens to electronic devices when they are thrown away and become e-waste. Where does it all go? What is the effect of e-waste on communities and countries?

Tips for greener IT

UK-based educators can watch a recent episode of TV programme *Dispatches* that investigates the carbon footprint of the IT industry (**helloworld.cc/dispatches**). You can add the following tips from the programme to your discussions:

- Turn off electronic devices when not in use
- Use audio only when on online calls
- Dispose of your old devices responsibly
- Look at company websites and see what their commitment is to green IT, and consider whether we should support companies whose commitment to the planet is poor
- Use WiFi instead of 3G/4G/5G, as it uses less energy

These lists are not exhaustive, but provide a good starting point for discussions with learners. We should all play our small part in ensuring that we #RestoreOurEarth — this year's Earth Day theme — and having an awareness and understanding of the impact of our use of electronic devices is part of the way forward. (HW)

BEVERLY CLARKE

Beverly is the national community manager for Computing at School, which is the teacher-focused part of BCS, the Chartered Institute for IT. She is a former computing teacher and is the author of *Computer Science Teacher: Insight Into the Computing Classroom*. Beverly is always interested in IT that can lead to the development of people and communities.

THE IMPACT OF VALIDATION

Polly Card explores visualising Black and Latino male students' educational experiences and the importance of validation from educators

Black and Latino men have been historically underserved and underrepresented at community colleges, a form of public post-secondary education in the USA. Fewer than 40 percent of men of colour graduate within six years, the expected timescale being two to four, and only two-thirds who start college, finish (helloworld.cc/edustats). These statistics are important, as college degree attainment is a key predictor of labour market outcomes. To give voice to some successful students behind these statistics, as part of my PhD research I produced a collaborative film-based ethnography named *Tidepools* (helloworld.cc/tidepools).

The medium of film was used to make the research accessible to a wider audience. It is helpful to provide visual examples and concretise behaviours for educators, to increase interest in and commitment to supporting Black and Latino students' educational success. This article reinforces the information presented in the documentary, exploring areas that impact the success of men of colour. This is followed by tips for educators about how to validate learners and support student success, drawing upon research and students' reflections on their time in educational settings.

What impacts the success of students?

In this research, success was defined as self-reported by the students. All those interviewed had a 3.0 grade point average and intended to persist at community college. Here are some factors that were found to impact the academic success of male students of colour:

Environmental factors: Affected students are more likely to face long commutes to college, or have family and work commitments that mean they can't attend full time. One educator in the film reported that this "can oftentimes prevent men of colour from being fully engaged and successful in college". One Latino student, Marcos, noted that he felt pressure to be a wage-earner during his education. "The very first couple of years I worked a lot … that would be like carpentry, electrical work, plumbing … I was a full-time student and I was working more than 40 hours a week as well." Dmitri, a student who travelled daily from Mexico to attend college at Southwestern Community College in Chula Vista, described his journeys across the border: "It's a four-hour commute in which every day you look at a guy, with a weapon, who is interrogating you … But you gotta do what you gotta do, right?"

Stereotypes: Racial prejudice and gender stereotypes shape how educators and societies see students. The students identified low expectations and negative stereotypes based on their race, ethnicity, and gender as salient elements of their experiences, both on their campuses and in wider society. One student, Devin, was studying at community college and learning to speak four different languages. He noted, "People don't expect a Black person to be learning all these different languages. When I speak well or do well [some people are] profoundly surprised … Mom made it really clear that I had to try twice as hard to be successful, [she said] make sure you do extra, to be on even ground with everyone else."

Educators: Faculty, mentors, and those who interact with students have the potential

> “ EDUCATORS CAN TRANSFORM VULNERABLE STUDENTS INTO POWERFUL LEARNERS WHO ARE EXCITED ABOUT LEARNING

Devin

Marcos

The participants in the film all reported that specific educators gave them positive encouragement and feedback, increasing their sense of belonging

to help students overcome hurdles that typically disadvantage their peers. Marcos, for example, spoke of an educator who helped him find work on campus, and introduced him to a program that invited him to visit university campuses. Visiting university campuses and being encouraged by educators made him see himself as a candidate, and as someone who could achieve in higher education.

> INSTITUTIONS WOULD BENEFIT FROM A CURRICULUM THAT IS INCLUSIVE, DEMOCRATIC, AND REFLECTIVE OF STUDENT BACKGROUNDS

How can educators validate students?

Participants spoke frankly about challenges they had faced and how educators had helped guide them through their educational experiences. The findings suggest that educators can transform vulnerable students into powerful learners who are excited about learning. To validate learners, educators can adopt the following roles:

Understanding supporter: One role a validating educator plays is that of an understanding supporter. Research suggests that educators, regardless of their ethnic and racial background or gender, can effectively teach men of colour by providing a balance between the academic challenge of rigorous coursework and support (**helloworld.cc/wood2015**). This can be direct support, for example supporting them to learn the academic course content, or making them aware of academic support such as campus resources.

Navigator: Educators may adopt the role of an experienced navigator to the school or higher education setting and beyond. Many minority students are first-generation college-goers and are less likely to have family and friends who can introduce them to university campuses and facilitate access to appropriate extracurricular activities. Omar was a first-generation college student whose parents were Mexican immigrants. Educators helped him navigate the system towards gaining a degree. “I never knew what transferring meant [to go to a university] … it opened up my eyes to many more possibilities.” Omar found that educators gave him the connections to progress in life: “I wouldn’t be where I am today if it wasn’t for the mentors I had at community college.”

Direct approach: A direct and friendly approach towards students, and purposefully reaching out to help them succeed, leads to validating experiences. All students spoke of how important it was for them to be personally engaged with the content, notably through relevant coursework, which helped them engage with learning and feel a sense of belonging. This observation suggests that institutions would benefit from a curriculum that is inclusive, democratic, and reflective of student backgrounds. One student, Terry, noted that what turned things around for him was taking courses with content that he found culturally relevant, and meeting an educator with a teaching philosophy that he considered to be “radically different from other experiences”. The educator told the students that she cared about them, and this gave Terry confidence and transformed his approach to learning.

All educators can teach in a way that supports the success of men of colour. This research shows that it helps to do so using validating practices that foster a sense of belonging, and with an understanding of students’ lived experiences, including their racial and gender identity, as well as the external pressures they face in their lives and communities. Gloria Anzaldúa, a cultural theory researcher, states that “change requires more than words on a page — it takes perseverance, creative ingenuity, and acts of love” (**helloworld.cc/anzaldua2002**). *Tidepools* is an example of film-based ethnography that goes beyond words on a page, documenting and sharing how validating acts between educators and students contribute to the success of Black and Latino students. (HW)

POLLY CARD

Polly is a visual researcher. She creates documentaries for education, art, and communities, and is currently a videographer for the Raspberry Pi Foundation (**@pollycard**).

COMPUTING EQUITY CHAMPIONS

The digital divide is real, and now recognised more than ever. **Yolanda Payne** highlights the work being done to bring equity to computing

There are very few aspects of modern life that are not touched by technology, computers, and computer science. As the need for technological solutions to the problems universally felt in the pandemic quickly became apparent, so too did the realisation that many students, families, and communities did not have adequate access to what has become a basic necessity: technology.

When looking at the difficulties that we face — personally, professionally, and in society at large — they can seem overwhelming. In situations like these, it's important to recognise the issues, decompose those problems into workable portions, and figure out how we can use what we have to make an impact.

That battle for equity in computing has already been taken up by many educators, some long before the pandemic, and a few of those engaged in this work are profiled here. Their work has had ripple effects of positive change on the lives of the students impacted and their wider communities. I hope that their examples can provide powerful blueprints for those beginning to engage in the work, and inspire you to work towards change in your context.

YOLANDA PAYNE

As opportunities in technology and education abound, Yolanda hopes to help students find their path in the field. A former classroom teacher, Yolanda is now a fellow at the Constellations Center for Equity in Computing at the Georgia Institute of Technology, USA. She also facilitates Code Club and Girls Who Code chapters, while still finding time to tinker at home with her two sons (**@ydcp**).

STEM NOLA

Since its founding in 2014, STEM NOLA (**stemnola.com**) has impacted over 65,000 students from kindergarten to college level. Dr Calvin Mackie created the organisation in New Orleans, USA, with a mission to grow, engage, and inspire future innovators, creators, makers, and entrepreneurs. The organisation aims to expand STEM educational opportunities, especially for under-resourced communities and communities of colour. They want every child, regardless of their race, ethnicity, or gender, to have opportunities to excel in the STEM field, which they believe will be an engine for quality jobs and careers in the 21st century.

Dr Mackie was inspired to create STEM NOLA after his own children became disillusioned with STEM, with his third-grade son coming home one day saying, "I don't like science anymore. My teacher was just writing on the board and not doing anything hands-on." So, in their family garage, Dr Mackie started doing his own STEM projects with his son. Quickly, his son started getting A grades, while his friends struggled. He soon realised that the difference was the result of hands-on exposure to STEM. If his son's friends were exposed to those same experiences, they would be doing just as well. Dr Mackie sought to share these resources with others who might not get the same opportunities, and STEM NOLA soon became the vehicle to do just that. "We are creating a cradle to career pipeline, so that our children can have all the skills necessary to not only compete in their community, but compete with the world."

Community engagement is at the heart of everything STEM NOLA does, and they pride themselves on serving as a trusted partner to families and communities, helping them navigate the STEM ecosystem and bringing them high-quality STEM programs and events. They offer three main strands of support: STEM NOLA @ Home, STEM NOLA @ School, and STEM NOLA in the community. STEM NOLA @ Home includes virtual live sessions that schools and families can join and follow, along with hands-on activities. Accompanying activity cards are available in English and Spanish, and a parent resource guide is provided to make these sessions as accessible as possible. Casey McGee, STEM NOLA's communications manager, explains the importance of a hands-on approach to learning: "We lead K-12 students in activities like building mechanical lungs, traffic lights, hydraulic bridges, and battery-powered cars. When kids and their families build a model with their own hands and understand the science behind it, that's when the inspiration takes root. We create authentic connections between the classroom and the real world. By rooting our content in the experiences and environment of our participants, STEM comes to life through their everyday lives."

STEM NOLA promotes a hands-on approach to learning

DATA STORIES

Data Stories (**datastories.cc**) is a community of learners, educators, and trailblazers, with 100 members from 20 states in the USA. Data Stories provides access to free instructional materials and activities (both self-guided and teacher-led), and online after-school clubs and programs, to give students an opportunity to delve into the fundamentals of data science, analytics, and data storytelling. Their educational resources use tidy data sets (data that has been cleaned, but not scrubbed) in freely accessible online data analysis and data storytelling platforms.

Students are allowed to choose individual topics, and use the data science skills acquired to support their questions and opinions with facts and data sets. By using data sets available to professionals, students are empowered to use their voice and also gain marketable real-world skills. Data Stories also hosts guest speakers from a wide array of industries, to support learners in developing a keen understanding of how to think like a computer scientist or data analyst.

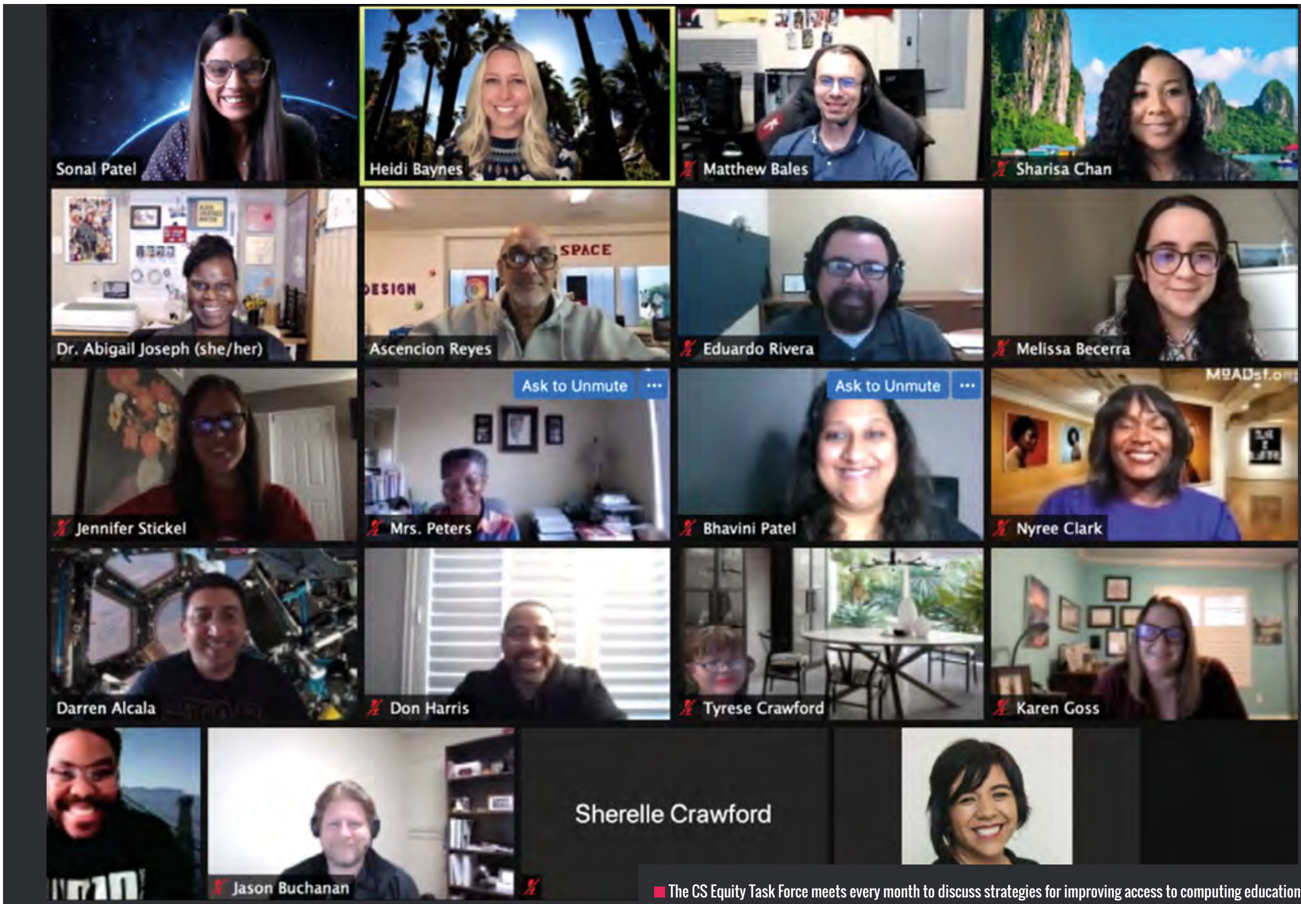
Cameron Fadjo, the founder of Data Stories, explains that its founding came out of the pandemic. "After years and years of working to elevate the visibility of computer science education in K-12 education, the possibility that the pandemic was going to upend all that work struck a chord with me. My inspiration to found Data Stories was to create a scalable program that meets a critical need (preparing students to become coders and analysts in a data-driven world), while empowering a generation."

The folk at Data Stories believe that distilling ideas down into key findings, which can then be explained to audiences through storytelling, is a valuable knowledge economy skill. Cameron explains, "Data storytelling is all about using data analyses to explain a few findings from data analysis through the power of stories. Many young people are deeply invested and compelled to share their own stories. By injecting thoughtful analyses and compelling visual elements into a story being shared by a young person, we believe the story is elevated and is potentially more credible when there are facts and findings to support a position."

