

(Hello World)

THE MAGAZINE FOR COMPUTING
& DIGITAL MAKING EDUCATORS

PEER MENTORING

Engaging nursery children with computing

INTERVIEW: THE INTERNET SOCIETY

Growing a stronger internet

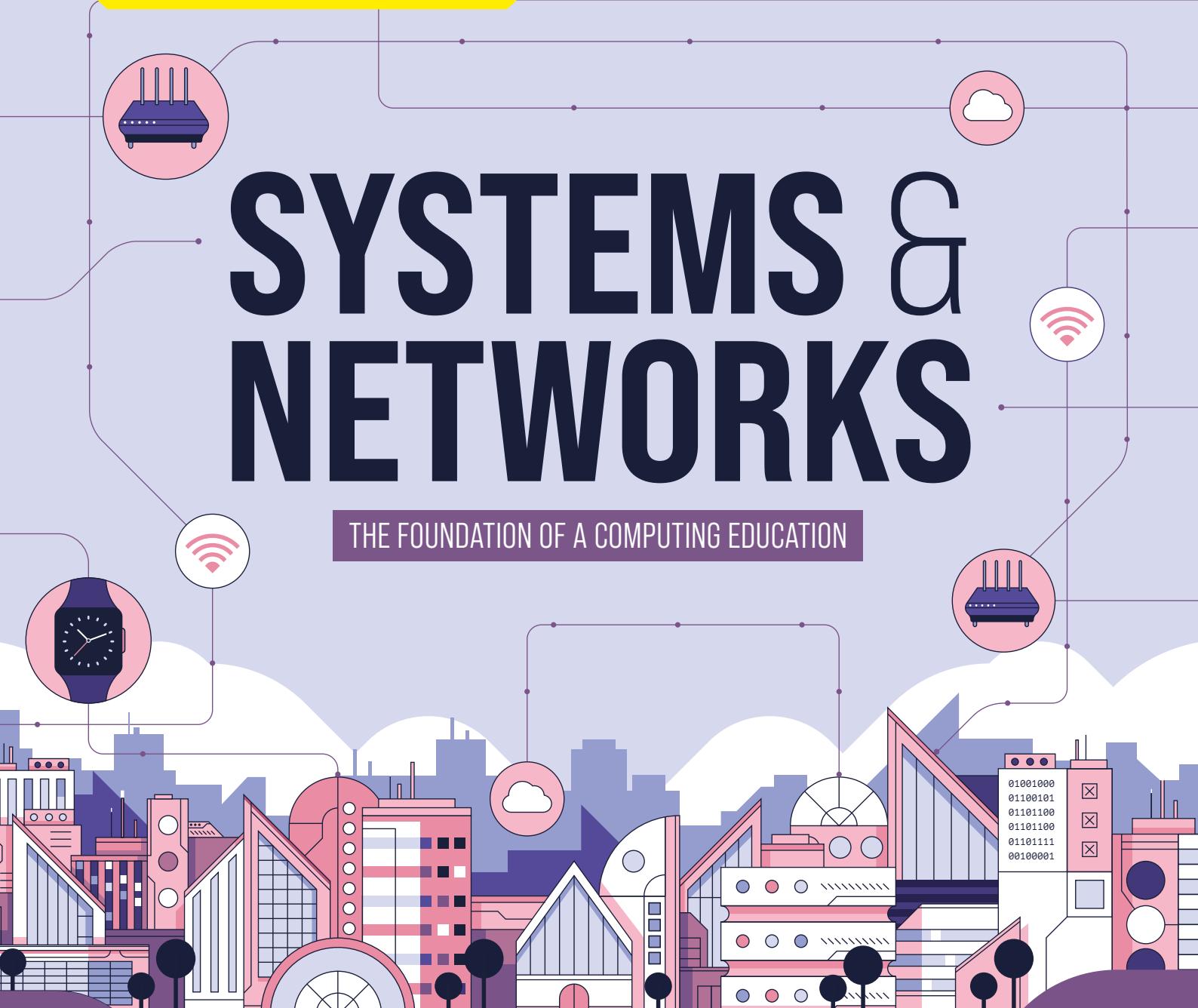
SCAFFOLDED READING

Teaching literacy in computing lessons

Issue 20 | Jan 2023 | helloworld.cc

SYSTEMS & NETWORKS

THE FOUNDATION OF A COMPUTING EDUCATION



PLUS

COOLEST PROJECTS • TECHNOLOGY PRESERVATION • COPILOT • HINGE QUESTIONS
NETWORK SIMULATOR TOOLS • MINECRAFT ESPORTS • ASTRO PI • DIGITAL SCHOOLHOUSE
PREVENTING BURNOUT • GOOD PROGRAMMING REQUIREMENTS • PICOZERO LIBRARY



Raspberry Pi
Foundation

Teach and learn with the Raspberry Pi Foundation

Free for everyone anywhere in the world



Teachers' resources

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

raspberrypi.org/teach



Project library

Browse 200+ online project guides that help young people create with digital technologies.

projects.raspberrypi.org



Research to practice

Explore our research into what works best in teaching school-age young people about computing.

raspberrypi.org/research



Coding at home

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

raspberrypi.org/learn

(HW)

HELLO, WORLD!

Have you tried switching it off and on again?", "The Wi-Fi must be down!", "How come I can't see the cloud up in the sky?" Computing systems and networks is one of those computer science topics in which misconceptions abound, and learners can struggle to grasp its abstract concepts. However, as Sway Grantham argues on page 26, these are the concepts that should be at the very heart of computing curricula, and without them, it can be hard to build knowledge in other key areas of the subject.

This issue offers examples and inspiration for unpacking these tricky ideas with your learners. We explore the research into approaches to teaching networking (on page 12) before seeing some of those methods in action. Ben Garside discusses the merits of inviting a network manager into your classroom (on page 21); we profile a number of different network



simulator tools (on page 22); and we delve into ways to make networks concrete for primary learners (on pages 20 and 32).

For even more great ideas about teaching computing systems and networks, check out *The Big Book of Computing Content* for a collection of our favourite articles on these topics from previous Hello World issues (helloworld.cc/bigbook2). As always, please get in touch with your thoughts and ideas about Hello World at contact@helloworld.cc, and if one of your New Year's resolutions for 2023 was to give writing an article a go, visit [helloworld.cc/writeforus!](http://helloworld.cc/writeforus)

Gemma Coleman
Editor



FEATURED THIS ISSUE



MATT HOLT

Matt is a primary teacher in West Yorkshire, UK. On page 42, he outlines his approach to cross-age peer mentoring with nursery children.



KURI DIFEDE

Kuri is a computer science teacher on Long Island, USA. She shares how she implemented asynchronous learning with her students on page 56.



PETER RUTHERFORD

On page 48, Peter, a primary teacher in London, UK, describes an end-of-year project creating arcade machines and retro-style games.

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

Olivia Mitchell, Ty Logan, Sam Ribbits

Photography

The Raspberry Pi Foundation,
Adobe Stock

Graphics

Rob Jervis

Cover

© Nick Slater, Folio Art

CONTRIBUTORS

Rachel Arthur, Iona Bepey, Victoria Berkowitz, CAS Assessment Working Group, Katherine Childs, Gemma Coleman, Michael Conterio, Valerie Critten, Josh Crossman, Anjali Das, Kuri DiFede, Catherine Elliott, Jill Felbeck-Jones, Tina Fountain, Helen Gardner, Ben Garside, Sway Grantham, Alex Hadwen-Bennett, Hannah Hagon, Ben Hall, Jessica Hamer, Laura Holborow, Matt Holt, Sophie Hudson, Emily Jones, Mirnali Koneswaran, Shashi Krishna, Chris Lovell, Duncan Maidens, Jen McCulloch, David Messer, Siobhán Morgan, Alan O'Donohoe, Martin O'Hanlon, Jared O'Leary, John Parkin, Andrew Rice, Neil Rickus, Trisha Rhodes, Chris Roffey, Peter Rutherford, Laura Sach, SCARI Computing Project, Bonnie Sheppard, Mike Smith, Andy Storey, Patricia Tanner, University of Cambridge: Faculty of Education, Jane Waite, Chris Walker



Raspberry Pi
Foundation

Hello World is
published by the
Raspberry Pi
Foundation:

Contributing Partner

bcs Academy
of Computing



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. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).