The hope is that with a return to a semi-regular school year, Data Stories club organisers will be able to enlist more students in the pursuit of data storytelling. "As businesses seek employees who are able to analyse large amounts of information and effectively communicate their findings to a wider audience, we think our Data Stories programs and the larger efforts around data storytelling and data literacy will create numerous professional and personal opportunities for generations to come."

© Rawpixel.com/stock.adobe.com

Distilling ideas down into key findings that can be explained to audiences through storytelling is a valuable knowledge economy skill



The CS Equity Task Force meets every month to discuss strategies for improving access to computing education

SAN BERNARDINO COUNTY SCHOOL SYSTEM

The San Bernardino County School System in California, USA, has embraced policies and programs that give all students access to quality educational experiences, including in their computer science studies. Activities such as workshops, hackathons, and e-sports leagues are designed to engage as many students across the district as possible, with initiatives in place that engage students across grade levels and in conjunction with surrounding districts. These experiences are designed to engage students and align with their interests, as well as exposing them to quality practices in integrating technology.

Recruitment and participation of underrepresented groups is an integral part of the implementation of these opportunities. For example, the school system recruits diverse groups of mentors, teachers, and guides from industry and education for these events and activities, to give all students a glimpse of themselves in future roles. Students are shown real-life examples of what they can be, and are reminded that they have a place in the field of technology.

The cornerstone of these initiatives has been the CS Equity Task Force, formed by Sonal Patel (coordinator of digital learning innovation at San Bernardino County Superintendent of Schools) and Heidi Baynes (edtech coordinator at Riverside County Office of Education). The Task Force is made up of a diverse group of educators, administrators, counsellors, and library-media specialists, and has the aim of increasing access to computer science education for students from kindergarten to grade 12. The group meets every month to work on activities to fulfil this mission, ranging from sharing updates on the work of computing education organisations; to providing networking opportunities for those interested in computer science and equity; to hosting meaningful discussions focused on strategies and ideas to implement changes and integrate best practice in computing education.

We have access to technology that can change lives. There are millions of talented people who could both be a benefit to, and benefit from, the promises that technology holds. There are already people doing the work. The question now is: what are the rest of us waiting for? (HW)

> THE BATTLE FOR EQUITY IN COMPUTING HAS BEEN TAKEN UP BY MANY EDUCATORS, SOME SINCE LONG BEFORE THE PANDEMIC, AND THERE ARE RIPPLES OF POSITIVE CHANGE

The WiFi's down! Or is it?

© Elnur/stock.adobe.com

MANAGING MISCONCEPTIONS

Ben Hall introduces some common types of misconceptions and strategies for spotting them

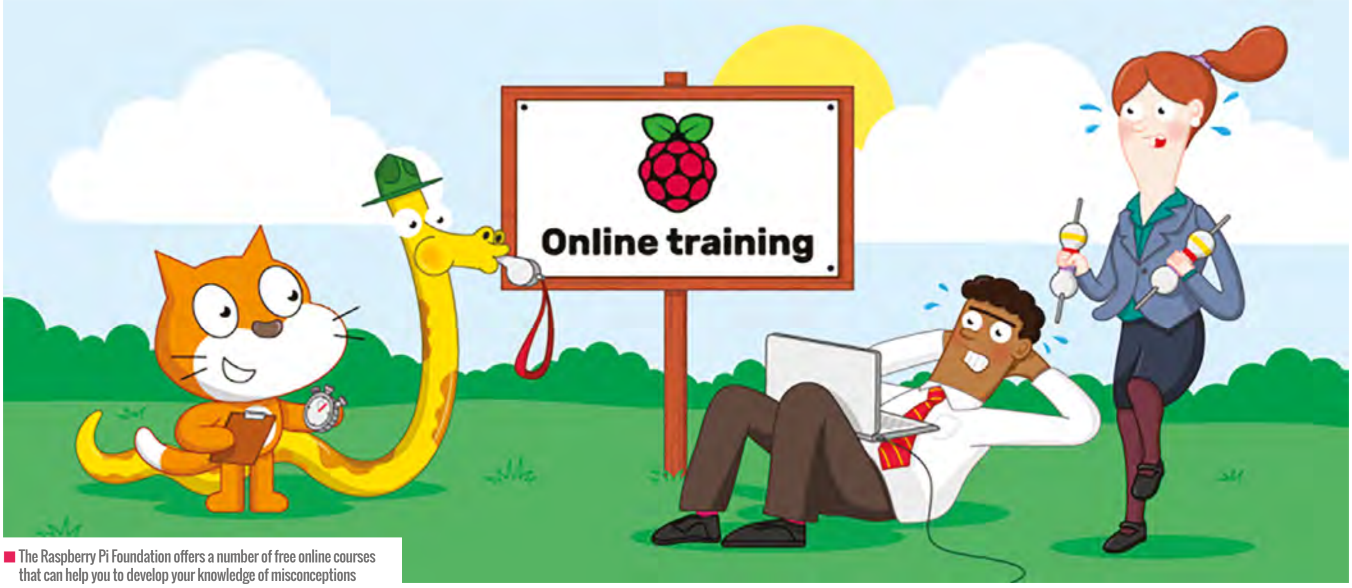
As teachers, a crucial part of our job is to address the misconceptions, the false or incorrect views and opinions, of our learners. Misconceptions can be challenging for teachers to identify because the learner might think they understand a concept well, but they might be wrong! They may also try to build subsequent knowledge on top of their existing (incorrect) knowledge, which will likely come unstuck because the first level of understanding was not solid enough. The more quickly we identify a misconception, the less existing knowledge we have to replace, and the easier it will therefore be to rectify. In this article, I will discuss common types of misconceptions, where they may have originated, and how we might go about addressing them.

Vernacular misconceptions

In computing, many misconceptions are vernacular, where the same words and phrases are associated with multiple concepts. An example of a vernacular misconception is the belief that the internet and the World Wide Web are the same thing. This arises from people using the terms internet and World Wide Web interchangeably without understanding the difference between them. Prolonged exposure to such a misconception can make it difficult to shift, particularly when the terms involved are so widely used and misrepresented. A misconception like this needs to be addressed head-on, at the point when learners are first introduced to the internet as a network of networks, and then the terms need to be used accurately once they've been introduced.

Accidental misconceptions

Misconceptions can also be accidentally introduced, simply by using common language or phrasing. For example, when we store data on the cloud, it's actually stored in server farms on land, or sometimes even



The Raspberry Pi Foundation offers a number of free online courses that can help you to develop your knowledge of misconceptions

> WELL-THOUGHT-OUT CURRICULUM DESIGN CAN HELP MITIGATE SOME OF THE PROBLEMS CAUSED BY MISCONCEPTIONS

under the sea, but never in the sky. The use of the word 'cloud' is in this instance a paradox. Another common example is the misuse of the term 'WiFi', and it's probably something we're all guilty of. When a web page fails to load, how often do people say there's a problem with the WiFi? There may well be, but it could also be a problem with the physical network, or a particular website could be unavailable. The ubiquitous use of the term is misleading, and hinders learners' understanding of the components of a network and the delivery mechanism of the World Wide Web.

Deliberate misconceptions

We sometimes introduce misconceptions deliberately, to make concepts more accessible and easier to understand for younger learners, in the knowledge that they will be undone later. In England, learners at ages five to seven are expected to understand what algorithms are. In the National Centre for Computing Education (NCCE), we define an algorithm as 'a precise set of ordered steps which can be followed by a human or a computer to do a task'. It is unrealistic to assume that a child aged five to seven will fully understand what is meant by 'a precise set of ordered steps', so in most cases their first experiences of algorithms will be in the form of pictures or diagrams, which lack the precision to fit the true definition of an algorithm. As they progress through school, and their conceptual understanding of the term 'precise' develops, their expectations of what an algorithm should look like will alter.

Thinking about the tools

Often, the tool you are using can lead to conceptual misconceptions being formed. Take the example of the ScratchJr and Scratch platforms, which will provide many learners' first experiences of coding. On both platforms, it is relatively easy to run separate commands concurrently; you just need more than one green flag block with commands attached to them. If children learn that they can do this in Scratch, they may reasonably assume that it is also easy to do this in Python and other text-based languages, whereas the reality is that threading (the method used to run commands concurrently in Python) is actually quite complex. The origin of this misconception is often that primary teachers may not have sufficient knowledge of other coding languages to see the problem, and this can present an even bigger issue: you can't address or avoid a misconception if you are not aware of it in the first place. A conceptual belief that a concept such as threading is easy can lead to learners experiencing frustration when programming in other languages, and could potentially deter them from pursuing computing further.

Addressing misconceptions

Well-thought-out curriculum design can help to mitigate some of these problems. When designing the Teach Computing Curriculum (**helloworld.cc/tcc**), we were aware that concepts such as threading were troublesome for many teachers introducing Python to learners at lower secondary level, if they have been used widely in Scratch with learners aged five to eleven. When we were designing programming activities for learners aged five to eleven, we liaised closely with our colleagues who were designing activities for learners aged 11 to 16. Consequently, we chose to avoid activities that involved multiple code snippets running concurrently and the broadcast function, knowing that these would create problems which would need to be unpicked later on.

There are also a number of things that you as a teacher can do to spot misconceptions. Here are my top tips:

© Have a nice day/stock.adobe.com

■ Well-crafted multiple-choice questions can help to uncover misconceptions in the classroom

> “IMPROVING YOUR SUBJECT KNOWLEDGE CAN HELP YOU TO IDENTIFY AND ADDRESS MISCONCEPTIONS

Learn as much as you can about the topic you’re teaching

Improving your subject knowledge can help you to identify and address misconceptions. Through the Raspberry Pi Foundation and the NCCE, you can access free online courses covering areas such as teaching primary computing (**helloworld.cc/teachprimarycourse**), programming (**helloworld.cc/programmingcourse**), computing systems and networks (**helloworld.cc/systemscourse**), and physical computing (**helloworld.cc/physcompcourse**). All of these courses highlight misconceptions and suggest ways of addressing them. If you’re new to teaching a certain topic, you can also ask more experienced staff, or search online for common misconceptions in that area.

Plan specific questions to expose and challenge misconceptions

Effective questioning can allow you to pick out misconceptions, or to challenge a learner’s misconceptions and encourage them to accept the accurate conceptual understanding. Thinking carefully about potential misconceptions when you’re planning, and creating questions to deliberately tackle them, is a great strategy. Here are some examples for the misconceptions we mentioned earlier:

- The internet was invented in the 1960s. When was the World Wide Web invented?
- When photos are saved to the cloud, where do they actually go?
- When you cannot access the internet, does that mean your WiFi is down?
- Your algorithm needs a computer to understand the command ‘pick up’. How could you break this down to ensure ‘pick up’ is unambiguous?
- I see you have created a program with multiple scripts running at once. Can you combine some of them to make your code more linear?

Explore a range of pedagogies that focus on understanding as well as skills

Some pedagogies also suit the challenging of specific misconceptions and ensuring that learners understand the concepts behind the tool they are using. For example, asking a learner to trace some code and then predict what they think the code will do could reveal a huge range of misconceptions, such as whether they have recognised that a loop running two out of four times will still have a partial outcome.

Use diagnostic questions, such as multiple-choice questions, and deliberately introduce the misconception as a possible answer

Assessment is a powerful tool to help you uncover misconceptions. By using carefully planned multiple-choice questions (MCQs), you can identify whether a learner has grasped a particular concept, and you can also predict possible misconceptions if the answer they give is incorrect. They can be used in a variety of different ways, both during and in between lessons. However they’re introduced, they should always inform teaching. Each question provides you with a rich snapshot of understanding from a group, but the question is only valuable if it informs what you do next. You can find some excellent tips for designing and using MCQs at **helloworld.cc/mcqs**.

Having an awareness of the different types of misconceptions, and a range of strategies to tackle them, will help to develop our learners’ understanding and experience in the computing classroom. (HW)

BEN HALL

Ben is a learning manager at the Raspberry Pi Foundation, where he develops resources for the NCCE. He is a CAS Master Teacher and a Raspberry Pi Certified Educator (**@hengehall**).

EXPLORING THE HINTERLAND

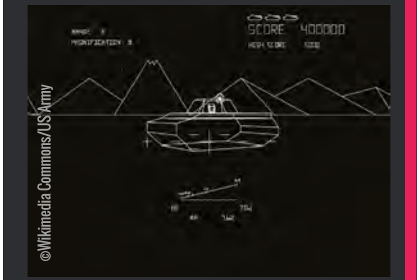
Alan Harrison defines hinterland knowledge and looks at how reading around a subject can improve your teaching

Having a laser focus on the exam board specification while teaching GCSE is extremely worthwhile: the exam is, after all, how students will be assessed. Knowing the difference between a worm and a virus will gain a mark. Knowing that in 1971 Bob Thomas of BBN wrote the first worm, called Creeper, will not. Telling the examiner that the inventor of email Ray Tomlinson wrote the Reaper worm to get rid of Creeper is also pointless. But the teacher who can share this hinterland knowledge with their students brings colour and interest to the subject, which can improve both engagement and retention. Meanwhile, teachers of lower secondary and below should be delivering a rich curriculum that is enjoyable in its own right, not just laying the groundwork for GCSE studies. In short, there is both joy and pedagogical value in exploring the hinterland with your students.

MAKING LEARNING CONCRETE

"Bring abstract concepts to life with real-world, contextual examples and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives." (**helloworld.cc/hinter5**).

For example, when explaining the difference between raster and vector graphics, include a bit of history and some concrete examples such as radar screens, early video games, and the story of the Bradley Trainer (see image below) to bring the ideas to life.



©Wikimedia Commons/US Army

Atari's vector graphics arcade game Battlezone was adapted by the US army to train soldiers

What are core knowledge and hinterland knowledge?

We can think of the core as the examinable material: what do students need to know to pass the exam? But as English expert Christine Counsell warns in her 2018 blog, "If … for the purposes of teaching, we reduce it to those propositions, we may make it harder to teach, and at worst, we kill it."

Counsell also makes the point that students of English who commit to memory the plot, characters, and stylistic features of a text from the revision guide will undoubtedly do well, but a true understanding can only be gained by "reading, bathing in the text, delighting in the text, alone and with others" (**helloworld.cc/hinter1**).

In his blog post *Signposting the hinterland* (**helloworld.cc/hinter2**), Tom Sherrington explains that the hinterland is as important as the core, and serves the purpose of:

- Increasing depth: niche details about a particular area of study deepen and enrich the core
- Increasing breadth: wider surveys across the domain of any curriculum area help to locate any specific core element within a wider frame

Sherrington gives several examples, including a music teacher sharing "a timeline of musical genres through samples from Purcell to Björk — something we repeatedly reference as we explore our chosen composers in more depth".

The computing hinterland

Like the music teacher, we can illuminate our subject with reference to the past. The backstory of computing is interwoven with world history, and I regularly discuss new learning in the context of historical events. Teaching algorithms? Start with the origin of the word (from the Persian scholar al-Khwarizmi of the Islamic Golden Age) and discuss how John von Neumann (yes, he of the architecture) devised merge sort to crunch numbers while working on the atomic bomb that fell on Nagasaki. Explaining validation? Tell the story of Margaret Hamilton's Apollo 11 flight software and the Lauren bug, and how it led to her inventing software engineering.

We can find a computing angle in almost any event of recent history or current

ALAN HARRISON
Alan is head of computing at William Hulme's Grammar School in Manchester, a United Learning school in the UK. He is a CAS Master Teacher and community leader, and a content author for Raspberry Pi. His book, *How to Teach Computer Science*, published by John Catt Educational, is available now (**@MrAHarrisonCS**).

> NARRATIVES ILLUMINATE A TOPIC AND MAKE IT STICK IN LONG-TERM MEMORY

Margaret Hamilton with the navigation software that her team produced for the Apollo project

affairs. From the encrypted messages sent during the Babington Plot against Elizabeth I to the problem of racial bias in algorithms used by 21st-century police forces, bringing these narratives into class helps illuminate the topic, improve engagement, and make learning stick around in long-term memory.

Why is hinterland important?

England's new Ofsted inspection framework requires teachers to give learners the "cultural capital to succeed in life" (**helloworld.cc/hinter3**). Cultural capital is the legacy left to us by the great minds of the past, and is every child's birthright. Computer science hinterland knowledge can also help learners build their cultural and science capital, which is important if they are to see themselves as computer scientists (**helloworld.cc/hinter4**).

Several of the pedagogy principles published by England's National Centre for Computing Education can be delivered through exploration of the hinterland — in particular, the reference to storytelling in their advice to make learning concrete (**helloworld.cc/hinter5**; see box for their advice and an example).

The hinterland paradox

Science and pedagogy blogger Adam Boxer clearly sets out the paradox of core and hinterland in his blog (**helloworld.cc/hinter6**):

"There is content which we wish students to remember, and by contrast content we cover in class which we don't deem necessary for them to remember ... The other hand of our paradox though is that without that material, without the 'stuff we don't need our students to remember', our curriculum becomes denuded of wider meaning and majesty: it ceases to be one thread of the epic story of humanity and becomes a sterile and sanitised exam-ready product."

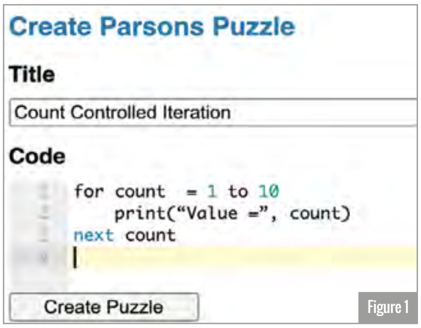
It can be hard to solve this paradox, but try we must. I know that when I delve into the hinterland, my learners are usually engaged, and find it valuable. Indeed, the story of how von Neumann's merge sort led to the atom bomb got a round of applause that I tell myself wasn't entirely ironic. Your mileage may vary, but I promise the journey will be enjoyable.

If you'd like to hear more from Alan, read our review of his book How to Teach Computer Science *later in this issue.* (HW)

DEVELOPING PROGRAMMING SKILLS WITH PARSON'S PROBLEMS

A valuable part of a teacher's toolkit is the use of Parson's Problems to improve pupil ability in sequencing

Creating algorithms and working with a reference language or pseudocode is an area that many students find difficult. This stems partly from their similarity to, but not exact replication of, actual code. For small programs, such as those under 30 lines, the creation of pseudocode can seem trivial, and going straight to the code may seem like the removal of an unnecessary step. However, not only will having no experience with pseudocode affect students in the exam hall, but it will also not prepare them adequately for the design, creation, and repair of code.


Create Parsons Puzzle

Title

Count Controlled Iteration

Code

```
for count  = 1 to 10
    print("Value =", count)
next count
```

Figure 1

I find Parson's Problems useful for developing pseudocode understanding and fostering program comprehension. For the purposes of this article, my leaning is towards the UK's OCR GCSE Computer Science syllabus version of pseudocode. If you deliver other syllabuses, the syntax will have differences, but the techniques still hold.

How do Parson's Problems work?

A Parson's Problem is a programming exercise in which lines or blocks of code are presented in the wrong order, and students have to rearrange them. They are used to engage students with the logical flow and sequence of a problem solution, with the syntax element removed.

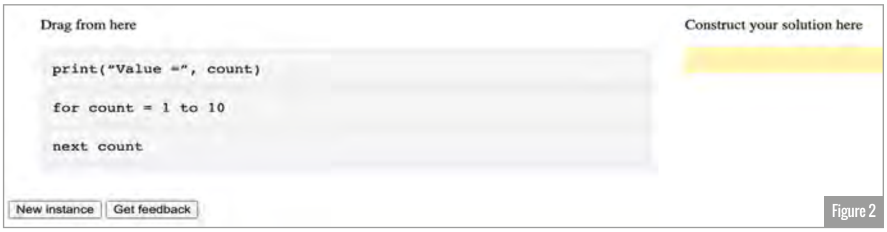
If I were to place this within the PRIMM (Predict–Run–Investigate–Modify–Make) paradigm for structuring programming lessons, Parson's Problems fall into the Investigate and Modify stages. As an illustration of their structure and use, an essential skill students must acquire is the ability to work with iteration. This is a programming construct in its own right, but follows specific logic — it has a sequence. Here is a basic 'for' loop:

```
for count = 1 to 10
    print("Value =", count)
next count
```

As a Parson's Problem, this might be shown as:

```
    print("Value =", count)
for count = 1 to 10
next count
```

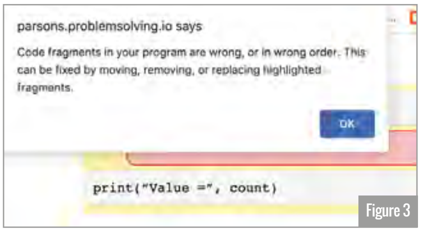
All the elements are there, but not in the right order. To the experienced, this may seem trivial. However, we usually have a range of levels of understanding in our classroom, so it is best not to assume. This provides the opportunity for students to discuss code and code creation, and starting from the basics like this provides a foundation. Building solutions, especially for easy-to-solve problems, is a low-stakes approach that helps reduce cognitive load.


Drag from here

```
print("Value =", count)

for count = 1 to 10

next count
```

Construct your solution here

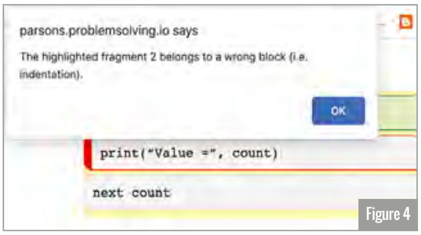
Figure 2


parsons.problemsolving.io says

Code fragments in your program are wrong, or in wrong order. This can be fixed by moving, removing, or replacing highlighted fragments.

```
print("Value =", count)
```

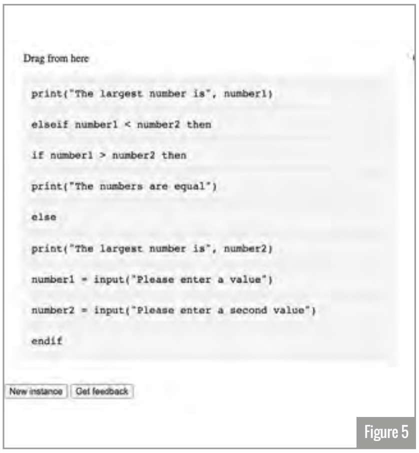
Figure 3


parsons.problemsolving.io says

The highlighted fragment 2 belongs to a wrong block (i.e. indentation).

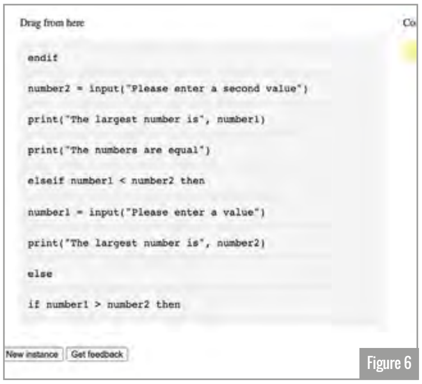
```
print("Value =", count)
next count
```

Figure 4


Drag from here

```
print("The largest number is", number1)

elseif number1 < number2 then

if number1 > number2 then

print("The numbers are equal")

else

print("The largest number is", number2)

number1 = input("Please enter a value")

number2 = input("Please enter a second value")

endif
```

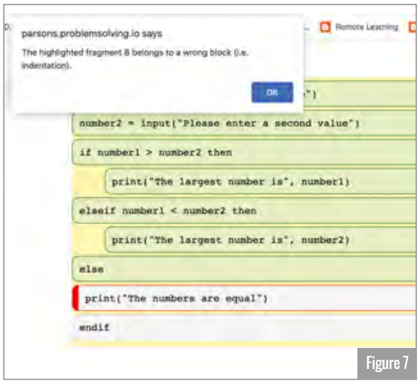
Figure 5



Drag from here

```
endif

number2 = input("Please enter a second value")

print("The largest number is", number1)

print("The numbers are equal")

elseif number1 < number2 then

number1 = input("Please enter a value")

print("The largest number is", number2)

else

if number1 > number2 then
```

Figure 6



parsons.problemsolving.io says

The highlighted fragment 8 belongs to a wrong block (i.e. indentation).

```
")
number2 = input("Please enter a second value")
if number1 > number2 then
    print("The largest number is", number1)
elseif number1 < number2 then
    print("The largest number is", number2)
else
print("The numbers are equal")
endif
```

Figure 7

Parson's Problem generator

My go-to site for creating Parson's Problems is **parsons.problemsolving.io**. It allows the user to input correct solutions (**Figure 1**) and clicking on 'Create Puzzle' will rearrange the pseudocode or code (**Figure 2**).

Students can then drag the lines of code to the right-hand window and press the 'Get feedback' button. If the lines of code are in the right order, they will be highlighted in green. If the order is wrong, a pop-up will appear. This pop-up will either indicate what the problem is likely to be — for example, it may highlight the problem area with a red tint and explain that a code fragment is in the wrong place or needs to be removed entirely (**Figure 3**) — or it will explain that there is an indentation problem with a code fragment (**Figure 4**).

Once you have created a puzzle, the online portal creates a unique URL that can be reused across the year. You can experiment with the earlier example at **helloworld.cc/ppexample1**. Inbuilt into the portal is a random mix-up algorithm, making this site a great 'create once, use many times' resource. It also has the benefit of multiple students not receiving the scrambled problem in the same order. For example, the problem at **helloworld.cc/ppexample2** will present a range of initial problems (**Figures 5 and 6**).

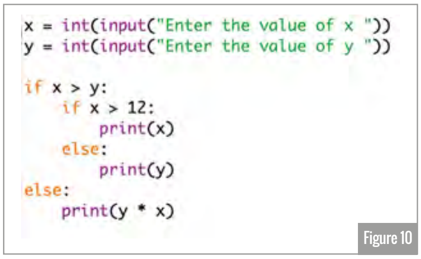
Benefits of the generator

The generator gives students instant, focused feedback that points them towards the type of error they have made and gives them the opportunity to retry. If your students work primarily with an interpreted language such as Python, the debugging process for a Parson's Problem runs in the same way, where the interpreter will stop at the first error (**Figure 7**). This similarity to Python gives students the opportunity to develop their debugging skills in a way that transfers to working with Python. They are obliged to look at the error line, interpret what the error means, and then provide a solution. If successful, the Parson's interpreter will move onto the next error, repeating until a solution is reached.

Next steps

Using Parson's Problems can also extend to providing students with broken code. The problem in Figures 8 to 10 build on the previous selection algorithm and add a new problem element in the form of a logic error (what happens if the two values are equal?). Having a series of related Parson's Problems creates a route for students from algorithm through to code testing. For example, interpreting the algorithm as a flow chart (**Figure 8**), to its pseudocode version (**Figure 9**), and then onto the actual code (**Figure 10**), takes a student from problem statement, through design, and on to creating the code.

Any technique we use in the classroom only forms part of our much larger toolkit. Parson's Problems allow for a relatively low-stakes approach to working with pseudocode or a reference language, and are an ideal scaffolding method, developing students' reading and interpretation skills before they begin writing programs. Give it a go in your classroom today! (HW)



```
x = int(input("Enter the value of x "))
y = int(input("Enter the value of y "))

if x > y:
    if x > 12:
        print(x)
    else:
        print(y)
else:
    print(y * x)
```

Figure 10

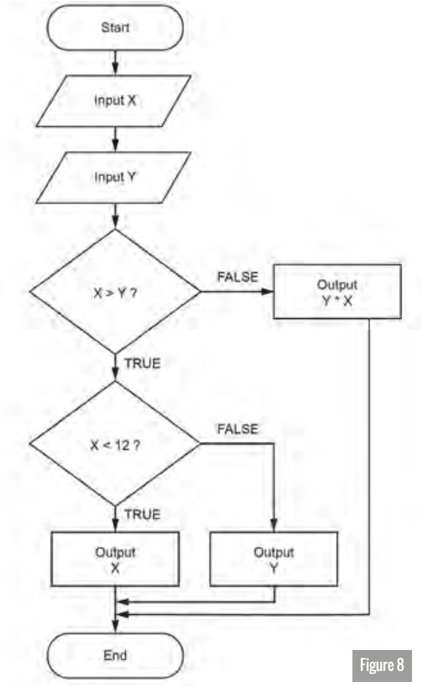
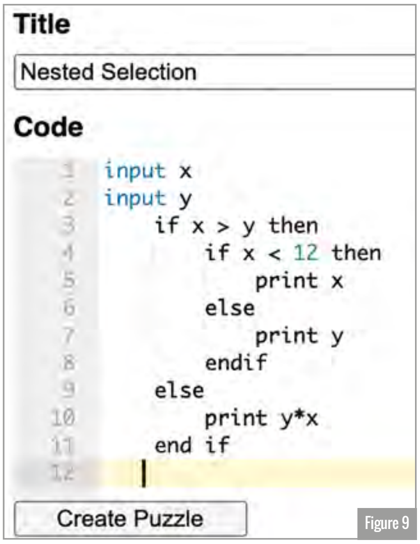


Figure 8



Title

Nested Selection

Code

```
input x
input y
    if x > y then
        if x < 12 then
            print x
        else
            print y
        endif
    else
        print y*x
    end if
```

Figure 9

MICHAEL JONES

Michael is the director of computer science at Northfleet Technology College in the UK. He is a Specialist Leader of Education, an RPi Certified Educator, and a CS Champion for the NCCE (**@MikeJonesCSTalk**).

GOING BACK TO BASICS: PART 2

Rebecca Franks continues to share her experiences of taking a group of beginner digital makers through their first physical computing projects

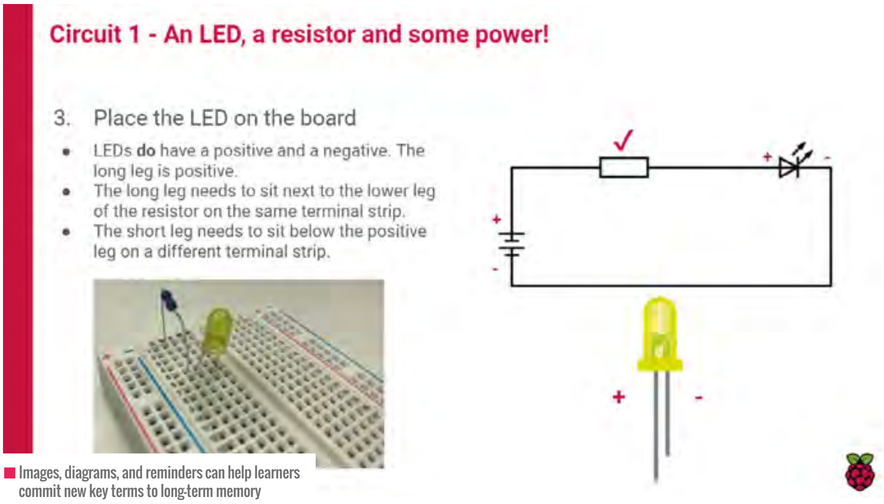
For the last few months, I have been leading sessions with a group of beginner digital makers who wanted to learn how to use LEDs to make an art project. This group had no prior experience with electronics and I wanted to give them the confidence to have a go and develop their skills. I have had my own struggles with digital making, and I really wanted to see if I could help others overcome the same hurdles.