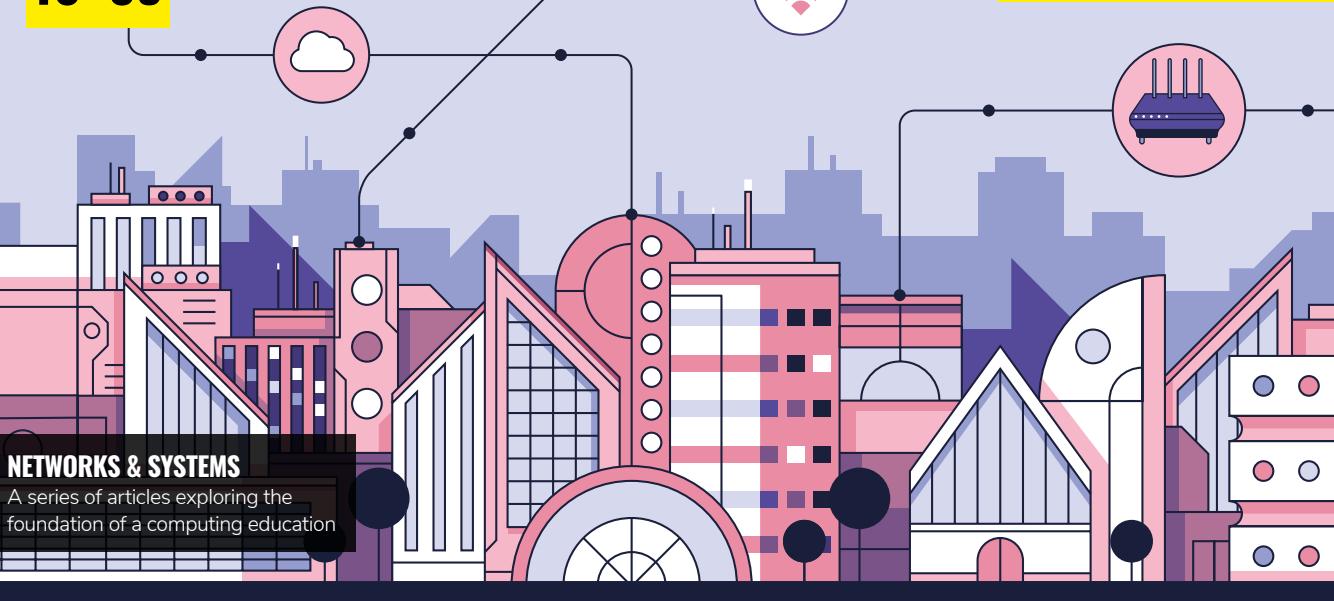


(HW)

CONTENTS

10–39

COVER FEATURES



NETWORKS & SYSTEMS

A series of articles exploring the foundation of a computing education

NEWS, FEATURES, AND OPINION

6 NEWS

Coolest Projects launch; new research seminar series; CAS course; computational thinking research

10 INCLUSION COLUMN

Engaging all learners with computing systems and networks

12 INSIGHTS — NETWORKS

Teaching approaches to networks

14 INSIGHTS — NETWORKS

An internet policymaking framework

15 INSIGHTS — NETWORKS

Unplugged approaches to networks

16 RESEARCH — CAS

Research into hinge questions

18 INTERVIEW

The Internet Society discusses its philosophy of an open and accessible internet

19 THE INTERNET REVOLUTION

Exploring the growth of the internet

20 NETWORKING ROLE PLAY

How to make networks concrete at primary level

21 MAKING CONNECTIONS

Organising a network manager visit

22 SIMULATOR TOOLS

A rundown of popular network simulator tools

24 SWITCHING IT OFF AND ON AGAIN

Exploring troubleshooting advice

26 FOUNDATIONAL KNOWLEDGE

Why computing systems should be at the heart of computing curricula

28 WEAVING THE REAL-LIFE INTERNET

Activities that show how the internet works

29 GETTING HANDS-ON

Using Raspberry Pi Pico to learn about computing systems

30 THE IPO MODEL

The importance of the IPO model to all aspects of computing

32 ONLINE SAFETY

Understanding networks to ground e-safety in reality

34 NETWORKING JOBS

Profiles of networking careers to inspire students

36 THE CLOUD

How the cloud has changed approaches to teaching networking

38 SAFE NETWORKING

Exploring IP addresses

40 SUBSCRIBE TODAY!



DIGITAL SCHOOLHOUSE

A community outreach programme

46



60

GIRLS AND COMPUTING

A study into what works for girls in computing



70

SCAFFOLDED READING

Teaching literacy in computing lessons

LEARNING

RESOURCES & LESSON PLANS

78 NETWORKS AND PROTOCOLS

Introducing learners to the concept of a protocol

82 EXPERIMENTAL LAN

Building an experimental LAN for students to play with and explore

CONVERSATION

84 INSIDER'S GUIDE

Lesson backup ideas

88 PICOZERO LIBRARY

Discussing accessible software on Raspberry Pi Pico

90 A COMPUTING JOURNEY

Moving from discrete computing to tech integration

92 ASTRO PI

One teacher's experience of the Astro Pi Challenge

94 KNOCK KNOCK

Programming jokes

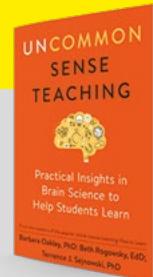
95 BEBRAS

A fun challenge in computational thinking

REVIEWS

96 BOOK REVIEWS

Uncommon Sense Teaching and Hello Ruby: Expedition to the Internet



- 41 A BALANCING ACT**
Supporting students with equations
- 42 PEER MENTORING**
Engaging nursery children with computing
- 44 MINECRAFT ESPORTS**
Setting up a league in primary school
- 46 DIGITAL SCHOOLHOUSE**
Community outreach with the Digital Schoolhouse programme
- 48 ARCADE MACHINE PROJECT**
A cross-curricular project to build excitement
- 52 PREVENTING BURNOUT**
Educators' top tips for avoiding burnout
- 54 CLOZE QUESTIONS**
The power of images in cloze questions
- 56 ASYNCHRONOUS CLASSES**
Introducing asynchronous computing classes
- 58 SELF-REGULATION**
Building student agency

- 60 GIRLS AND COMPUTING**
A study looking at what works well
- 62 DEFINING REQUIREMENTS**
What do good requirements look like in programming?
- 64 CROSS-CURRICULAR CODING**
Integrating coding into different subjects
- 66 CT FOR PRESCHOOLERS**
Assessing computational thinking in our youngest learners
- 69 COMPUTING AND ENGLISH**
Integrating computer science into the English classroom
- 70 SCAFFOLDED READING**
Teaching literacy in computing lessons
- 72 COPILOT**
Using an AI pair programmer tool
- 74 TECHNOLOGY PRESERVATION**
Exploring older technology
- 76 A SENSE OF PROGRAMMING**
Exploring how visually impaired students understand programming concepts

COOLEST PROJECTS IS BACK FOR 2023!

Young digital makers can show the world their creations in the Coolest Projects showcase

Helen Gardner

Coolest Projects (coolestprojects.org) is the world's leading technology showcase for young people across the globe. In 2023, it will return for an online event on 6 June 2023, open to young creators up to age 18 from anywhere in the world.

Get inspired

In 2022, over 2,500 young people from 46 countries shared their creations in the Coolest Projects online showcase (helloworld.cc/coolest2022). Makers could choose from a variety of categories, such as the environment, health, and education, and their projects included everything from foam-flinging toys to robotic hands and super-fun Scratch games.

Over 300 projects used technology to solve problems in the creators' communities. Freddie from Canada created a bicycle safety device with a proximity sensor, clock, fall detector, and radio assistance. He wanted to reduce accidents and encourage more people to access the benefits of cycling (helloworld.cc/coolestfreddie). Hattan and Dan from Sudan created an air-quality monitoring robot that measures humidity, temperature, and nitrogen and methanol in the air. It then sends measurements via Bluetooth to help factories monitor emissions (helloworld.cc/coolestdan).

Some projects were all about having fun. Harshit from an Irish CoderDojo created a game called Runaway Nose, using a webcam and facial recognition, which gets the user moving around to catch on-screen circles on their nose! Harshit wanted to

create a fun game to get people active, even on rainy days (helloworld.cc/coolestharsht). The project was one of judge Richa Shrivastava's (co-founder of Maker's Asylum) favourites. "I thought it was amazing that Harshit could mix facial recognition and his need to move into a game and make it so interactive and fun!"

Young creators like Harshit told us that Coolest Projects made them feel more confident in their computing skills, and that it had made them more interested in programming and computers. As with Freddie, Dan, and Hattan, Coolest Projects gives makers an opportunity to solve problems that matter to them, have their achievements celebrated, and develop skills and confidence in idea generation, coding, and user testing.

Get involved

If you know any young digital makers, encourage them to submit what they've made to Coolest Projects. Projects at all

skill levels are welcome, and there are categories for hardware, mobile apps, web, Scratch, and advanced programming. We love seeing great ideas and works in progress, so young people can also submit unfinished projects and prototypes.

If you want to help your young creators make something new for Coolest Projects, take a look at the Raspberry Pi Foundation's project paths at projects.raspberrypi.org. Each path helps young people to learn coding and design skills to make things that matter to them, and the final 'Invent' stage of each path is a unique challenge that young people could use as their entry into Coolest Projects.

The event is completely free and open to all. Registration is open from **6 February to 26 April 2023**, and young people who are 18 years old or younger can submit their work at coolestprojects.org. Keep an eye out for the announcement of this year's VIP judges and the cool swag for all participants coming soon too! **[HW]**



Coolest Projects gives makers an opportunity to solve problems that matter to them

NEW PRIMARY COMPUTING SEMINARS

The Raspberry Pi Foundation is hosting free online seminars about primary computing teaching and learning

Jane Waite and Bonnie Sheppard

Computer science has been taught in universities for many years, but has only more recently been introduced in schools. That means there isn't a lot of research about computing education for school-aged learners yet, and even less research about how young children in primary schools learn about computing.

Over the last three years, the Raspberry Pi Foundation has offered educators a close look at current research in computing education (helloworld.cc/rpfseminars). Our newest seminar series, which started earlier this month, looks at primary computing education and aims to connect research and teaching practice, and to further primary computing education across the globe.

Each seminar is an opportunity for educators to hear from specialists discussing their research in straightforward language and to pick up new ideas for activities and approaches to try out with learners. In this series, guest speakers will present their research on topics that will help educators to:

- Support young people to learn to program
- Explore opportunities for the assessment of programming
- Integrate computing with other competency areas, such as literacy

What you can expect

Each seminar starts with a presenter explaining, in easy-to-understand

terms, some of their recent research. This presentation is followed by a discussion in smaller groups. We then regroup for a Q&A session with the presenter. The seminars are a chance to reflect on your own experiences and engage in discussions with other members of the computing education community. One seminar attendee recently said, "I love this initiative, your choice of speakers has been fantastic. You are creating a very valuable CPD resource for computer science teachers and educators all over the world. Thank you."

These upcoming seminars are not just for primary teachers — we invite everyone who is interested in computing education to join us. The seminars are attended by a valuable

mix of teachers, volunteers, tech industry professionals, and researchers, all interested in exploring how computing education research can be put into practice.

Whether you teach in a classroom or support learners in a coding club, you will find out how our youngest learners develop their computing knowledge. You'll also explore what this means for your learning context in practical terms. Like all of our research seminars, these sessions are free and open to everyone. We look forward to seeing you soon and discussing how we can apply research results to better support all our learners.

To find out more, sign up to our mailing list and check out our seminar schedule at helloworld.cc/rpfseminars. (HW)

UPCOMING SEMINARS

7 February: Dr Jean Salac from the University of Washington will speak about a new learning strategy that has been found to improve students' understanding of computing concepts and to increase equal access to computing.

7 March: Dr Bobby Whyte from the Raspberry Pi Foundation will share practical examples of how primary computing can be integrated into literacy education. He will specifically look at storytelling elements within computing education and discuss the benefits of combining competency areas.

6 June: Aim Unahalekhaka from Tufts University in Massachusetts will present her research into how children learn coding through ScratchJr. Participants are encouraged to bring a device with ScratchJr so that they can look at practical project evaluations and teaching strategies that can help young learners to create purposefully.

YOUR VOICE IS POWER

Make beats, learn code, and promote equity with this new competition

Iona Bepey

How can coding through musical beats help to promote equity and social awareness in schools? Computing at School (CAS) is delighted to be helping students aged 11–18 to find out, in partnership with Amazon, Pharrell Williams' education nonprofit YELLOW, and Georgia Tech. Launched to great success in the USA in 2021, the Your Voice is Power (YVIP) competition is now debuting in UK schools (yourvoiceispower.co.uk).

Exploring music and social change

Your Voice is Power is a remix competition that challenges students to code an equity-promoting musical remix of Pharrell Williams' song Entrepreneur. Teachers can bring the competition to life through the accompanying project-based curriculum, which introduces the fundamentals of computer science and coding (JavaScript and Python) to students through exploring music and social change. Across five one-hour learning modules, students engage in fun activities and thought-provoking

discussions to learn how these topics intersect. For example, in one module, students hear stories celebrating amazing entrepreneurs who 'set their own tempo', inspiring learners with music clips to compose and code a unique 16-bar song.

Beyond the opportunity for students to develop and use skills in Python and JavaScript, YVIP focuses on promoting social justice and equity for all. The YVIP curriculum touches upon focal points within the UK's understanding of a range of key issues, from the Black Lives Matter movement to the sociopolitical origins of rap and hip-hop. "YVIP is a wonderful cross-curricular and cross-phase opportunity for our pre-service primary and secondary teachers," said one training attendee from the University of Reading. "They've collaborated and discussed links between music and computing, but also how it [the competition] promotes discussion around cultural capital and equity."

Support for teachers

Support is available to help teachers make the most of the competition and the modules. Educators can download a one-hour recorded session for the classroom, taking students through how to use Georgia Tech's EarSketch platform, which they will use to code their music. More in-depth pre-recorded training sessions are also available to teachers, at which specialists from BCS, The Chartered Institute for IT, will take participants through the modules, look at how to get the most out of EarSketch using Python, and share all the information they



■ Your Voice is Power challenges students to code an equity-promoting musical remix of a Pharrell Williams song

need to enrich their curriculum and inspire their students.

A teacher who attended one of the first training sessions said: "It's really helped me get a better idea of what is involved, and I'm sure we would have students who would love to take part. It sounds like a great way to get some cross-curricular links going with other departments, and particularly to encourage greater participation from girls and some of the students who might not always feel like computing is relevant to their aspirations. The resources and EarSketch look really great; it's so useful that they're available online and don't need anything installing."

Teachers, parents, or caregivers can sign up today at helloworld.cc/YVIP, read more about the terms and conditions and prizes, and find the resources mentioned throughout the article. The competition closes on 30 March 2023. Make beats, learn code, and promote equity! (HW)

PRIZES

- **Grand Prize-winning student:** £500 Amazon gift card
- **Grand Prize-winning school:** £2000 Amazon gift card
- **Four runners-up:** £250 Amazon gift card
- **Forty finalists:** £50 Amazon gift card

COMPUTATIONAL THINKING COMPETENCY

Researchers have trialled a new approach to testing students' computational thinking in terms that go far beyond the programming skills learnt in computing lessons

University of Cambridge: Faculty of Education

Computational thinking is a relatively new goal of many education systems, and features particularly prominently in computing education. Outside the school system, however, computational thinking tends to refer to a richer set of cognitive skills linked to problem-solving, which may or may not involve computer programming. A new University of Cambridge study tested an assessment called the Computational Thinking Challenge (CTC) which attempts to measure computational thinking as a general capability (helloworld.cc/lai2022). The analysis showed that it is possible to assess students' computational thinking as a broader competency, suggesting it can be cultivated across the curriculum, rather than in computing lessons alone.

A broader competency

Beyond education, computational thinking is often understood in broader terms. In fields such as computational biology and linguistics, it is seen as a set of cognitive skills and abilities which, for example, include breaking a problem down into its constituent parts, identifying similar challenges, and sorting through large quantities of information. "This broader conceptualisation is often neglected in formal learning because it's difficult to understand and measure," said Dr Rina Lai, who led the research and development of the CTC. "The upshot is that there's a disjunction between what we want at a policy level, or in workplaces, and what's happening in schools." The CTC aims to address this by providing teachers with a tool for measuring students' overall

computational thinking competency. Lai hopes to continue its development and make it freely available online for teachers.

The CTC is a computerised assessment consisting of 14 questions covering problem-solving and programming elements. Crucially, these are presented in everyday language that non-coders can understand. Upon completion, students receive personalised feedback, encouraging them to apply computational thinking in other contexts.

The study

In their analysis, Lai and co-author Michelle Ellefson, Professor of Cognitive Science, examined the extent to which this test measures computational thinking as a multidimensional construct that goes beyond students' programming skills.

of computational thinking as an overall competency. Lai said, "We needed to know this because, while you can design a test to be multidimensional, having the evidence it works at this level is another matter."

The researchers then investigated how the test was working in psychometric terms, and whether, as well as providing an aggregate score for students' computational thinking competency, it also offered a more detailed picture of their programming and non-programming problem-solving skills.

This has implications for how computational thinking assessments are designed. Traditionally, these tests tend to use the same type of question, whereas the CTC combines different question formats and types to tap into both programming and non-programming skills. The analysis showed that the CTC

“ COMPUTATIONAL THINKING CAN BE CULTIVATED RIGHT ACROSS THE CURRICULUM ”

A sample group of 1130 secondary-school pupils in the UK took the test, and the researchers conducted several analyses of the resulting data. First, they compared how well the data fitted with two different models. One represented computational thinking as a unidimensional construct, and the other represented it as a multidimensional construct consisting of both programming and non-programming skills. The data clearly fitted the multidimensional model more closely, and appeared to provide a picture

score reliably reflects the complexity of computational thinking, incorporating measures of these different elements.