In issue 16 of Hello World, I described the first session that I planned. I decided to start with the very basics and show the group what a breadboard is, how it works, and how they can use it to build circuits. In this article, I will update you on how that session went and also outline the plans for session two.

How do you pronounce LED?

What I absolutely love about my group of digital makers is that they really aren't afraid to ask questions. This is exactly as it should be! When you are learning something new, the more questions you ask, the better it is for you and for those around you. I made this very clear to my group at the beginning. I wasn't there to judge them or criticise them for not knowing things; I was there as a support, to guide them along the way. Whatever questions were on their minds, they could ask without fear of judgement.

Establishing this kind of learning environment is so important for newbie makers. There is often this perception of elitism for those that can make things, and those that can't. This is a really interesting viewpoint, as nearly all the digital makers that I have met as an adult have been incredibly patient and supportive. I think the perception comes from experiences that we have growing up, and also our own fears of looking like we don't know what we are doing.



Circuit 1 - An LED, a resistor and some power!

3. Place the LED on the board
   - LEDs **do** have a positive and a negative. The long leg is positive.
   - The long leg needs to sit next to the lower leg of the resistor on the same terminal strip.
   - The short leg needs to sit below the positive leg on a different terminal strip.

Images, diagrams, and reminders can help learners commit new key terms to long-term memory

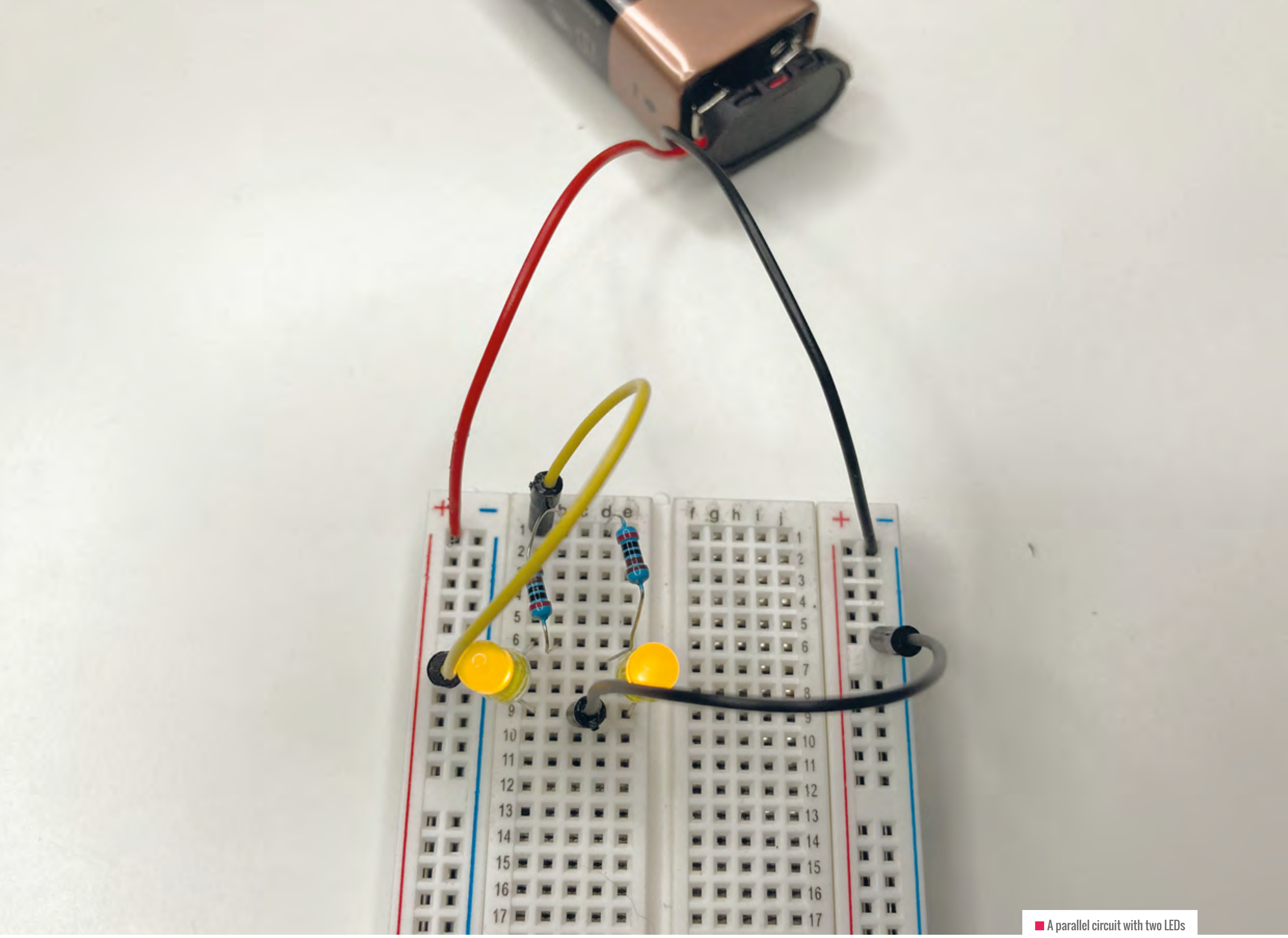
Introducing new information

An LED has two legs. The longer leg is positive and the shorter leg is negative. This seems like a really simple sentence that is easy to remember, but it really isn't. I think it must have taken me several projects to commit that fact to my long-term memory. Throw that sentence in with jumper wires, breadboards, resistors, and a whole plethora of new terminology, and it isn't surprising how quickly the fact is forgotten.

I used lots of images, diagrams, and reminders with my group to help establish the new key terms in their long-term memory. I also used quick-fire quizzes to see what they could remember, to give me an idea of how much support they still needed with certain terms.

For the first few circuits, I used step-by-step guides on how to build them. I included instructions, a circuit diagram, and pictures to help them see what they needed to do at each step. As the circuits progressed, I tried to flip that model a little and taper off from step-by-step instructions. I would show them the circuit diagram first and see if they could then build it independently.

To help with confidence, I encouraged them to have a go and not worry too much if it didn't work. If they didn't quite get there, I then provided step-by-step instructions on a slide deck. My method was to slowly but surely get them to problem-solve themselves and check whether their LED was facing the right way, or if they had placed the jumper wire in the correct part of the breadboard.



A parallel circuit with two LEDs

> **“ TO HELP WITH THEIR CONFIDENCE, I ENCOURAGED THEM TO HAVE A GO AND NOT WORRY TOO MUCH IF IT DIDN’T WORK**

This approach was fairly successful, but I think I probably tapered off a little too early. It is difficult to predict when a group will benefit from the scaffolding being removed, and I made sure I was more careful of this later on in the session.

It worked!

It is so rewarding to watch a group of makers as they reach their aha moment. One by one, they all finished their final circuit of the session, which was to set up two LEDs in parallel. They were very proud of what they had achieved, and somewhat surprised. They left the session feeling accomplished and eager to carry on to the next session.

This next session will take my group through building a circuit that blinks an LED, using only electronics to turn the LED off and on. It uses an integrated circuit called a 555 timer. This is a classic electronics project for newbie makers. I will start the session by getting the group to see what they can remember, and build a new series circuit with the addition of a buzzer and a push button. I will then move on to taking them through building the more complex circuit that uses the 555 timer to blink the LED.

Join me in the next issue of Hello World, in which I will update you on how this session went and reveal my plans for session three. (HW)

REBECCA FRANKS

Rebecca is a learning manager at the Raspberry Pi Foundation. She writes resources for the Teach Computing Curriculum and has over 15 years of experience teaching computing.

FEATURE

Computing classrooms present a number of unique behaviour management challenges

© goodluz/stock.adobe.com

BEHAVIOUR MANAGEMENT IN THE COMPUTING CLASSROOM

As a new academic year begins, **Neil Rickus** shares guidance on managing behaviour in computing lessons

Pupils' behaviour is often cited as an area of concern for teachers, particularly those new to the profession, who might be working within a classroom environment for the first time. In this article we'll discuss the importance of managing behaviour, along with techniques to navigate your first lessons with a class, and look at how to address some of the unique challenges presented by the computing classroom.

So, why should we manage behaviour? Pupils make greater progress when they are in an environment where they feel safe and able to focus on their learning, which leads to lessons being more enjoyable for both pupils and teachers. Teachers who are able to manage behaviour effectively are also likely to exhibit improved personal well-being and stay in the profession for longer. Furthermore, the UK's Department for Education outlines how good behaviour helps minimise bullying (**helloworld.cc/behaviour1**), while Amanda Spielman, Ofsted's chief inspector, recently described how good behaviour ensures parents can be confident in the support provided by a school (**helloworld.cc/behaviour2**).

Initial lessons

During the first lesson with a class, classroom rights and rules can be agreed in conjunction with pupils. William Lau, in *Teaching Computing in Secondary Schools*, outlines how pupils have the right to learn and to be heard and respected. He discusses appropriate rules for the computing classroom, such as outlining the need to not wheel around on chairs or throw equipment around (**helloworld.cc/behaviour3**). As part of the first lesson, rewards and sanctions can be introduced, as well as a discussion of how these will be implemented during lessons. Seating plans can also be used from the first lesson, with pupils' prior attainment and learning needs used to determine their appropriate positions in the room.

The first lessons with a class are also a good time to establish the routines and procedures for the academic year ahead. While these should be in line with school policies, it is important that pupils are given ownership of how they are implemented, where feasible. Teachers should consider their context carefully, though: a small A level computer science group, for example, is likely to have different routines and procedures to a large lower-secondary class. Regardless of the approaches chosen, consistent implementation helps to ensure all pupils are aware of the teacher's expectations.

Many teachers will have an activity for pupils to undertake as soon as they enter the classroom. In the initial lessons, these activities can be used to find out more about the learners in the class. However, tasks should be carefully chosen to set the tone for the year. While it may be convenient to spend time setting up accounts or undertaking file management tasks, having pupils complete an open-ended project that is inclusive and promotes creativity can help show what an engaging and enjoyable subject computing can be.

> TEACHERS WHO MANAGE BEHAVIOUR EFFECTIVELY HAVE BETTER WELL-BEING AND STAY IN THE PROFESSION LONGER

Computing-specific challenges

The computing classroom presents a number of unique challenges to managing behaviour, which also need to be considered. As the class will often be seated in front of a computer, access to the device could be limited during teacher input or unplugged activities, which helps to keep pupils on the required task. Classroom management software can be installed to control machines, along with providing a range of other features to facilitate teaching (see the 'Classroom management software' box).

Pupils are likely to be accessing electronic resources regularly, so the process followed can form part of the class procedures and routines. Getting to the resources should be as simple as possible and follow the same steps each time, such as by using Google Classroom or Microsoft Teams. Consideration should also be given to how pupils will obtain physical items, such as printed resources and headphones, and how they will move around the classroom.

The reliability of technology means there will be occasions when a backup plan is needed. For example, offline resources might be required if the school's internet connection becomes unavailable, or paper copies of a presentation could be used if a room's display technology isn't working. A backup plan should therefore be carefully considered prior to each lesson and be readily available in the event of technology failure, which will enable it to be quickly put into place if needed.

Pupils having access to the internet in lessons can also lead to issues related to online safety. The teacher's expectations in this area can be reiterated each time pupils are using a digital device, to remind them of the appropriate behaviour.

Excellent computing teaching

It is often considered that teaching engaging lessons, using appropriate pedagogical approaches, leads to improved behaviour in class. It is therefore important that teachers consider the methods they are using to make the subject accessible, with suitable scaffolding employed where necessary to ensure an appropriate level of cognitive load. For example, teachers might consider using shared programming activities, or allowing pupils to develop open-ended, creative projects based on their own interests (**helloworld.cc/behaviour4**).

With regards to the content taught in lessons, the requirements of all learners and their social environments also need to be carefully considered. In particular, the need to make computing culturally relevant is increasingly being recognised as important for pupils' engagement — see page 54 for more on this. The more relevant we can make our lessons, the more likely we are to see pupils engaging with the subject.

How do you manage behaviour in the computing classroom? Have you successfully implemented other approaches? Do get in touch on Twitter **@computingchamps**.

CLASSROOM MANAGEMENT SOFTWARE

Classroom management software enables computers in a room to be controlled by the teacher, which can help with behaviour management. Along with limiting access to a device, the software often has many other features, including:

- Displaying lesson slides on a pupil's screen
- Monitoring pupil activities on a device
- Easy distribution of web-based content to machines
- Sharing pupils' work with the class

© vladwel/stock.adobe.com

NEIL RICKUS

Neil is a senior lecturer in computing education at the University of Hertfordshire, a primary education consultant for the BCS, and the founder of Computing Champions. He is a CAS community leader, a CEOP Ambassador, and a Raspberry Pi, Google, and Microsoft Certified Educator (**@computingchamps**).

THERE'S AN APP FOR THAT

In a world where there's an app for every possible need, this lesson starts a unit that aims to take learners from designer, to project manager, to developer, in order to create their own mobile app

AGE RANGE

11-14 years

REQUIREMENTS

- A web browser
- Student accounts on **Code.org**

This is the first lesson of a mobile app development unit that focuses on developing programming skills as well as other aspects of creating a digital product, such as user interface design, testing, user feedback, and teamwork.

Specifically, this lesson introduces students to the purpose of the unit: to create a mobile app. Some students will be raring to go, while others might feel daunted by the task ahead of them. To support both ends of this spectrum, the lesson aims to establish the need for decomposition and careful planning before any coding takes place.

Before delivering this lesson, you will need to create a teacher account using Code.org to generate login cards for your students. We recommend that you don't use the learners' real names, but use non-identifiable data instead, so that you're not sharing their data with any third party.

You can find instructions on setting up a class using Code.org at **helloworld.cc/codeorgaccount**. (HW)

OBJECTIVES

- ✓ Identify when a problem needs to be broken down
- ✓ Implement and customise GUI elements to meet the needs of the user

▶ The following lesson plan is taken from the 'Mobile app development' unit of the Teach Computing Curriculum (TCC). This unit is a programming unit that builds upon skills learners have developed using Scratch. It acts as a bridge between block- and text-based programming environments, as well as developing project management skills such as planning, design, feedback, and evaluation.

ABOUT THE TCC

Teach Computing

The Teach Computing Curriculum is part of England's National Centre for Computing Education and supports students aged 5-16.

Every unit of work in the TCC contains: a unit overview; a learning graph to show the progression of skills and concepts in a unit; and lesson content, including a lesson plan, slides, and formative assessment opportunities. Find them when you sign up for a free account at **ncce.io/tcc.**

ACTIVITY 1: APP FOR THAT 10 MINUTES

Introduce the lesson by playing a light-hearted video from Sesame Street (**helloworld.cc/sesamestreet**) in which the characters sing a song about apps.

Following the video, pair the learners and ask them to come up with ideas for apps that could change the world. Steer the class away from coming up with ideas for games without this purpose, or apps with the sole purpose of making money. Instead, for example, you could focus their minds on well-being, either for themselves or for others they know, with ideas such as a mindfulness colouring app, or an app that gives users a green score for how environmentally friendly they have been that week. The aim of the activity is to get students to start thinking creatively about apps, and what features and functionality might be possible using mobile devices.

Pick three of the ideas from your students and ask them to describe in more detail how their app would work. Prompt the students if necessary, by asking questions such as how they will get user input and how the app will store the data.

If time allows, ask the class to vote for their favourite app idea.

ACTIVITY 2: DECOMPOSE THE PROBLEM 5 MINUTES

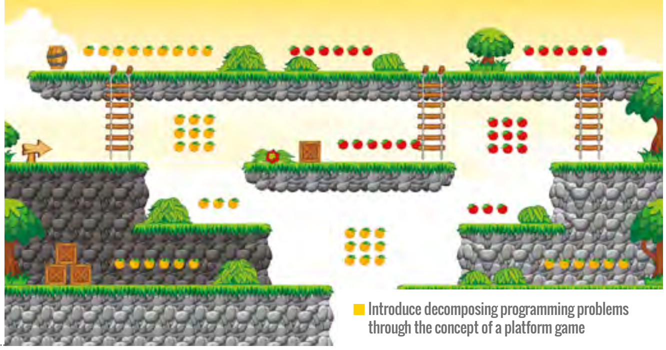
Using the concept of a platform game example, walk the students through how it could initially be decomposed, and highlight that by simply taking the problem and breaking it down into four steps, you can instantly make the project more manageable:

- Create the layout of the level
- Add a user-controlled character
- Add enemies
- Add the score, a timer, and power-ups

Explain that when you get to each decomposed step, the idea is then to decompose it further, to make the step even more achievable. For example, part of adding a user-controlled character involves adding movement to that character. Ask the students how they could decompose this step further. Take suggestions before revealing a suggested answer:

- Add left and right movement
- Add a gravity effect
- Add the ability to jump

Don't forget these are just suggestions, and that your students might have valid ideas that aren't on this list.



Introduce decomposing programming problems through the concept of a platform game

© aliafandi/stockadobe.com

ACTIVITY 3: DECOMPOSE TAPPY TAP APP 10 MINUTES

Introduce your students to the brand-new and exciting app that you're going to develop with them as a class over the next two lessons: Tappy Tap App. By working on this app, the students, together as a class, will learn many of the programming skills and development environment techniques that they will need to get started with developing their own app later in the unit.

The app will:

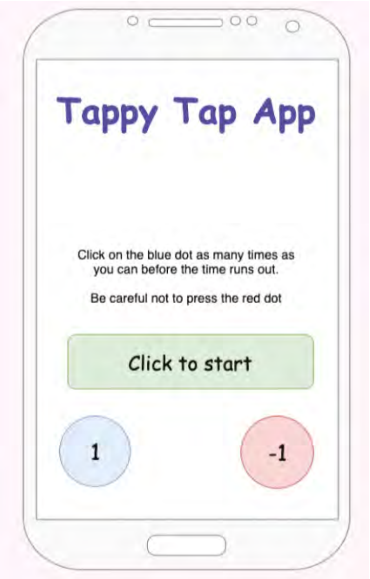
- Get users to click a moving blue dot as many times as they can within a time limit
- Give the user a point for each accurate click on the blue dot
- Move the blue dot each time it is clicked

Display the success criteria for the app on the board for the students, including the requirements of the welcome screen, game play screen, and final score screen. Let students know that you'll use success criteria at the end of a project to help judge its success. You can find a worksheet with more granular success criteria in the lesson files at **helloworld.cc/appforthat**.

Ask your students, on their own, to think about the requirements of the app and to decompose the task of building it. Ask them to write down their plan, ideally creating no more than five steps.

Once they have finished, ask them to pair up and share how they have decomposed the problem. Encourage them to make any changes they feel are necessary after the conversation with their partner.

Take answers from one pair. Write down their answers on a whiteboard and allow the class to discuss them briefly.



Tappy Tap App

Click on the blue dot as many times as you can before the time runs out.

Be careful not to press the red dot

RELEVANT LINKS

TCC 'Mobile app development' unit
helloworld.cc/appdevelopment

ACTIVITY 4: INTRODUCTION TO APP LAB 12 MINUTES

Introduce students to App Lab by showing a short intro video made by Code.org (**helloworld.cc/applabvideo**). This part of the lesson is simply to make sure all students are able to log in and become familiar with the App Lab environment. Distribute the login details to students and ask them to complete a short set of tasks, including:

- Log into App Lab using the details provided
- Browse to 'Start a new project'
- Select App Lab
- Rename your project 'Tappy Tap App'
- Add a button to the screen

It's not essential that students finish these tasks, but it will give you time to iron out any problems while they are working through them.

© David Fuentes/stock.adobe.com

ACTIVITY 5: DESIGN THE WELCOME SCREEN 15 MINUTES

Students are now going to spend time designing the welcome screen for Tappy Tap App, using the design tools in App Lab. This will give them time to become familiar with the environment and how to customise the properties of the screen.

They should use the success criteria from the worksheet, as well as the design on the board, to help them.

Some students might complete this task sooner than others. You could provide them with additional challenges, called 'explorer tasks' in the TCC, to work through to develop their skills further. For example:

- Add a new game screen
- Watch the following video (**helloworld.cc/applabvideo2**)
- Code the button to open the game screen

PLENARY ACTIVITY: PLAY THE GAME

5 MINUTES

If there is time, finish the lesson by giving your students a preview of a working version of the Tappy Tap App that they will be developing further in the next lesson (not shown here) at **helloworld.cc/tappytapapp**. Give them a few minutes to play, then ask the learners to think, while they are playing, about whether or not the game meets all the success criteria.

© imtmphoto/stock.adobe.com

Give students time to play the game!

SHIPWRECKED:
TURN YOUR RASPBERRY PI INTO A LIGHTHOUSE

Combine computing, tech, design, and art in this practical workshop, which brings together those less interested in tech with students who are knowledgeable in computing. Don't get shipwrecked!

AGE RANGE

Flexible age range: this workshop can easily be adapted for different age groups

REQUIREMENTS

- Raspberry Pi, any vintage
- SD card with Raspberry Pi OS
- Micro USB cable and network cable
- Keyboard, HDMI cable, and monitor (optional; if not running Raspberry Pi headless)
- Card, scissors, tape, and glue
- LEDs (helloworld.cc/leds)
- 330-ohm resistors (helloworld.cc/resistors)
- Jumper wires
- Hook-up wire
- Wire strippers

This workshop uses Raspberry Pi to drive a model lighthouse. We will make a cardboard model, wire an LED light, and program a flash sequence. Some may view lighthouses as ancient systems, but the processes involved and the underlying maths of the sequencing offer a challenge to young learners.

The whole build takes about two hours. You can pare down the build to a short lesson of assembling supplied modules, or you can run them as successive sessions over a number of weeks. Here, we focus on one session, but split the task across three teams: a construction team, an electronics team, and a programming team. The groups require different skills and levels of technical ability. Define your local objectives and tailor the project accordingly!

LEARNING OBJECTIVES

- I can interpret number patterns and recreate these through coding
- I can understand simple electronic circuits and wiring techniques
- I can understand how navigation safety is achieved through complex applications of a simple technology

ACTIVITY 1: THE HIDDEN CODE OF LIGHTHOUSES 20 MINUTES

You can run the first two activities with all students before splitting them into teams. A key feature of lighthouses is that they have a unique flash sequence to distinguish them from one another. This is an important safety element of maritime navigation. It can help to avoid confusion in poor conditions when you may not know where you are or which direction you are facing. A brief mention of lighthouse design history may help set the scene. You can look up local lighthouses to discover their light sequences and then tackle the challenge of recreating one in the code (**helloworld.cc/lighthouses**). Once you've decided on your sequence, you need to think about how you will code it. Will you make a flash, wait, make a flash, and repeat? Or have an array to dictate when the light is on, and then poll every second to see if this is an 'on' phase? Or use some other technique?

ACTIVITY 2: SETTING UP YOUR LIGHTHOUSE BRAIN 20 MINUTES

We will run Raspberry Pi headless, which simply means no keyboard or monitor is connected. Instead, we will control it through the network. If Raspberry Pis are in short supply, it is possible to set up sharing between coders using this method. By running an SSH server on Raspberry Pi, students and instructors can log in from a laptop or phone. Setting this up is beyond the scope of this article, but there are many helpful guides online (**helloworld.cc/remotepi**).

ACTIVITY 3: BUILD A TOWER **20 MINUTES**

A construction team will build the lighthouse tower. Secure a simple tall cylinder of cardboard to a cardboard base. On top, secure a short but wide piece of clear plastic tube to house the LED, topped with a cardboard hat. The construction may be decorated or painted.

© Eoin Houston

A basic cylinder of cardboard attached to a cardboard base

ACTIVITY 4: PUTTING THE LIGHT IN LIGHTHOUSE **30 MINUTES**

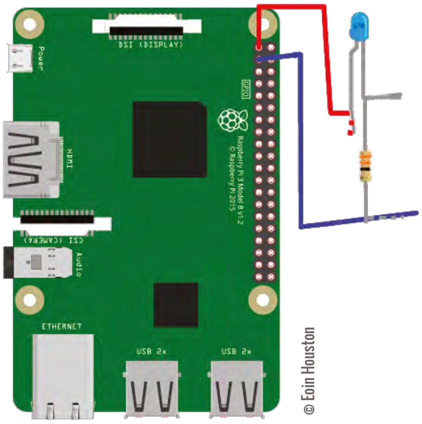
Let's get down to physical activities — assembling the components. This will be down to your electronics team. My favoured electronic construction technique is the 'ugly bug' method. This is where we simply twist wires and the component legs together to achieve a quick and functional circuit that does not require intricate and risky soldering skills. Breadboard construction would be neater, but space inside the model and simplicity of the circuit dictate otherwise.

For assembly, gather the electronics components on a clean flat surface with good lighting. Keep Raspberry Pi out of this assembly line for the time being. Best-practice wiring favours starting at the destination and working backwards, to avoid the chance of live wires shorting out. With a single LED, gently separate the leads, noting the longer lead (the positive lead) and starting with that. Cut a 20 cm piece of red hook-up wire and strip about 2 cm of plastic insulation from the end. Then twist the exposed copper wire and the lead together using your thumb and forefinger. Your younger students may need a little help here.

Next, twist together a small-value resistor to the shorter, negative LED lead and extend with a piece of black or blue wire. These join one after the other, or in series. For each of the hook-up wires, we can strip the other end and poke them into the end of the jumper wires and secure with a small bit of tape. Alternatively, chop a jumper wire in half and use the strip-and-twist technique to electrically join it onto the end of the hook-up wire.

We should now have an LED with long wires ending in pin connectors suitable for Raspberry Pi. A bit of tape around the joins on the positive wire helps prevent shorting out. Before connecting to Raspberry Pi and powering up, it's a good habit to do a bit of quality control. Sit back and look at the work. Does it look OK? Have the correct wires been twisted together? Is there tape on the required joins?

Once you have checked your LED wiring, connect the positive jumper wire to Raspberry Pi. We will use the GPIO (General-Purpose Input/Output) pins for this. These are the double row of 40 metal pins sticking up along one side. Finding the correct pin can be tricky, and a simple cardboard GPIO overlay helps greatly. The positive wire goes to the 3.3V pin. Next, connect the negative wire to GPIO02. Connect your network cable and USB power cable.



© Eoin Houston

Wiring diagram of the components

Before powering on, check that the LED wires are the correct way round, that you are using the proper pins, and that you have no electrical shorts, to avoid damaging your Raspberry Pi. Plug the big USB plug into the Raspberry Pi power supply. At this point, I keep an eye on it for ten seconds to watch the indicators on Raspberry Pi and ensure that no smoke appears!

ACTIVITY 5: PROGRAMMING YOUR FLASH SEQUENCE 10 MINUTES

There is scope for breaking up the programming activities into blocks and appointing a project manager to coordinate the different groups. The programming team will log in to the Raspberry Pi terminal, either remotely over the network or directly if you have a keyboard plugged in. The default username is 'pi' and the password is 'raspberry'.

Although this could be programmed in Python or Scratch (for example, **helloworld.cc/antenna**), I have chosen to write the logic in command line scripting known as Bash. This presents the opportunity to learn shell commands, UNIX-style constructs, and file systems, and there are no libraries or development environments to install and set up. I like to introduce the terminal initially, as it demonstrates the underlying technology and gives students an easy way to quickly make something tangible happen in the real world.

On older systems, we needed to set permissions to access the GPIO. However, in recent versions of Raspberry Pi OS, the default user 'pi' is a member of the GPIO group, meaning this is already taken care of.

Then we follow with a couple of commands to enable the relevant pin, set it for output, and then toggle it to a zero or low setting:

```
echo 2 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio2/direction
echo 0 > /sys/class/gpio/gpio2/value
```

If everything is working, the LED lights up! To switch off the LED, use this command:

```
echo 1 > /sys/class/gpio/gpio2/value
```

Here, there are opportunities to explore trickier concepts such as completing a circuit by applying zero volts and UNIX-like redirects, where at the command line we are directing the value 0 using the right-pointing chevron character to a file which represents a device (in this case, GPIO pin 02).

Making a lighthouse from Raspberry Pi can develop students' construction, electronics, and programming skills

© Alan Dunlop

ACTIVITY 6: AUTOMATING THE SYSTEM 30 MINUTES

All well and good, you say, but where is the proper scripting? This is where we automate the flash sequence. In this example, we flash twice in quick succession every 20 seconds. We are now ready to build on this and write some repeatable logic.

We need a text editor. Staying with the terminal, I use the nano default editor. If you want to use a GUI, any text editor will be fine. There is even an option to mount the Raspberry Pi file system remotely and use the text editor on your preferred device (**helloworld.cc/remote**).

If we use a tilde `~` on the command line, this is a shortcut to the current user's home directory. Making this a habit means the file will always be created in a suitable location, and the below command will subsequently work in most circumstances. Explore file-naming conventions if that is on your syllabus. I like a descriptive whole-word name that alludes to the purpose or function, camel case to avoid symbols and spaces, a number for labelling versions, and finally the optional .sh extension to indicate a shell script.

```
editor  ~/flashSequence1.sh
```

In the following script (in the next column), the initial line is a Linux quirk. This strange sequence of eleven characters is called a shebang. This is not part of the script, but tells the system what to do with the rest of the contents. In this case, it is saying it should be sent to the Bash interpreter. The lines beginning with a hash symbol # are ignored by the script interpreter. Hence, we use this character to mark human-readable comments just for the programmer to see.

For the repetitive logic, I have introduced an endless loop — in this case, using a 'while' loop. The initial line beginning with that word is a question or test. The bit following the word 'while' can be a comparison or evaluation — if this results in something true, then the following block of code is carried out. The block is denoted by the words 'do' and 'done'. In this case, as I have typed the word 'true', it will always trigger, endlessly.

Within my loop I am using the echo command. It sends the supplied bit of text or string to a file. If it were a normal text file, we could subsequently view it and see that it now contained our string. However, in this case, we discover something a little bit weird about Linux file systems. Special files represent real-world devices, hardware, streams, and numerous other things. This enables us to interact with them, sending information to devices, or getting data from them. Note the second 'sleep'. Without this, the loop would instantly flip back to the beginning to switch the LED on again, and we would never see the fraction of a millisecond the LED is off.

```
#!/bin/bash
#  A script to flash an LED attached to the
GPIO
# written by Eoin Houston
#  v1 29JUN21
#  Comments are king!
#  enable the pin and make it an output
echo 2 >  /sys/class/gpio/export
echo out > /sys/class/gpio/gpio2/direction
# introduce a never-ending loop to
activate the pin
# and after a delay disable the pin
while true
    do
    echo 0 > /sys/class/gpio/gpio2/value
    sleep 1
    echo 1 > /sys/class/gpio/gpio2/value
    sleep 1
    done
```

We save our code creation as a text file. By default, new files on UNIX systems cannot be run; this is a security feature to limit malicious activities. We must therefore mark it as executable with a chmod command. The second line is us actually running the script. After all that effort, we will hopefully see the LED merrily flashing away.

```
chmod  +x  ~/flashSequence1.sh
~/flashSequence1.sh
```

As we wrote a continuous loop, this will run until the end of time. To stop it, we press Ctrl + C on the keyboard.