Although the CTC is still in development, Lai concludes, "It tells us something more about students' computational thinking skills than a curriculum-aligned assessment could. Because it doesn't tap into what students are learning in computing education alone, it opens the door to assessing and growing this crucial competency in other contexts and subject areas as well." (HW)

ENGAGING ALL LEARNERS WITH COMPUTING SYSTEMS AND NETWORKS

Catherine Elliott outlines how you can use the Universal Design for Learning framework to make computer systems and networks more engaging and inclusive for learners

The Universal Design for Learning (UDL) framework aims “to improve and optimize teaching and learning for all people based on scientific insights into how humans learn” (helloworld.cc/udlcast). It is not subject-specific, and you can use it to ensure that your lessons are as inclusive as possible by providing multiple means of engagement, representation, and action and expression (see my column in issue 18 for an introduction to UDL: helloworld.cc/18). In this article, I will provide a number of examples of how to teach computer systems and networks through the lens of UDL.

Engagement

The engagement aspect of the UDL engages learners by making content relevant, providing choice, and minimising threats. To approach computing systems and networks in this way, you can:

- Use authentic and meaningful contexts to make content about networks and the internet relevant. What do your students use the internet for? What networks are they part of? For example, the Teach Computing Curriculum’s Year 3 ‘Systems and networks’ lessons ask pupils aged seven to eight to consider their personal networks.
- Foster collaboration and community by modelling how we can use the internet and the World Wide Web to work with

others on a joint project, for example by using a Google or Microsoft 365 document, or a class blog about a topic. Working together virtually may suit certain students better than face-to-face collaboration, as they can concentrate on their own specific parts of the task.

- Harness students’ personal interests to increase engagement with the subject, for example by using Minecraft to explore logic gates (see the article on page 17 in *The Big Book of Computing Content*: helloworld.cc/bigbook2).

Representation

There are three elements to the representation aspect of the UDL framework: presenting information in a range of ways and through a range of media; supporting the language required to understand the content; and supporting comprehension by building connections, highlighting key information, and modelling. So, how can you incorporate this approach for the topic of computing systems and networks?



Catherine Elliott is the SEND lead for the Sheffield eLearning Service (sheffieldclc.net), and since the announcement of the computing curriculum in 2013 she has been working on ways to make the subject accessible for 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).

- Provide key information in a range of different formats. Complex systems, such as the internet, can be represented by images, animations, audio descriptions, and videos (for example, helloworld.cc/internetvideo). Give your students opportunities to explore content in a way that makes sense to them.



Students could act out the parts of a computer network

© Hello Ruby

- Use concept maps to illustrate the connections between the different parts of a computer system, such as input and output devices. This can help students comprehend the big picture and the relationships between elements, which autistic learners can particularly struggle with.
- Pre-teach the key vocabulary for a topic, either by providing word lists and glossaries to students, or by presenting an overview of the terms that you will use. As well as supporting learners with remembering the key terms, this will reduce the amount of brand-new content they need to learn in lessons. See the CAS Include website for a number of image-supported glossaries for each area of the curriculum in England (helloworld.cc/CASincluderelources) — but note that the computer systems and networks terms are currently in development!

Action and expression

Learners navigate learning environments and express what they know in different ways. As teachers, we can allow for a number of different outcomes for learning using technology:

- When teaching about a specific computer system, for example a network in a school, students can represent what they know by drawing a diagram, creating an animation, making a video, or completing a presentation. In this way, students who struggle with written work can still communicate what they know.
- Consider the use of assistive technologies. This can be a teaching point for all learners to find out about how operating systems, hardware, and software are developed to support people with disabilities. What tools are available to support people with visual



LET STUDENTS EXPLORE CONTENT IN A WAY THAT MAKES SENSE TO THEM

impairments, for example? Students could research these and present their findings to the class.

- Unplugged activities are useful for teaching abstract concepts, and can harness a range of activities around physical movement, manipulation of objects, and the use of sounds. For example, you could model the flow of information around a network by giving students different roles (email message, switch, router, and so on); or students learning about physical systems and sensors could act out the parts of an automatic door or a security light using props.
- *Hello Ruby* (helloworld.cc/play) has a large number of lovely activities for younger pupils that will help them to learn about computers by using different sensory activities, such as drawing the internet, learning about packet switching using vegetables, and building a computer out of paper.

The great thing about the Universal Design for Learning framework is that it is, well, universal! By considering a greater range of ways of presenting information, and a wider range of means by which learners can express themselves, you will engage a greater number of young people with computing. This will not only benefit students with special educational needs and disabilities; you may also find it helps to engage a more diverse cohort of learners, from different backgrounds, genders, and ethnicities. [\(HWI\)](#)

#INSIGHTS

METHODS FOR TEACHING NETWORKING

STORY BY Katharine Childs

Teaching about computer networks involves balancing the technical, abstract concepts of network design with practical activities and applications grounded in real-life scenarios. A recent literature review summarised over 190 different research papers covering the range of teaching approaches used in computer networks courses (helloworld.cc/prvan2020). The researchers used the findings to compare the benefits and challenges of each approach. Although the research review focuses on undergraduate courses that often take place in computer labs, many of the explanations and findings can also apply to K-12 (ages 5–18) education.

Teaching methods

The teaching methods reviewed are all alternatives to the traditional direct-instruction approach in which a teacher presents information to a class via a lecture. The researchers created a categorisation system for the various teaching methods used in the research papers: visualisation methods, virtualisation methods, active-learning methods, and practical hands-on lab exercises.

Packets of data move around a computer network invisibly, and so it can be difficult for learners to understand the various protocols and stages involved in network data transmission. Visualisation techniques allow learners to 'see' the journey that data

takes across a network. Some examples of teaching activities using these techniques include using animation or a multimedia presentation to show learners what happens as data moves from one point in a network to another. Packet tracing can also show the 'hops' that data takes in a more concrete way. There are also network simulator applications that can simulate the layered network model and the details of network protocols.

Virtualisation methods use a type of learning environment that combines real network devices with simulation-based design. You can use tools to help create virtual labs, such as virtual machines, which allow students to experiment with real network devices via remote, internet-based connections. These techniques allow students to learn from any place and without time limitations. This approach has been shown to be beneficial for students because it enables them to experiment and learn from their mistakes (helloworld.cc/cui2012).

The review of existing research found that many courses use active learning methods to help learners engage with networking topics. An example discussed in this category of teaching approaches is cooperative learning, which involves students working together to achieve a common learning goal. It encourages critical thinking and helps students to develop communication skills. In this type

CLASSIFYING NETWORKING TEACHING

Four overarching methods for teaching about computer networks are presented in the research, as follows:

Category	Examples of activities
Visualisation techniques	Animations, network simulators, multimedia presentations, packet tracing, visual analogies
Virtualisation methods	Setting up a network in a virtual lab that accesses real-life equipment via a configured connection
Active-learning methods	Student collaboration and competition, problem-based learning, playing games, failure and counter-examples
Practical hands-on lab exercises	Setting up a network in a hands-on lab using real-life equipment



Learners take part in hands-on lab activities to learn about computer networks

of learning, the teacher has a supportive role and the students collaborate to share ideas and solve problems. Teachers can combine cooperative learning activities with problem-based learning, in which they design a network with deliberate errors and learners work together to discover the cause of the network failure.

Finally, hands-on lab exercises involve learning about networks by configuring real equipment, although the researchers note that the existing research suggests these types of learning activities depend on first understanding the theory behind network design.

Benefits and challenges

In the reviewed literature, the researchers found a number of advantages of using the various methods. Some of these advantages are practical: for example, it can be cheaper to use virtualisation methods than hands-on lab exercises. Other benefits relate to the motivation and engagement of students, which studies suggested are increased by using the active-learning paradigm. Perhaps most importantly, some benefits relate to the skills students develop and the knowledge they gain. Visualisation methods support students' understanding of some very abstract theoretical topics by tracing packet journeys across a network. Meanwhile, in hands-on lab exercises, scenarios involving fixing faulty networks

were shown to develop students' critical-thinking skills.

Thinking about the challenges of using these methods, it's perhaps unsurprising that educators had to overcome problems with group working and motivation when using active-learning techniques, while in hands-on lab exercises, the difficulties related to troubleshooting the technical equipment. The technical expertise required by both teachers and students to successfully set up and complete learning exercises was also highlighted as a potential challenge in virtualisation activities. However, research into visualisation techniques found that most potential challenges related to student understanding of the underlying theoretical concepts — studies that explored the use of analogies, animations, and multimedia found a risk of superficial learning or student misconceptions.

Many of the research studies included more than one of these teaching methods. Although the researchers did not find any evidence to suggest that there was an ideal combination of methods, the analysis suggests that a mixture of theory and practical learning activities is essential.