© Eoin Houston

A finished model

CONCLUSION: NO SOS

I hope this has worked out for your students. If they have a recognisable flash sequence, they are saving people's lives, and no one will be sending out an SOS! This guide is by no means comprehensive; there are many more aspects to explore. It is intended as a primer for you to develop for your local needs and as an accessible mechanism for introducing a wide range of topics. (HW)

EOIN HOUSTON

Eoin is a backstreet mechanic of the IT world. He is a fixologist, hacker, and technology enthusiast.

USING MACHINE LEARNING TO RECOMMEND A MOVIE

Recreate the intelligent systems used on streaming platforms to recommend movies and TV shows, using Machine Learning for Kids and your own data

AGE RANGE

11-14 years

REQUIREMENTS

- A web browser
- An online form creator (e.g. Google Forms or Microsoft Forms)

MAC BOWLEY
Mac is a learning manager at the Raspberry Pi Foundation. When he isn't teaching, you can usually find him tinkering with his latest project (**@Mac_Bowley**).

Machine Learning for Kids is an online platform that allows you and your students to create and train your own machine learning models. This activity combines data science and machine learning skills to recreate a part of the recommendation algorithms used on sites such as Netflix and YouTube. (HW)

RELEVANT LINKS

- Machine Learning for Kids: **machinelearningforkids.co.uk**
- Example survey: **helloworld.cc/moviesurvey**
- Example data set (download): **helloworld.cc/moviedata**

ACTIVITY 1: CREATING YOUR SURVEY 10-15 MINUTES

First, you will create a survey with your class. You can use a service such as Google Forms or a paper-based table. The survey you make needs to collect six different pieces of information from the users:

- Their favourite genre of movie, from these four options:
  1. Action
  2. Comedy
  3. Fantasy
  4. Mystery
- Their ratings for five movies you have chosen, on a scale of 1 (hate it) to 10 (love it)

Have a class discussion in which you explain the aim of the project and get students to pick the movies you will use. This is a great opportunity for your class to bring their backgrounds and experiences into the project. Your survey should contain:

©sitthiphong/stock.adobe.com

- One multiple-choice question for participants to choose their favourite genre
- Five scale questions (ranging from 1 to 10) for them to rate the movies

Now get as many people to take the survey as possible, perhaps asking multiple classes to fill it out.

You can find an example survey at **helloworld.cc/moviesurvey** and a corresponding example data set to download at **helloworld.cc/moviedata**.

ACTIVITY 2: CLEANSING AND SPLITTING THE DATA 10 MINUTES

You may find that survey participants leave a field blank if they haven't seen a movie. I purposefully did not provide learners with instructions for what to do in this case, because it offers an opportunity for a class discussion about what to do with those fields.

There are three options for you to consider. First, if you have plenty of data, you can remove any sets of reviews with empty fields. If someone has not seen one of the movies, there is no way to tell what their rating would be.

If you don't have a huge amount of data, there are other options. You can change any empty review fields to a 0 or -1. The rating scale goes from 1 to 10, so 0 is not used for anything. In essence, this would nullify the impact that field has on the final prediction. If you set them as a -1, then that review would have a negative impact on the final outcome. This option assumes that because someone hasn't seen a movie, they must not like it. You and your class have to decide how to deal with these fields through discussion. In my example data, I changed empty fields to a 0.

The final step of data preparation is to split your data into training and test sets. Remove one or two sets of reviews (a row in the example spreadsheet) corresponding to each genre and save them separately to use for testing later.

ACTIVITY 3: MAKING YOUR MACHINE LEARNING PROJECT 10 MINUTES

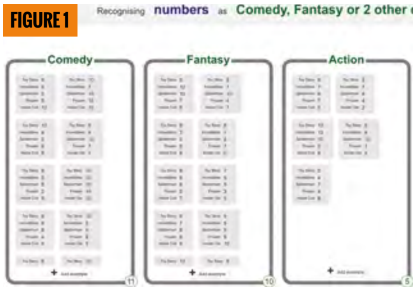
For this project, you are going to use the browser-based machine learning platform Machine Learning for Kids (**machinelearningforkids.co.uk**). You can make a free account, allowing you to have three machine learning projects and manage a class. Alternatively, you can use the 'Try it now' option to get access to one project for three hours before it is deleted.

Once logged in, head to 'Projects' in the top menu and click on '+ Add a new project'. Give your project a name and choose 'numbers' (i.e., the ratings for each of the movies) from the 'Recognising' drop-down list.

To finish the set-up, you need to add the movies to your project as values. You can add these by clicking the blue 'Add another value' button. Use the movie's title as the name for your value and set them all as 'number' (see **Figure 1**).

If you made an account and want to do this as a whole-class project, you can check the box in the top right. This will allow you to add your students to the project so everyone can add some data to it. Either way, click 'Create' at the bottom to finish up this stage.

FIGURE 1



Recognising **numbers** as **Comedy, Fantasy or 2 other**

Comedy

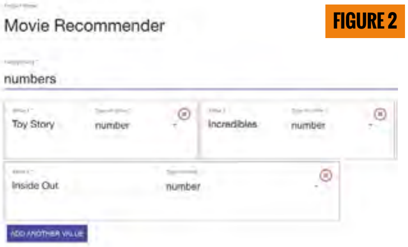
Fantasy

Action

Entering your data

Next you need to add some labels to your project. Head into the 'Train' screen, click the '+ Add new label' button in the top right, and enter one of the movie genres you selected earlier. Repeat until you have a label for each genre. These labels are what the algorithm is learning to output.

FIGURE 2



Movie Recommender

numbers

Toy Story number

Incredibles number

Inside Out number

Now it is time to enter your data! In the example data set, I sorted the data by genre so that all the sets of data for each genre were grouped to make it easier to enter the data.

For each set of responses, check which genre the participant has chosen and click 'Add example' inside the label corresponding to that genre. A pop-up will appear for you to enter the ratings for each movie (see **Figure 2**).

ACTIVITY 4: TRAINING AND TESTING 10 MINUTES

Once you've entered all the data, click 'Back to project' in the top left, and then 'Learn & Test' to go to the training page. Click the 'Train new machine learning model' button to create a new model for your project that has learnt from your training data.

Once the training is complete, some boxes will appear that allow you to test your model by inputting some test data. Use the four examples you saved earlier and check whether the model correctly identifies the genre from the review scores.

Wrap up this activity by discussing with the class what other elements Netflix and YouTube use with their recommendation algorithms. Now that you know someone's favourite genre of movie, how would your students make sure the next movie the person watched was one they liked?

*This activity is taken from the Raspberry Pi Foundation's brand new online course 'Introduction to machine learning and AI'. If you want more great activities like this, check out the course at **helloworld.cc/machinelearningcourse**.*

CODE CLUB AND THE CINEMA

Gemma Coleman catches up with **Ali Alzubaidy** and **Mustafa Khalid** about their innovative use of cinemas for their Code Club meets in Iraq

What inspired you to use cinemas for Code Club?

Cinemas have many features that attract people to watch movies. Among them are the large screen, good sound and lighting, and comfortable chairs. The other special thing is eating [pop]corn during the film. These sounded like the perfect ingredients for a successful and engaging Code Club!

What are the benefits of using cinemas for Code Clubs?

Along with the above benefits of sound and lighting and a big screen to project tasks onto, there are plastic barriers between the seats, and this achieves social distancing during the coronavirus crisis.

Another important point we would like to highlight is that not all parents have the financial ability to take their kids to the cinema. This was an important point that made us think about organising our event in the cinema, to help as many kids as possible have a cinematic experience, while also learning coding for free.

What feedback have you had from the children?

In our first workshop in the cinema hall, the members of the Code Club asked, "Which movie will we watch today?" We told them it was a surprise, and when we displayed on the big screen that we were here to learn to code, all the members said, "It is an amazing idea!" They liked lots of things, like the comfortable chairs, the hall sound, and the big screen, and they loved that after finishing your activity, you get to eat your [pop]corn!

How do you get the cinemas to let you use their space for Code Club?

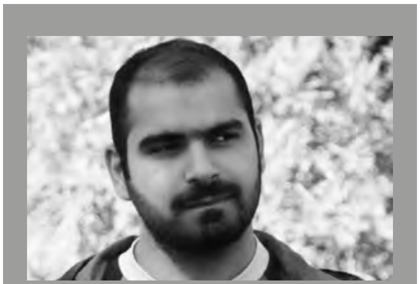
After organising a meeting with the cinema administration, we presented a summary about MyCodeClub's target, which is to turn cinema halls in Iraq into learning halls during their non-screening time. We explained that cinemas are a great environment in which to teach young people programming, because of the good effects on their logical thinking. They agreed to support our Code Club activities, providing the VIP hall for free every Friday from 9am to 12pm, in addition to providing free fruit, drinks, and [pop]corn for MyCodeClub members.

Would you recommend other Code Clubs try this out?

I recommend that volunteers always look for new things to keep their Code Club active. Cinema halls encourage young people to communicate and continue to learn coding and other activities because of the cinema's engaging side effects. (HW)

ALI ALZUBAIDY

Ali is an educator at the Ministry of Higher Education in Baghdad, Iraq. He is the founder of the Coding for Kids initiative (**coding4kids.org**), focusing on teaching kids STEAM subjects and coding in a fun way. He is also a Raspberry Pi volunteer on the translation team (**@Ali_Alzubaidy85**).



MUSTAFA KHALID

Mustafa is from Baghdad and is a PhD student in AI in China. He is the co-founder of Coding for Kids, a Code Club volunteer, and an IT company manager at AiNi (Artificial intelligence for Natural insights) (**@mustafakhalid90**).

Students can choose where they'd like to learn in the classroom

ME AND MY CLASSROOM

Tonya Coats shares how she creates engaging and collaborative spaces in her classroom, giving her students choices about how they learn

Adults need autonomy to be productive and to bring new ideas to life. The same can be said about students in the classroom. Learners can't be creative and innovative in an environment where they have no control. My goal for my second-grade classroom is to empower my students to have some degree of choice and plenty of opportunities to collaborate with other students. It's important for our youngest learners to have the opportunity to be comfortable while learning.

My classroom is composed of a variety of spaces so that students can choose how they want to learn. When I teach a lesson, some students prefer to use a low chair to recline while listening, while other students prefer to bounce. Balance balls, stools, pillows, rugs, low tables, and traditional desks and chairs enable students to pick what works best for them. No two students are the same, so having an assortment of seating that provides quiet body movement and comfort, to best suit their individual learning needs, is key. How can we expect students to solve problems and make choices if we are constantly making choices for them?

Having a space for gardening has also helped my class have organic positive social interactions and a way to decompress. Our class loves the responsibility of caring for plants and the opportunity to learn about the outdoor environment from inside.

I want my classroom environment to be open to communication, creativity, and critical thinking. This can't be done sitting still at the same desk all day long. Adapting to students' different learning styles will enhance their educational experience and make their learning meaningful. (HW)

TONYA COATS

Tonya has been a teacher for 20 years. She teaches at Rustic Lane Elementary in Jurupa Valley, California, USA. She is passionate about edtech and inspiring innovation. Her enthusiastic students always arrive prepared to learn and meet her school's high expectations.

It is impossible to overstate the learning opportunities that building robots can bring your students

©stock.adobe.com/GREKCO

OUR FIRST ROBOTICS COMPETITION

Diane Dowling shares the excitement and energy of her and her students' first robotics competition, the skills learnt, and the challenges met along the way

I had been teaching computer science for around ten years before I discovered robots. One of my students returned from a university taster day to inform me that she had seen something she was sure I would like. It was a poster advertising a free robotics competition for teams of sixth form students. The organisers — Student Robotics — would provide the core electronics kit, and all we had to do was design and build a robot to meet the set challenge (**studentrobotics.org**). The robot we would build had to operate completely autonomously for three minutes, with no human intervention and no remote control. This meant that when we turned on the robot, it would do only what we had programmed it to do, nothing more and nothing less. From that day on, my students and I were hooked!

The challenge

The challenge involved collecting and moving a set of large coloured boxes into zones marked on the floor of a 4 m^2 arena. Four teams' robots would compete at any one time and, at the end of three minutes, each zone would be won — and points scored — by the robot that had deposited the most of their boxes into it. To increase the level of challenge, each zone also contained a pedestal and, if the robot could place one of its boxes on the pedestal, it would automatically win the zone. Each team had free reign in the physical design of their robot. The only constraints were that it had to be less than 45 cm^3 and with no sharp protruding parts that could damage another robot.

The great thing about getting students to build robots is that the team needs a very wide range of skills. The first task was to make a basic chassis (base frame) and get it to move. The team started with a square piece of wood and planned to mobilise it by a set of motor-driven wheels. Working out which motors would do the job was the first of many physics and mathematical challenges, and I was delighted that the classroom whiteboard was appropriated for many sets of calculations and design work. This was real, applied learning and the students threw themselves into it, discussing which formulae to use and checking each other's work.

Eventually, motors and wheels were specified, purchased, and unpackaged with great excitement. The next thing that the team needed was some basic carpentry skills. At this stage, it became obvious that none of them had any! We bought a few tools and a workbench, and the classroom took on the air of a maker shed every Wednesday afternoon. Getting the wheels on straight proved an enormous challenge and would haunt us throughout the competition. However, by the end of the first month, the team had an open wooden box with two wheels, two motors, and a stabilising ball castor at the rear. The motor boards, power board, and brain board were connected on a Token Ring network, and all of these components and the battery block were secured to the chassis with hook and loop fasteners. The motors were wired up to the motor boards and the robot was ready for some code!

> “ EVERY TEAM MEMBER DECLARED IT WAS THE BEST THING THEY HAD EVER DONE

Student Robotics had provided an integrated development environment (IDE) and core software libraries to interface with. The team's own code had to be written in Python and then downloaded to a USB flash memory stick and inserted into the brain board. Powering the board and turning it on would automatically run the code, and any runtime errors were logged so we could scrutinise them later. I can still vividly remember the moment when the robot was powered on with a simple script for the first time. The lights flashed, there was a long pause, and then the robot shot across the classroom and crashed into the wall. The team was ecstatic — the robot could move!

Designing a good robot requires a lot of engineering skills, and these are rarely taught in the UK outside of university. The task proved an enormous challenge to the team's design and build skills. However, by Easter, the robot had grown and sported a set of wonky arms that could be mobilised by an actuator driving some string around a cotton reel. It was competition time!

Game day

The competition (pre-Covid) is played over a single weekend. Each team has a workstation and is given some tinker time; then comes a series of league rounds that determine the seedings for the knockout rounds on the afternoon of day two, culminating in the final between the top four teams. In between rounds, the robots can be tweaked and code updated as necessary.

My team was very consistent. We consistently gained a point for moving and, with refined code, eventually got to the point where the robot could reliably move to one of its boxes, scoop it with a wobbly arm, and swish it into a zone. Even now, I can remember the buzz of excitement and the sense of achievement every time we gained a point. We didn't come anywhere close to winning, but every team member declared it was the best thing they had ever done. The atmosphere, camaraderie, and adrenaline were palpable.

Over the years, the college team got better and better. Every Year 12 student got the chance to compete again in Year 13, and they were able to mentor and transfer skills from year to year. In 2017, the college team reached the finals and was selected in a play-off to represent the UK at the *FIRST* Global Robotics olympiad in Washington, USA! I am now a trustee of Students Robotics and many of my past students are volunteers in the organisation; robotics has changed all of our lives.
It is impossible to overstate the real learning opportunities that building robots can bring your students, from creative design skills, to coding, maths, physics, engineering, and teamwork skills. And you don't need expensive equipment to get started. A Raspberry Pi computer is totally adequate to power a small robot, and the Raspberry Pi Foundation has a free course that takes you through building a small buggy and the fundamentals of robotic control (**helloworld.cc/buggycourse**). You could easily stage your own simple competition in the classroom, with points for completing tasks such as moving, turning, line-following, and obstacle avoidance. Encouraging like-minded students to work in teams can be very challenging and may require some careful management at first. But in my experience, the shared goal will promote genuine team work and allow communication skills to develop in line with technical prowess. (HW)

ROBOTIC COMPETITONS TO TRY

- Student Robotics (**studentrobotics.org**)
- *FIRST*® Tech Challenge (**helloworld.cc/firstuk**)
- VEX Robotics Competitions (**helloworld.cc/vexcomp**)
- *FIRST*® LEGO® League (**helloworld.cc/legoleague**)

DIANE DOWLING

Diane is a learning manager at the Raspberry Pi Foundation, where she works on the Isaac Computer Science platform. In her spare time, she is a trustee of a national charity that runs robotics events for sixth-formers.

As a curriculum team, we would plan for everyday, commonplace scenarios in which a teacher might be absent at short notice

PLANNING FOR TEACHER ABSENCES

Alan O'Donohoe shares practical strategies and tips for keeping your class working when you have to be away from the classroom

One evening you receive a phone call from a member of your school leadership team, informing you of the dreadful news that a teaching colleague in your subject team has passed away suddenly and without warning. Instantly, a thousand thoughts rush through your mind, from concern for your colleague's family and loved ones, to the practical implications for their teaching groups.

This is the set of circumstances I found myself in just a few years ago, as head of computing, on hearing of the sad and unexpected loss of my friend and colleague. Accompanying the shock, while coming to terms with the impact of his sudden death, was the knowledge that I needed to quickly put plans in place for the next few days. Thankfully, as a curriculum team, we'd planned for everyday, commonplace scenarios in which a teacher might be absent at short notice. This bought us some planning time in the short term, but we needed to plan for the longer term and the uncertainties that might have ensued.

The purpose of this guide

Although extreme events like these are thankfully rare, there is a tremendous amount of value and comfort to be gained from planning for unexpected teacher absences. The experience I described is very much a nightmare scenario and one that nobody wants to imagine themselves in — but it's not an impossibility. The advice and recommendations in this article will help you prepare for a range of circumstances, from the more mundane to the totally unexpected, and help ensure that all your classes can still make learning progress when their computing teacher is absent.

In this instalment of the Insider's Guide, I recommend two long-term strategies that do require an initial investment of time. Rest assured, though, that these will repay you on those occasions when time or circumstances do not allow you to prepare more carefully considered work for your classes.

Recorded videos that you may have prepared during the pandemic could come in handy for cover lessons

> “ THE TIME WE INVESTED IN PLANNING PAID OFF MANY TIMES AFTERWARDS

Emergency rations

Prepare a one-size-fits-all pack of paper-based resources that can be used at short notice in any standard classroom, with any class. It’s far from ideal, but will provide a valuable contingency plan that can be used short-term in any situation.

When emergency building work took our ICT rooms out of service for an indefinite period of time, we were issued with a complicated timetable of replacement classrooms around the school site. Relocating ourselves and our classes intact to unfamiliar classrooms was challenging enough without the added difficulty of also having to plan lesson activities that could be completed in classrooms with no access to computers or the network.

When we originally resourced these packs, we commandeered sets of textbooks and revision materials that were no longer relevant to our present curriculum. We included a printed set of basic instructions for staff and students, for example, “Start with the first article. Make notes using your own words, and illustrate your notes. Answer the questions. Move to the next article.” We also included packs of paper and budget pens and stored all these in a box clearly labelled ‘Emergency Rations’ near the teacher’s desk.

The time that we invested developing these resource packs paid off many times afterwards. One morning, when a team member was involved in a traffic accident, he was able to ring the school reception and pass on instructions for students and the cover teacher to use the Emergency Rations until he arrived.

Side projects

In addition to teacher-directed learning activity in your regular lessons, I recommend that students also work on an extended parallel project they have chosen, in which they have a stronger sense of autonomy. These student projects typically last a whole year and incorporate a high degree of flexibility, creativity, and collaboration, with no requirement for formal assessment. The most popular student projects in my experience have been web development, games design, and multimedia projects, but have also included robotics and software development.

To maintain the highest levels of interest and motivation among students, and commitment to their side projects, a portion of curriculum time is made available in which they develop their projects, as well as acquire the necessary knowledge and skills they can use in and out of lessons.

These side projects really come into their own when a teacher is absent and students can work on their side projects rather than other tasks. Since students are heavily invested in their projects, it means fewer classroom management incidents for the supervising teachers. When I’ve asked the cover supervisor afterwards for

HELPFUL RESOURCES

- **helloworld.cc/coverworkexample:** This is an example of some cover work I set in advance for a colleague's class. The content can be used with different classes and the approach modified to suit different themes.
- **helloworld.cc/covervideos:** Teacher Sam Wickins has recorded a comprehensive collection of videos for those occasions when he knows he can't be in front of students.

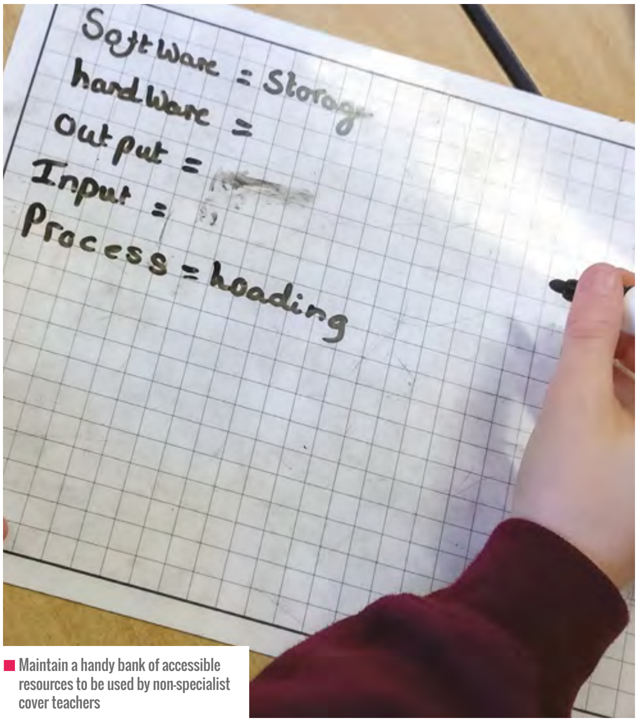
feedback, they've been very positive, telling me that the class just came in and got on with their projects with a minimum amount of fuss and no need for intervention. This also meant that there were no behaviour management issues that needed attention.

Practical advice

To complement these long-term strategies, here are a number of practical tips and tricks from the computing teaching community:

Imagine you're the cover teacher: Consider what information you would need if you were covering a lesson for a colleague in another subject. For example, keep a handy list of trusted colleagues available, along with their teaching bases, so that a cover teacher can refer to them if there are any issues.

Treasure your cover supervisors: If you develop and maintain positive relationships with these colleagues by ensuring they are adequately prepared, resourced, and valued, it will pay off. Aim to gather feedback from these colleagues afterwards about the quality and challenge of the work.



Maintain a handy bank of accessible resources to be used by non-specialist cover teachers

Share up-to-date seating plans: Cover teachers say that accurate seating plans are enormously useful in helping them manage classroom behaviour. One trick I used for occasions with an unfamiliar class was to draw a simple outline map of the classroom, then discreetly ask a student to complete the seating plan for me. Some schools require all teachers to maintain seating plans for all their classes, with copies kept centrally and in the classroom. Specialist software such as Class Charts, ClassDojo, and iDoceo can make life easier for the class teacher to monitor, plan, track, and manage behaviour. If you use one of these, consider how to share it with a stand-in teacher, by exporting or sharing it.

> “ MOST STUDENTS BENEFIT FROM NOT HAVING THE TEACHER TO FALL BACK ON

Maintain a bank of accessible activities: Computing teacher Khawer Ishtiaq keeps a bank of activities for last-minute absences, all linked to the curriculum for each unit of work: “We try to keep the tasks quite straightforward in anticipation of non-specialist teachers. We also have a bank of ‘Computing in the News’ activities.”

Keep a list of planned topics: Teacher and community leader of Computing at School York, Pete Dring, doesn't like asking colleagues who are too ill to come in to have to prepare cover work, but admits it can be hard to cobble something worthwhile together on their behalf at the last minute. Pete finds the following approach helpful: “Everyone in the department has a shared spreadsheet where we map out (very roughly) what we plan to teach each lesson for a half term, with resources shared centrally. In theory, that means when I get a last-minute email from the cover supervisor, it's a simple job to print off/organise the instructions for the cover teacher.”

Share cover work with students: Make use of your learning management system, whether it be MS Teams, Moodle, or Google Classroom, to ensure that the work you planned is accessible to all your students. If you establish this as the norm, your students will come to expect it. You'll be able to include more detail in the instructions, and clickable links to relevant resources. If your seating plan is included as part of this for all to see, there are likely to be fewer arguments later on.

Alert your colleagues: You will have already reported your unplanned absence to school, but if you can find a way to inform your colleagues as well, they may be able to check in on the lessons to see if everything is running to plan. For a planned absence, you can let your colleagues know where to find the work you've set, and any other arrangements in place, in case they are called upon to assist.

Practise password recovery: Computing teacher and author Alan Harrison recommends training all staff in the password reset process for the websites and resources they use, long before they

Students could work on extended side projects if you're away

need to cover your lesson! Alan explains: "It's also worth training the students in self-service password reset for any services they use, e.g. Quizlet, Seneca … Too many cover lessons come back with a note saying "I did what I could, but ten students forgot their passwords."

Record demonstration videos: Computing teacher Raihaan Chaudary created videos for remote learning during school lockdown, and later discovered that they also worked well for cover lessons. "I created videos using Screencastify [screen recording software] to demonstrate how to complete tasks in detail. These were uploaded to YouTube to allow for easy access for students. I have also used Edpuzzle, enabling me to pause the video at certain times for students and ask a question, which can be a self-marked MCQ [multiple-choice question], or free text to be marked by myself. Edpuzzle also allowed me to monitor which students had watched the videos and how much of the video they had watched."

Hour of Code: Computing teacher Emma Spreadbury has found that free online resources such as the Hour of Code work very well for lower-secondary classes, particularly those one-off lessons in which students are at a tricky bit in a project or need more input before moving on (**hourofcode.com**). For older students, she's pointed them towards a bank of programming challenges she's collected and shared via MS Teams so students can quickly find them. Emma recommends: "Preloading work (both in terms of resources and student awareness) before an absence has always proved incredibly useful. Simply described, students should be in a position where they can self-manage with direction from me. I can inform the cover teacher that the students know what they're doing. I genuinely think most students benefit from not having the teacher to fall back on or feel like they are expected to produce something significant by the end of the lesson."

Avoid activities that rely on additional resources: It might seem straightforward to ask students to watch online learning videos, but will they need to use headphones? What if the headphones are locked away? The next time you lay eyes on those headphones, will you regret setting that work? It's not ideal, but perhaps easier to manage if the whole class is watching one video at the same time.

Being away from the classroom is never ideal, but having a sturdy backup plan in place can make those unexpected moments in life a little less stressful, and keep students' learning on-track.

This article is dedicated to the memory of Mark Greenwood (1962–2014). Mark was a visionary leader, a champion for education technology, and an outstanding computing teacher who encouraged his colleagues to focus on solutions, not problems. (HW)

ALAN O'DONOHOE

Alan has over 20 years' experience teaching and leading technology, ICT, and computing in schools in England. He runs **exa.foundation**, delivering professional development to engage digital makers, supporting computing teaching, and promoting the appropriate use of technology (**@teknoteacher**).

THE BEBRAS PUZZLE PAGE

Each issue, **Chris Roffey** shares a computational thinking problem for your students

DOMAIN

Algorithms and programming

SKILLS

Generalisation, pattern recognition

AGE

6-12 years

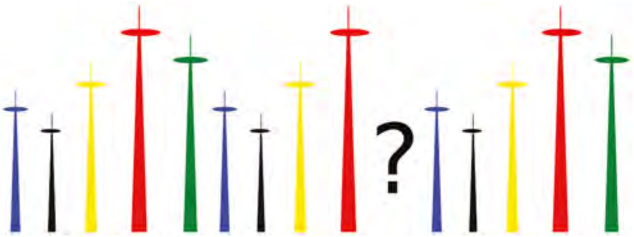
DIFFICULTY RATING

Ages 6-8 hard; ages 8-12 easy

THE PROBLEM: MISSING PICTURE

Pattern recognition

In the picture to the right, a pattern is repeated more than once. What is the missing picture? ***Solution on page 97.***



Further information

Pattern recognition is an important skill for computer scientists. Programmers, for example, often look for repeating patterns in problems so that they can use loops, functions, or recursive solutions in their programs. This problem is an easy one, aimed at younger students, who should be able to identify the pattern that repeats exactly three times.

COMPUTING KEYWORD SPOTLIGHT: PATTERN RECOGNITION

Defining everyday techniques used by problem-solvers

At first, it can seem an easy task to write pattern recognition problems for your students. However, when these kinds of tasks are proposed to the international Bebras community, they often get rejected as 'banana questions'.

Consider this question:

What comes next in the following sequence? 2 4 6 8

The obvious answer is 10. However, any answer is possible, even 'banana'. The pattern is not defined properly, and could be:

2 4 6 8 banana 2 4 6 8 banana 2 4 6 8 …

or perhaps:

2 4 6 8 banana 8 6 4 2 banana 2 4 6 8 banana 8 6 4 2 …

… or any one of an infinite number of other sequences!

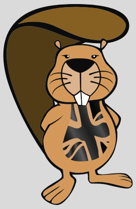
Professor J.P. Pretti from the University of Waterloo and the Canadian Bebras Team wrote the question above as an illustration of the dangers of writing questions that a 'reasonable person' would get right, or that rely on common sense.

A first draft of this issue's challenge might have been worded: 'What does the missing picture look like?' If we worded the task in this looser way, a student applying common sense would say, 'a green tower'. A creative student might say, 'a banana'. A computer would say, 'This is not possible to compute.'

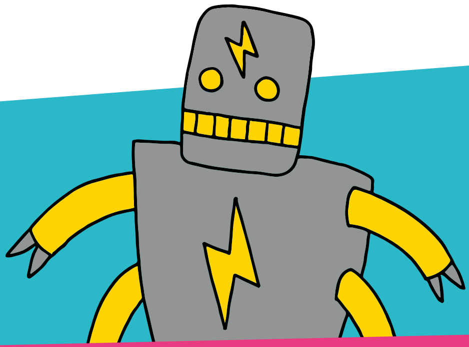
Computer science is about precision, and we must give credit to correct answers to questions, even if they are not what we expected or wanted. So how do we avoid writing banana questions? When checking them, we should ask ourselves, 'Does this question have a precise answer, or am I expecting them to guess what I am thinking?' Very often, the question can be fixed by rewording it more carefully, as we have done in this issue's Bebras Puzzle.