Effective learning about computing

The research review found evidence that all of these teaching methods led to greater learning gains than traditional direct-instruction methods. Some of the

FURTHER READING

- ✓ Prvan, M., & Ožegović, J. (2020). Methods in Teaching Computer Networks. *ACM Transactions on Computing Education*, 20(3), 1–35. helloworld.cc/issue20insights1

researchers' conclusions applied across all the teaching methods, as follows:

- **Using step-by-step instructions:** in some learning activities, such as device configuration, it is essential to follow a set of instructions. However, as these limit students' critical thinking about the task they are performing, the researchers suggest that educators avoid these where possible.
- **Virtual versus practical:** studies suggest that students learn more when using virtual labs than when taking part in hands-on activities, because their learning has fewer distractions and is more task-focused.
- **Matching methods to topics:** the team concludes that there is not yet enough evidence in the existing research to decide which method is suitable for each topic in a computer networks curriculum. They suggest that this would be a useful focus for future research.

You can read about some of these methods in action in this issue: check out the article on page 22 for a summary of network simulator tools, or Ben Hall's article on page 20 to learn about an active-learning technique for primary students. [\(HW\)](#)

INTERNET POLICYMAKING IN EDUCATION

STORY BY Mirmali Koneswaran

The digital divide issue — the inequality surrounding access to digital devices and the internet — came to the fore during the pandemic, particularly within education. Pre-pandemic data showed that in the UK, 21 percent of children from lower-earning households had no access to a laptop or desktop, and 6 percent had no access to the internet at home (helloworld.cc/UKdigdivide). Further afield, in 2020, only 42 percent of children in Colombia and 12 percent of children in Nepal had access to the internet (helloworld.cc/digdividelessons).

The Internet Society is a global non-profit organisation that advocates for an open and globally connected internet that is secure and trustworthy (internetsociety.org). This article will introduce the framework proposed by the Internet Society to widen internet access in the educational sector, encompassing five priorities for policymakers and educators to consider:

HOW TO BE PART OF THE CHANGE

- Stay informed about current policies (for example, the OECD, oecd.org)
- Collaborate with your local NREN (geant.org is the collaboration of European NRENs)
- Integrate online safety, digital citizenship, and cybersecurity into your curriculum

infrastructure and access; vision and policy; inclusion; capacity; and content and devices (helloworld.cc/internetframework).

Infrastructure and access

If teachers and students are to make full use of the internet, networks and services must be available, affordable, and regularly maintained. Currently, though, broadband access is not equally distributed. Individuals in developed countries, for example, are four times more likely to have internet access than those living in the least developed countries.

Educators can become effective advocates for secure infrastructures and access to resources by collaborating with their local National Research and Education Network (NREN). NRENs are platforms that provide internet services specifically to support the needs of the research and education communities within a country (helloworld.cc/NREN), and educators can work with them to develop internet infrastructures for their school or college.

Vision and policy, and inclusion

Educators should work together with policymakers to enable technology to benefit both students and national development. Educators are more informed about students' needs and aware of their institutions' specific challenges. If educators have a voice in policymaking, they can influence decisions that will help improve their students' digital literacy; identify and reduce students' (particularly girls') vulnerability to online safety risks; and steer the vision to address critical

FURTHER READING

- Internet Society. (2017). Internet Access and Education: Key considerations for policy makers [white paper]. helloworld.cc/issue20insights2

concerns such as gender inequality and access inequality for those areas without internet access.

Capacity, and content and devices

One significant benefit of the internet is improving the quality of education. Educators play an important role in ensuring that students are equipped with digital and digital literacy skills, and the knowledge and understanding they need to protect themselves online. Teachers should therefore have access to professional development to help them build new digital skills and learn how to use the internet effectively, and should bring that skill set to class.

Access to the internet opens up innovative new ways to teach young people. However, it should be noted that not all educational content can be integrated online, as many countries, particularly in Africa and Asia, rely heavily on mobile devices. Educators and policymakers should consider this issue when making positive change; they should be empowered to represent their schools' unique digital inequality story and collaborate with policymakers to reduce the impact of the digital divide, both globally and locally. (HW)

PULLING THE PLUG ON COMPUTER-CENTRIC TEACHING

STORY BY Bonnie Sheppard

The idea that educators can teach computer systems and networks entirely without computers might initially appear unusual. However, the unplugged approach to teaching computer science is in no way new, with the first studies into its benefits published more than ten years ago. Since then, there has been a steady growth of free resources and pedagogical methods with which educators can easily introduce unplugged activities to their classrooms through games, teamwork, and engaging storylines.

CS Unplugged, one of the first collections of unplugged resources, was created by researchers at the University of Canterbury in the early 2000s (csunplugged.org). The resources aimed to make computing more accessible to students of all backgrounds and improve the gender balance in computing. Initially, they were developed for outreach and extension activities. However, it soon became obvious that this approach to teaching was also highly beneficial in the classroom. In a series of follow-up studies, researchers found that students who were taught with unplugged methods showed an increased interest in computer science, an often advanced understanding of computing concepts, and better abstract thinking.

A powerful approach to networking

Computer networks is an area of study that lends itself well to an unplugged approach. Much of what students learn is quite abstract, and an unplugged activity can help bring the topic to life. For example, in the 'Routing and deadlock' game (helloworld.cc/issue20insights3),

`cc/deadlockgame`), five students sit in a circle, facing each other. The educator gives each student a different coloured piece of clothing to wear. The group is then asked to pass around different fruits, to get each fruit

This U-shaped teaching approach, which moves from abstract to simple and back to abstract, ensures that learners gain a solid understanding of the subject matter and its technical vocabulary.

MUCH OF WHAT STUDENTS LEARN IS QUITE ABSTRACT, AND AN UNPLUGGED ACTIVITY CAN BRING A TOPIC TO LIFE

to the person with the T-shirt of the same colour. Every student receives two fruits, apart from one person, who only receives one. In the course of the game, students will discover that to avoid deadlock situations, they sometimes have to give up their matching fruit. Through this activity, they will become familiar with the idea of buffer size and free servers, which can then be used in a subsequent classroom discussion about computer networks.

It is important to note that unplugged computing can fall short of its potential if students are not supported in their learning process. For this reason, it is recommended to follow a semantic wave when teaching with unplugged methods. The teacher first carefully introduces students to the relevant abstract concept and the correct terminology. Then the meaning of the concept is unpacked in an engaging unplugged activity by using familiar analogies and simpler language. Lastly, the simpler meanings are linked back to the specialised terminology and concepts.

Moving away from the classroom computers every now and then allows students to experience computer science in a different mode. Activities that are based away from a computer can deepen knowledge and raise motivation — for both students and teachers! (HW)

FURTHER READING

- ✓ Bell, T., & Vahrenhold, J. (2018). CS unplugged — how is it used, and does it work? In H. Böckenhauer, D. Komm, & W. Unger (Eds.) *Adventures between lower bounds and higher altitudes* (pp. 497–521). Springer, Cham (helloworld.cc/issue20insights3)

- ✓ The Raspberry Pi Foundation (2021). Semantic Waves. *Hello World, The Big Book of Computing Pedagogy*, pp. 46–48 (helloworld.cc/issue20insights4)

 In each issue, the CAS Assessment Working Group share their current activities, findings, or reports relating to computing assessment.

In this column, they explore the power of hinge questions as an assessment-for-learning technique.

ABOUT THE CAS ASSESSMENT WORKING GROUP



The Computing at School (CAS) Assessment Working Group is a CAS community aiming to provide clear statements of informed opinions relating to current issues of formative and summative assessment in the teaching of computing, reflecting academic and professional understanding. The group presents the teacher's voice but is informed by colleagues in awarding organisations, commerce, and academia (helloworld.cc/casawg).

IT ALL HINGES ON THIS QUESTION

The CAS Assessment Working Group share how hinge questions can be a powerful assessment-for-learning strategy in the computing classroom

WRITTEN BY John Parkin

Formative assessment offers all teachers the power to gauge how well children are learning in lessons.

Awareness of formative assessment has grown since the publication of the article *Inside the Black Box* (helloworld.cc/black2010). Here, researchers Black and Wiliam explained that teachers need to use teaching approaches to understand what is going on inside the brains of each learner in the classroom. This greater understanding led to new classroom strategies, including the introduction of hinge questions (helloworld.cc/wiliam2015). Teachers can use hinge questions across all age ranges and subjects, and they are particularly useful for computing teachers. In this article, I will explain what

hinge questions are, how to use them in the classroom, and how they can work in computing lessons.

What are hinge questions?

The idea behind hinge questions is quite simple. During a lesson, how can a teacher quickly find out whether learners have understood the lesson content? To help answer this, the teacher presents a multiple-choice question about the key concept being taught. As well as the correct answer, the question will contain alternative incorrect answers that include misconceptions students might hold about the topic (these are called 'distractors').