ABOUT BEBRAS

Bebras is organised in over 55 countries and aims to get students excited about computing and computational thinking. Over the last two years there have been a quarter of a million participants per year in our annual challenge. Our archived questions let you create your own auto-marking quizzes at any time during the year. To find out more and to register your school, head to **bebras.uk**.



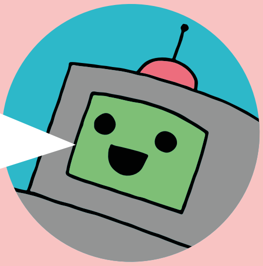
{code club}



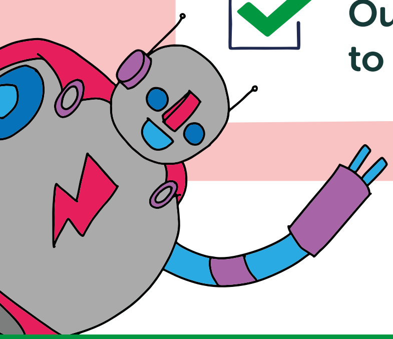
START A CODE CLUB IN YOUR SCHOOL!

It's easy to get your school coding! Code Club provides everything you need to run coding clubs for 9- to 13-year-olds with free, step-by-step project guides for learning Scratch, Python, and HTML/CSS.

> "I run a Code Club in our school for children in Year 5 and 6, which is always popular. Code Club activities encourage learning independence, and the children love to share the games they've coded with each other."
>
> **Jill, Teacher**



- ☑ Code Club is free
- ☑ Code Club offers engaging, hands-on activities that have been designed to spark curiosity and inspire creative thinking
- ☑ Our free online training is perfect for teachers who are new to coding or are starting a Code Club for the first time



Join Code Clubs across the world giving young people opportunities to have fun while learning to code!

Get involved at codeclub.org

Code Club is part of the Raspberry Pi Foundation (registered charity no. 1129409)

HOW TO TEACH COMPUTER SCIENCE: PARABLE, PRACTICE AND PEDAGOGY

What has Lord Byron got to do with computer science? Alan J. Harrison's book is full of delightful nuggets of information that will bring your teaching to life

INFO **BY** Alan J. Harrison | **PUBLISHER** John Catt Educational | **PRICE** $20.74 | **ISBN** 9781913622572 | **URL** helloworld.cc/parable

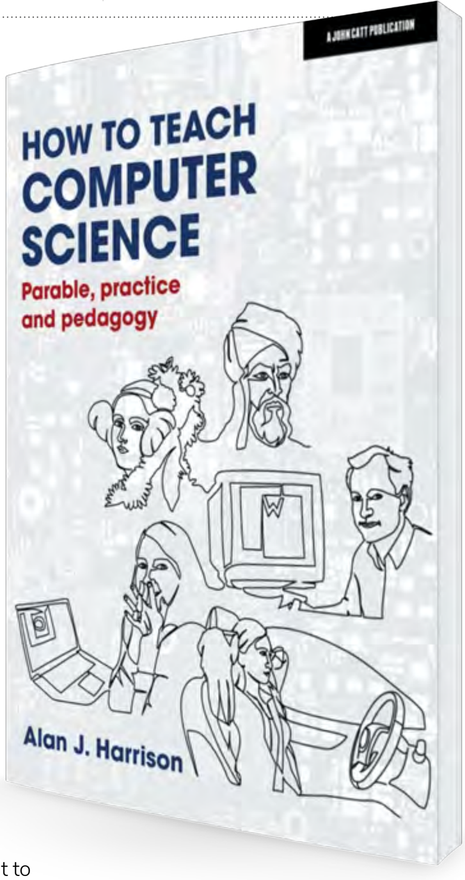
Diane Dowling

This little book is a must-have for every teacher of computer science. Although it is structured around the core content for England's GCSE, there is something here for every computer science teacher, at every learning stage and curriculum. In particular, this book has a truly awesome collection of journeys into the history of computer science.

I was gripped from the first page, which dropped me into Switzerland in 1816 with a poem penned by Byron to his daughter Ada, who is usually acclaimed as the world's first computer programmer. We learn why Ada's mother, Annabella Milbanke, pushed Ada to learn science and mathematics and how, as a young adult, Ada was fascinated by the potential of Charles Babbage's difference engine. We learn how she envisaged how this mechanical calculator might be able to manipulate sound and images in the way that it "weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves".

For me, as an A-level specialist, these well-researched and well-written journeys into the past, and some wonderful images, are sufficient reasons to recommend the book.

For GCSE teachers and those teaching computer science at an equivalent level, the book also provides a set of useful checklists and material for planning lessons. Each chapter briefly explores topic-specific pedagogical content knowledge through core concepts, fertile questions, higher-order thinking, analogies and concrete examples, cross-topic and cross-curricular learning, unplugged activities, physical computing, project work, and misconceptions. In addition, it covers general pedagogical principles such as flipped learning in the introduction, together with advice on inclusion and issues relating to gender balance. There are also recommendations for further reading and references throughout to resources and supporting materials, which will be very useful to new teachers and those wishing to improve their subject knowledge.



The book is titled *How to teach computer science*. In the narrow sense, I am not convinced that it provides an answer to this question, in that this is not an instruction manual on how to go about planning a scheme of work or a set of lesson plans. However, taken holistically, the book does advise on how to teach the subject. By looking beyond the strictures of the curriculum, it will enable the teacher to plan an exciting and stimulating journey for learners of all ages. (HW)

CLEVERLANDS

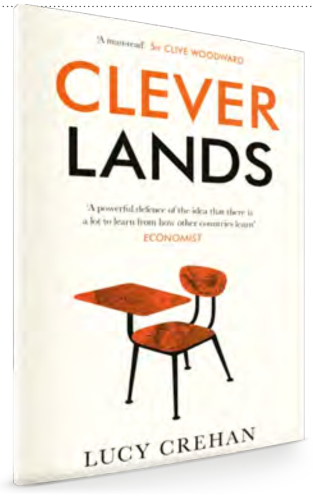
A tour through the world's top-performing education systems

INFO **BY** Lucy Crehan | **PRICE** $12.43 | **ISBN** 9781783524914 | **URL** **helloworld.cc/cleverlands**

Gemma Coleman

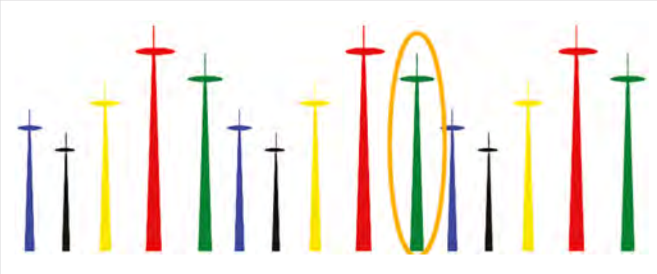
In *Cleverlands*, Lucy Crehan takes readers through her "geeky gap year", visiting five education systems around the world that are considered top-performing: Finland, Japan, Shanghai, Singapore, and Canada. The author, now an educational consultant and self-declared education explorer, was inspired to take the trip after teaching in a secondary school in a deprived area of the UK. She felt that no matter how hard she worked as a teacher, it seemed to have little effect on the outcomes for her students, who were bound up in an education system that wasn't working for them. The aim of her trip was to understand how different education systems work and what we can learn from them.

The book is a joyful blend of policy research, historical deep dives, and meanderings through the psychology and culture of each nation. With each education system divided into a number of manageable chapters, it makes a good read to dip in and out of. I felt rather mournful after her visits to some schools (why couldn't I have started school at seven, like in Finland, and have grown up in a system that embraces play?), and enjoyed her concluding round-up of the principles that make a high-performing education system. This is a must-read for anyone interested in global education and for those dreaming of a different approach! (HW)


'A must-read' Sir CLIVE WOODWARD
CLEVER LANDS
'A powerful defence of the idea that there is a lot to learn from how other countries learn'
ECONOMIST
LUCY CREHAN

BEBRAS PUZZLE SOLUTION (PAGE 94)

The missing picture is added below:



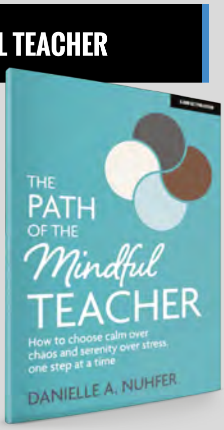
As we are told there is a repeating pattern, we can notice the blue, black, yellow, and red towers appear three times. Adding a third green tower completes the pattern.

ESSENTIAL READING

Three books that focus on well-being

THE PATH OF THE MINDFUL TEACHER

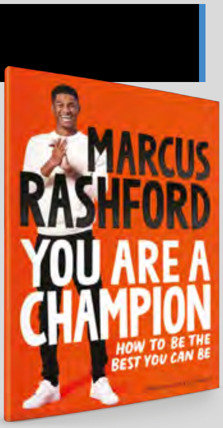
BY Danielle A. Nuhfer
PUBLISHER John Catt Educational
PRICE $19.95
ISBN 9781913622619
URL **helloworld.cc/mindfulteacher**


THE PATH OF THE Mindful TEACHER
How to choose calm over chaos and serenity over stress one step at a time
DANIELLE A. NUHFER

The author — a teacher, mindfulness practitioner, and teacher wellness coach — introduces educators to techniques for finding work–life balance, reducing symptoms of burnout, managing stress, and increasing classroom job satisfaction.

YOU ARE A CHAMPION

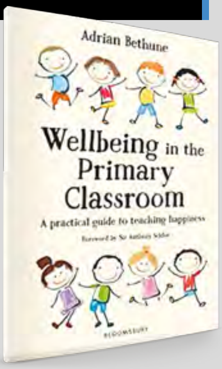
BY Marcus Rashford and Carl Anka
PUBLISHER Macmillan Children's Books
PRICE $18.69
ISBN 9781529068177
URL **helloworld.cc/champion**



In this book for children aged 9 to 12, footballer Marcus Rashford encourages young readers to be comfortable in their own skin, get out of their comfort zone, use their voice to stand up for others, and never stop learning.

WELLBEING IN THE PRIMARY CLASSROOM

BY Adrian Bethune
PUBLISHER Bloomsbury Publishing
PRICE $26.95
ISBN 9781472951557
URL **helloworld.cc/primarywell**


Adrian Bethune
Wellbeing in the Primary Classroom
A practical guide to teaching happiness
BLOOMSBURY

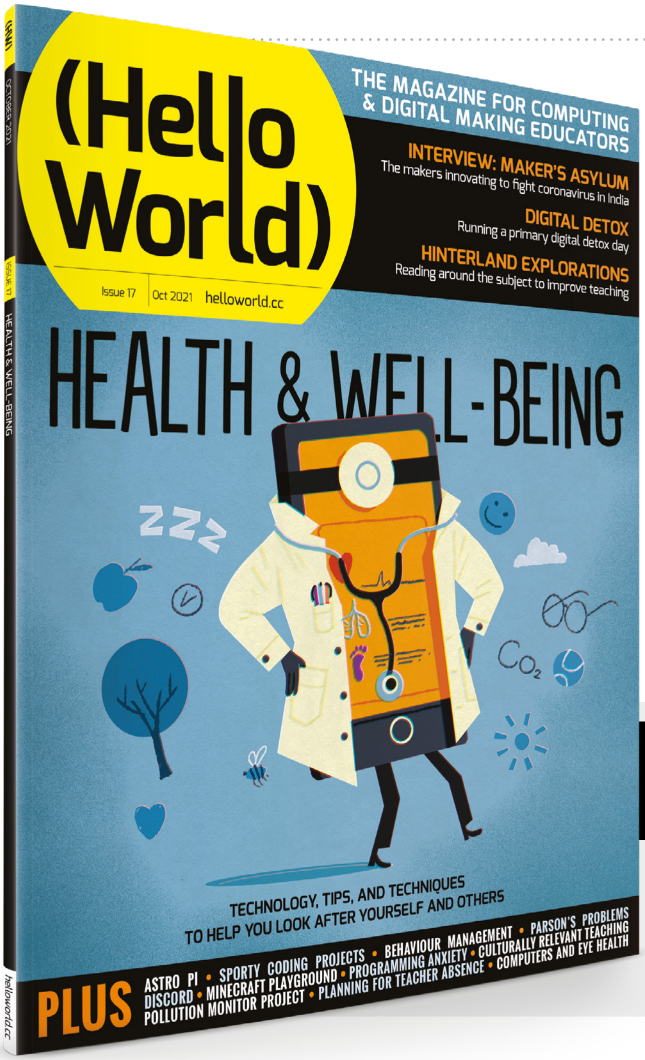
A guide for supporting mental and emotional well-being in the primary classroom, this book is full of research-based activities and techniques that will help teachers introduce ideas of positive mental health to their youngest learners.

"HELLO, WORLD!"

Everything you need to know about our computing and digital making magazine for educators

Q WHAT IS HELLO WORLD?

A Hello World is a magazine for computing and digital making educators. Written by educators, for educators, the magazine is designed as a platform to help you find inspiration, share experiences, and learn from each other.


(Hello World)
THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS
INTERVIEW: MAKER'S ASYLUM
The makers innovating to fight coronavirus in India
DIGITAL DETOX
Running a primary digital detox day
HINTERLAND EXPLORATIONS
Reading around the subject to improve teaching
Issue 17
Oct 2021
helloworld.cc
HEALTH & WELL-BEING
TECHNOLOGY, TIPS, AND TECHNIQUES TO HELP YOU LOOK AFTER YOURSELF AND OTHERS
PLUS
ASTRO PI • SPORTY CODING PROJECTS • BEHAVIOUR MANAGEMENT • PARSON'S PROBLEMS
DISCORD • MINECRAFT PLAYGROUND • PROGRAMMING ANXIETY • CULTURALLY RELEVANT TEACHING
POLLUTION MONITOR PROJECT • PLANNING FOR TEACHER ABSENCE • COMPUTERS AND EYE HEALTH
OCTOBER 2021
ISSUE 17
HEALTH & WELL-BEING
helloworld.cc

Q WHO MAKES HELLO WORLD?

A The magazine is a joint collaboration between its publisher, Raspberry Pi, and Computing at School (part of BCS, the Chartered Institute for IT).

Q WHY DID WE MAKE IT?

A There's growing momentum behind the idea of putting computing and digital making at the heart of modern education, and we feel there's a need to do more to connect with and support educators, both inside and outside the classroom.

Q WHEN IS IT AVAILABLE?

A Your 100-page magazine is available three times per year. Check out our new podcast too, to get more great Hello World content between issues.

IT'S FREE!

Hello World is free now and forever as a Creative Commons PDF download. You can download every issue from **helloworld.cc.** Visit the site to see if you're entitled to a free print edition, too.

WANT TO GET INVOLVED?

There are numerous ways for you to get involved with the magazine.
Here are just a handful of ideas to get you started.

- **Give us feedback**
  Help us make your magazine better – your feedback is greatly appreciated.
- **Ask us a question**
  Do you have a question you'd like to share? We'll feature your thoughts and ideas.
- **Tell us your story**
  Have you had a success (or failure) you think the community would benefit from hearing about?
- **Write for the magazine**
  Do you have an interesting article idea? Visit **helloworld.cc/writeforus** to submit your idea.

GET IN TOUCH

Want to talk? You can reach us at:
contact@helloworld.cc

FIND US ONLINE

www.helloworld.cc

@HelloWorld_Edu

fb.com/HelloWorldEduMag

NEXT ISSUE OUT
FEBRUARY 2022

(Hello World)

helloworld.cc