The teacher should get a quick response from each student in the class, so they can gauge

2 Why do we need to think computationally?

- To help us to program
- To help us solve complex problems more easily
- To help us to think like a computer

3 Which of the following is NOT a computational thinking technique?

- Decomposition
- Pattern recognition
- Coding

 Two multiple-choice questions that can be used as hinge questions



What is the purpose of a firewall?

- A To identify if a network has been or is at risk of being hacked.
- B To stop unauthorised access to a network or device from the internet or other network.
- C To make sure all users have access to the data that is relevant for them.
- D To update the anti-virus automatically on all user's computers regularly.

A hinge question about networks for lower-secondary students

whether learners have understood the concept. This does not need to be complicated: learners can use their fingers to indicate if the correct answer is 1, 2, 3, or 4. Other strategies include having an ABCD card, small whiteboards, or electronic voting systems. Using these approaches can help the teacher make a quick decision about next steps. Once the votes are in, it is decision time! Based on the answers, the teacher can decide to carry on with Plan A and continue the lesson, or go to Plan B and return to consolidate understanding about the main concept. It's as simple as that!

To create a good hinge question, it is important to spend time devising it. You need to boil down the main idea you're teaching, and anticipate any misconceptions and errors. By combining the correct answer with three to five distractors considering the misconceptions students could make, you'll ensure you really see who knows the correct answer. Careful planning will pay dividends later on!

In the computing classroom

While hinge questions might immediately lend themselves to questions that are more theoretical in nature, you can also use them with more practical aspects of computing. For example, you could show students screenshots of code created in Scratch and ask them to identify the correct sequence. While it will take a little longer to create hinge questions like this, it will support pupils' learning.

If you want to create your own hinge question for a computing lesson, think about a misconception students frequently have and the kind of incorrect answers they share with you. Based on this, you can then create your question. If you are teaching a new topic, looking at misconceptions that others have identified could be a good start — for example, a blog post on the Computing at School website lists several common misconceptions in a range of topics and across different ages

TO CREATE YOUR OWN HINGE QUESTIONS, IDENTIFY THE COMMON MISCONCEPTIONS AND ERRORS STUDENTS OFTEN MAKE

(helloworld.cc/CASmisconceptions). BBC Bitesize (helloworld.cc/bitesizehinge) and Diagnostic Questions (diagnosticquestions.com) also offer a number of free multiple-choice questions that computing teachers could use as hinge questions.

In summary, hinge questions offer a quick and efficient way of helping computing teachers to identify whether or not students have understood the concepts covered in a lesson. Why not try using them in your classroom today? **(HW)**

THE INTERNET SOCIETY: A FORCE FOR GOOD

“IT’S PEOPLE WHO BUILD THE INTERNET”

Gemma Coleman and Fatema Kothari
 discuss the work of the Internet Society and its mission to develop and promote an open internet to empower people around the world

If you just build the technology, there will be people that come, but ... a lot of people unfortunately won't have the means to access it unless you heavily invest in internet adoption and digital skills." Fatema Kothari sums up the Internet Society's ethos pretty neatly: "Our areas of focus have always been growing the internet, protecting the internet, and strengthening the internet. And that has only become more crucial as more and more people come online."

The Internet Society (ISOC, internetsociety.org) is a global, volunteer-driven nonprofit established in 1992 by early pioneers of the internet. Its founding mission was simple — to promote the open development, evolution, and use of the internet as a force for good that could benefit people across the world — and it has consistently kept that principle at the heart of the organisation. Fatema is one of its many volunteers, the previous president, and now a board member of the San Francisco Bay Area Chapter, advocating for a secure internet for all.

Growing, protecting, and strengthening

There are still 2.9 billion people in the world who are offline, which greatly impacts their quality of life and opportunity. "When you think about the digitisation of essential services like healthcare and education, there is no going back to an offline-only world," Fatema warns, "We want to ensure that all communities and regions are able to connect to the internet and use it in a way that's meaningful to them."

For Fatema's chapter, this has included working with a local organisation, Tribal Digital Village Network, to bring internet access (or better internet access) to just under 8000 Native American tribal members. Through a grant from the Internet Society Foundation, this work will help

children get online for schoolwork and support adults with looking for jobs, thus empowering and enriching lives.

With growth, though, comes the responsibility to ensure everything is done in a secure way: "It's supposed to be a tool for bettering people's lives, not something that becomes weaponised. How do we protect that through regulation and good education, so people learn to advocate for themselves?" Fatema says. The Internet Society takes a bottom-up, grassroots approach to action and education. "We've been very intentional not to go down that route of 'Let me tell you how this will affect your life and make it better.' It's more like, 'Why don't you tell us what you need?'" For Fatema's chapter, this has spanned issues of fair use, reverse engineering in Silicon Valley, and AI, and has culminated in a series of tech policy webinars and encryption training. "We had policy experts and board members demystifying a lot of these topics. It's important that not just people with law degrees or engineering degrees understand these issues!

"As a chapter, what makes us strongest is the participation of volunteers in ongoing activities and telling us what is important to them. Really listening to the community is how we shape most of our work," Fatema explains. "The most critical piece of work of the Internet Society is giving voice to the people and having them know that they can drive policy to help shape the future of the internet. It's not just a subset of people working in a small lab building the internet; it's people who build the internet!" **(HW)**



THE INTERNET REVOLUTION

Duncan Maidens explores the growth of the internet

The internet is fundamentally a set of computing devices interconnected across the Earth and space. And how are they connected, I hear you cry? Well, the answer, of course, is the network. While the power of computers has grown according to Moore's law (helloworld.cc/moore), and the programs that run on them have become ever more sophisticated, the network actually underlies this. It's the phenomenal growth in the network's reach, capacity, and reliability that has really driven the internet revolution.

A growing digital network

The first part of the revolution was bringing the network to end users without re-cabling the world. First, the humble telephone line was requisitioned, and we converted data into those warbling tones in a modem and sent information over the telephone network. Technological advances with digital subscriber loop (DSL) vastly improved the use of this medium. Now, we don't think about the 60-year-old twisted pair cable carrying 30–50Mbps. A few years later, the mobile phone network landed, initially with voice calls; but the digital revolution soon had these channels transmitting data. This created the opportunity for the smart devices

that are now in the hands of over half the world's population.

The infrastructure behind all this was a growing network of wires, fibre-optic cables, and microwave and satellite links. However, the propagation delay to a satellite 22,000 miles above the Earth resulted in a delay that was too long for real-time communications, in spite of the huge cost of these devices. Physically connecting continents with cables had already been done with older technologies, and the new fibre-optic cables meant we could now send data over greater distances before needing to boost it. Specialist ships, robust cable construction,

Rapid developments

With these networks, we routinely talk about megabits and gigabits per second, without giving a thought to the enormous improvements and developments in technology that have made these speeds possible. If the car and the infrastructure it uses had developed at the same pace as the sending of data over a network has developed, we would now be driving at about 126,000 miles per second, down a 25-billion-lane highway taking up the same space as a single-track road, and using one tank of petrol to go 24,000 miles. On top of that, there would be a zetta (10 followed

ITHE DIGITAL NETWORK, WITH ITS UNIVERSAL REACH, IS THE PREFERRED CHOICE FOR ALL SORTS OF CONNECTIVITY

and detailed ocean floor surveys mean that we now have about 850,000 miles of undersea cables. This has brought internet access to remote places, and created high-capacity links between continents.

This growth of the digital network, and its universal reach, mean that it is now the preferred choice for all sorts of connectivity. Whereas in the past, companies used their own bespoke networks for their connectivity needs, it is now much cheaper to use the internet's capacity. Thus, our banking, document transfers, infrastructure control, and e-commerce are all now transferred over the same network that we use to watch Netflix, surf the web, and send emails.

by 20 zeros) number of cars on the roads in a year, and you would usually arrive at your destination with no more than a few seconds delay.

The demand on networks will only continue to grow as developers create, and consumers use, more data-hungry applications. The opportunities in networking are as vast as they are in any field, from the deep science of radio wave propagation, to undersea surveys and cable construction, to developing the hardware that connects the links in a network, to the software that runs on the hardware. What new developments will we see in the coming decades? I, for one, am excited to find out. (HW)



DUNCAN MAIDENS

Duncan is the director of computer science education at the Raspberry Pi Foundation.



UNPACKING NETWORKS

Ben Hall addresses how primary educators can teach abstract networking concepts in concrete ways

Networking can be one of the most difficult concepts to teach in computing. Much of what takes place is hidden or invisible: switches and routers are usually tucked away in a cupboard, wires are routed through ceilings and under floors, and wireless connections cannot be seen. The speed and complexity of millions of bits of data travelling at the speed of light is also very difficult to comprehend, especially for primary-aged children.

To structure learning about networks, it is beneficial for learners to follow a semantic wave (see page 46 of *The Big Book of Computing Pedagogy* for more on semantic waves, helloworld.cc/bigbook). After introducing an abstract networking concept, educators can then make that learning more concrete by unpacking the concept and relating it to the real world.

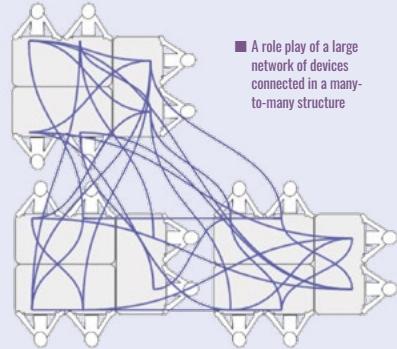
Role play

You can begin by using role play. In its simplest form, a network is two or more devices connected together. One learner can

play the role of computer one, and another the role of computer two. Connect them together by getting learners to hold either end of a piece of string. To simulate data moving, get them to pass an object, such as a keyring, along the piece of string to each other. This is the first part of networking unpacked: things flow between devices that are connected.

You can then create a larger network of devices connected in a many-to-many structure (see image). At this point, you will need to highlight that many-to-many connections are quite inefficient and messy! You can reinforce this by reassigning roles. In groups of about six, make one learner a switch and the remaining learners computers. Each 'computer' will hold a piece of string, and the 'switch' will sit in the middle holding the other ends of all the pieces of string. Now, to pass a message to another computer, a learner will pass it to the switch, who will then 'switch' the message to the right piece of string to get it to its destination.

You can make this model more complex by adding devices such as printers or servers, all linked to the switch. If you want to extend the analogy to the internet, you could join the groups of learners to each other by introducing routers. These will be between networks, joining one network to one or more other networks. You should then be able to get a message from one end of the classroom to another by passing it through all the various connections. This is a very accurate, albeit simplified, representation of what happens on computer networks.



To repack the semantic wave, remember to connect the activity back to a real computer network before showing learners the process happening on a network itself. Before you do this, you will probably have to address an alternative conception that the physical activity may have introduced. In reality, many devices on a network are not physically connected, as they are Wi-Fi-based. You can show learners a wireless access point and explain that the data is sent through the air to their devices, such as mobile phones. These devices are not physically connected to each other, but you can still communicate with other people by using them.

Now that learners understand the fundamental principles of networking, they will be able to apply their knowledge to more complex and larger-scale situations. The string activity is not just a simplified version of local networks, but also of the wider internet. In the case of the internet, though, the string is thousands of miles of fibre-optic cables on the ocean floor — still connections between computers, but on a much larger scale! **(HW)**



BEN HALL

Ben is a learning manager at the Raspberry Pi Foundation. He is a CAS Master Teacher and a Raspberry Pi Certified Educator (@hengehall).



MAKING CONNECTIONS

Ben Garside suggests inviting a network manager into your classroom to bring networking to life

Tradiitionally, networks can be the area of your curriculum that feels a little concept-heavy. Like every other area of computing, however, it's important to bring networking to life through real-world examples and hands-on experiences. One tip I'd like to suggest is to invite a network manager to come along to your lesson to speak with your students.

What can a network manager offer?

At my previous school, we were fortunate enough to have an on-site team that was responsible for the upkeep of the school network. When teaching networks one year, I asked the network manager to come and give a quick talk about what his job entailed. I hadn't expected it to engage the students as much as it did, and what started as a 20-minute careers talk ended up filling most of the lesson with a barrage of questions from learners.

The following year, I thought more deeply about what the network manager could offer to enhance the curriculum. The next time he visited, I asked him to come into a lesson midway through the unit. Having briefed him about the key concepts and terminology students had studied, he came to class and gave an introduction to his role before giving learners a tour of the server room. He showed them the servers and switches,

and explained the purpose of the UPS, why the room was air-conditioned, and how the servers are backed up, where to, and how regularly.

This approach supported the students' mental model of how networks work by seeing the physical hardware that made the network they used each day function. The questions the students asked were more informed than the previous year, and clearly, some misconceptions were being addressed.

Practicalities, preparations, and tips

To ensure you get the most out of a visit, here are my top tips:

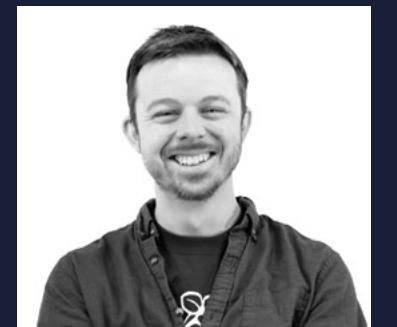
1. Think about where in the sequence of lessons a visit would best fit. What knowledge would it be beneficial for students to have already in order to get the most out of it?
2. The network manager might not be used to being in front of a classroom of young people. Help them prepare by showing them an overview of the unit, what knowledge pupils should have before the visit, and what they will learn about afterwards. It's also helpful to inform your students about the visit ahead of time and ask them to submit questions to the network manager in advance.
3. If the network manager can give a tour of the server room, consider the practicalities. The room might be small and noisy. Is it possible to take half the class at a time and get someone to supervise the students left in the classroom? What can those students be doing while others are getting their tour?

"I DON'T HAVE A NETWORK MANAGER!"

Not everyone reading this will have a network manager on-site to give tours of a physical server room. If this is the case, there are other steps you can take to provide similar experiences:

- Take your class on a visit to a local data centre and ask for a talk from the staff there

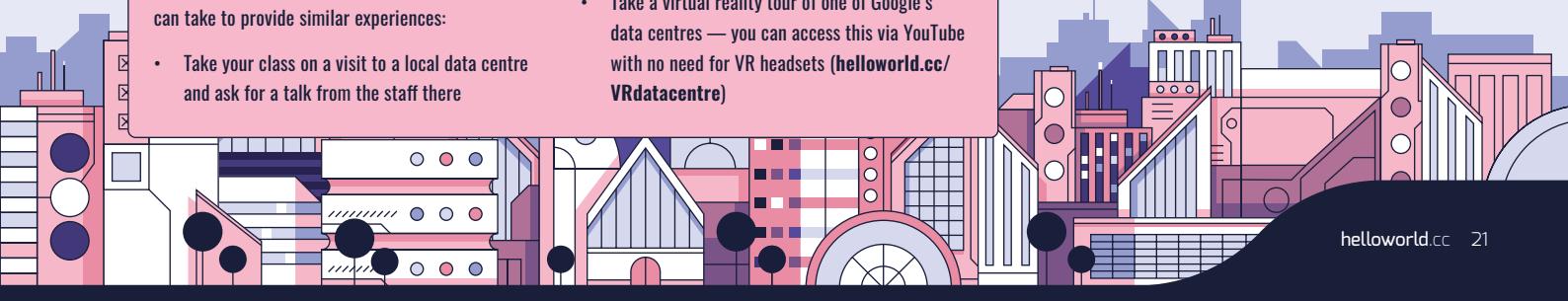
- Industry experts are often willing to come into schools to give talks
- Take a virtual reality tour of one of Google's data centres — you can access this via YouTube with no need for VR headsets (helloworld.cc/VRdatacentre)



BEN GARSIDE

Ben is a learning manager at the Raspberry Pi Foundation. He has worked on the production of the Teach Computing Curriculum and has written online courses for educators including *Introduction to Machine Learning and AI*.

I personally found inviting a network manager into my class a great tool to engage students with networks and bring some quite abstract concepts to life. For other great ideas to make this topic more concrete, read Ben's article on the previous page and our low-down of the best network simulator tools on page 22. (HW)



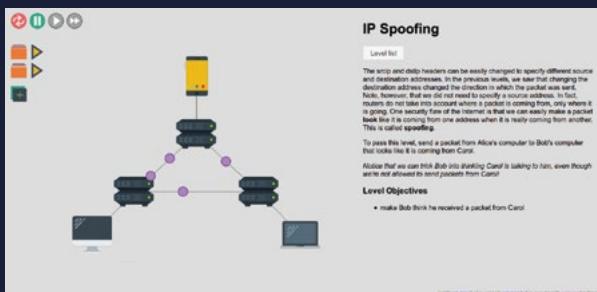
NETWORK SIMULATOR TOOLS

A rundown of some popular network simulator tools educators can use in their classrooms

Networking can be a difficult topic to engage students with. Making a network using physical components is a great option for bringing the subject to life, but this can be a time-consuming and

expensive approach, as well as requiring a lot of space and expertise. Using a network simulator tool is an excellent alternative, so here's a summary of a selection of tools to help you get started.

Thank you very much to Javier de las Heras and Paul Baker for their contributions to this feature. Also, thanks to Paul Powell for his article about FILIUS, which the feature has drawn upon. (HW)



■ A typical activity in the Netsim simulator

CS4G NETSIM netsim.erinn.io

Netsim is a simulator game created by researchers at the University Waterloo, Canada, specifically to teach high-school students the basics of how computer networks function.

ACTIVITIES INCLUDE:

- Depicting how common protocols operate
- Crafting network packets, with a focus on manipulating header fields and injecting them into the network via computers that users control
- Spoofing a source address to steal data
- Inducing a Smurf attack to perform a distributed denial of service

FEATURES:

- Free, open source, and web-based
- Works on tablets, as well as on desktops and laptops
- Specifically created for secondary-school students
- An emphasis on security
- The aim of the game is not to be right or wrong: "While each level [of the game] has a specific success trigger, the crafting of a packet that does not trigger the end of the level is not a failure. It is simply a learning step that gives the player more information about how the Internet works."

(helloworld.cc/atwater2017)

FILIUS helloworld.cc/filiustool

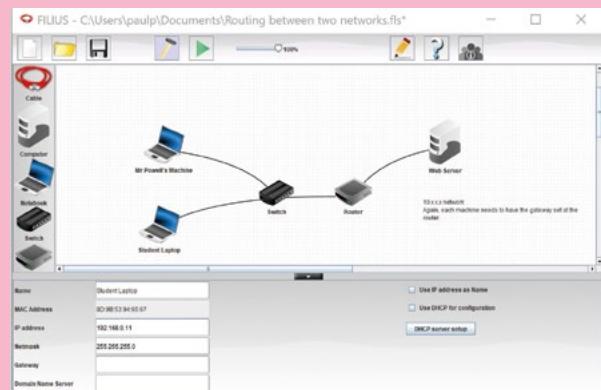
FILIUS is a tool created by researchers at the University of Siegen, Germany, specifically for enhancing computer science lessons on networks for secondary-school students.

ACTIVITIES INCLUDE:

- Designing simple networks with computers, switches, routers, and cables
- Setting up and simulating multiple networks, routing tables, web servers and web browsers, email, and DHCP
- Documentation mode, allowing users to add annotations to their network and group certain parts of their network for a better understanding

FEATURES:

- Free to use, downloadable software
- Developed for secondary-school students, promoting explorative learning
- Teachers can load and save their created networks, so they can prepare them in advance to teach specific concepts
- Comprehensive user manual, including example exercises and solutions (helloworld.cc/filius)



■ Two connected networks in FILIUS, showing client/server communications

CREATELY helloworld.cc/createlynetwork

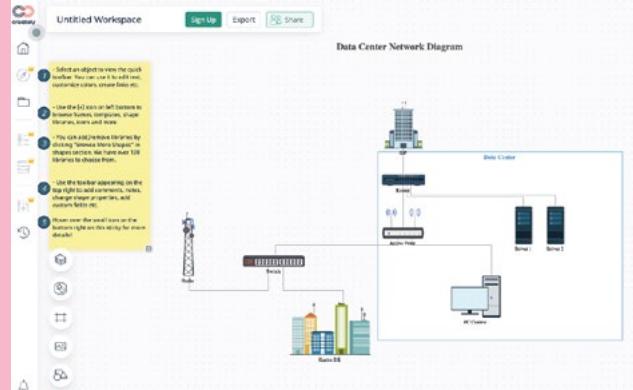
Creately is a visual software program with a network diagram software option. It offers a visualisation tool in which students can design networks, drawing upon a range of network diagramming templates.

ACTIVITIES INCLUDE:

- Designing and visualising network infrastructure of any complexity or scope
- Doesn't offer simulation activities, so is more limited than other tools

FEATURES:

- Free version, with easy registration; also possible to use without an account, but work will not be saved
- Shareable workspace, so that students can collaborate on the same diagram in real time
- Premium versions allow access to more templates and features



■ Creately offers a shareable workspace so students can collaborate

CISCO PACKET TRACER helloworld.cc/packettracer

Cisco Packet Tracer is a powerful industry-standard simulator tool, which can be used for teaching networking, as well as IoT and cybersecurity skills.

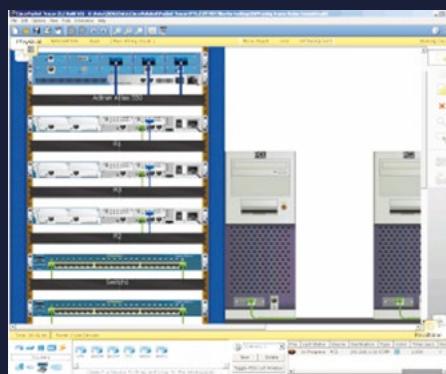
ACTIVITIES INCLUDE:

- Practising building simple and complex networks, from simple star network topologies to networks featuring switches and hubs
- Visualising how a network works, with graphical representations of students' model networks
- Illustrating the four-layer TCP/IP model and how the layers interact
- Animating networks by adding data packets and practising rack, stack, and cabling skills in the virtual lab

FEATURES:

- Free to use, downloadable software
- Free, short online courses are also available to develop teachers' and students' knowledge of using the tool
- A wide community of users (students and industry professionals), including a global Facebook community to share ideas with (helloworld.cc/ciscoFB)

- A steep learning curve, but plenty of YouTube videos available for support, as well as a built-in tutor option in the tool for tips; educators can also save basic PKT files for students to interact with as a starting point
- Can be used right from the basics of star topologies or simple routing through to creating advanced IoT networks and integrating Python code



■ Packet Tracer's physical workspace offers a graphical view of the logical network



HAVE YOU TRIED TURNING IT OFF AND ON AGAIN?

Josh Crossman explores why turning it off and on again may just be the best piece of troubleshooting advice you can give

Have you tried turning it off and on again? It's a common question when troubleshooting computing issues. Whether you have been the person making the suggestion, or the person being told, what was the reaction? Was it laughter, disbelief, frustration, or exasperation? Such reactions often demonstrate what a lot of people think about this advice — that it doesn't work, and is just something people say when they have no other advice to offer.

So why do people find it so hard to believe that turning it off and on again could work? It's a simple instruction which generally requires no technical ability. This can be at odds with some people's perceptions that computing is technical or difficult — an alternative conception we should try to change!

It's also an approach that you can apply to many different issues, and this can make it difficult to accept that switching something off and on again could actually fix your issue.

Finally, it could be that the advice has previously been tried and not worked for the person involved. To a lot of people, this suggestion seems like a joke. Just a little understanding of what is actually happening, though, can demonstrate that turning something off and on again is

actually a useful troubleshooting procedure that can benefit both teachers and learners.

What is happening?

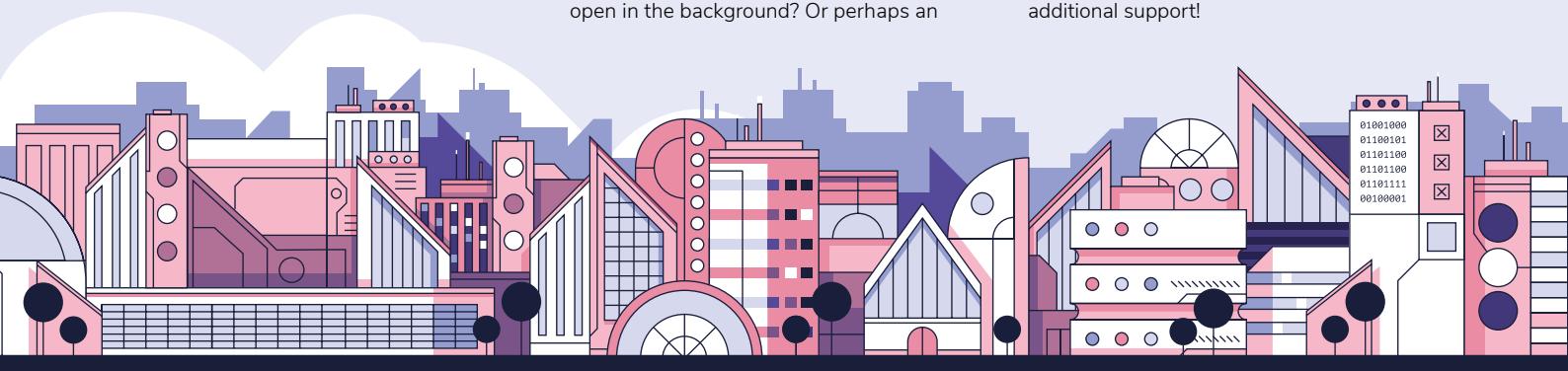
So what are the merits of turning something off and on again, and why should this be the first thing to try if there is an issue with an application or device? Let's say I am experiencing an issue with my laptop and I don't know what the problem is. What I do know is that my laptop is constantly running different back-end processes that make it work in the way I expect it to. If just one of these processes goes awry, it can have a cascading effect on the rest, causing issues. By switching my laptop off and on again, I can ensure that it is rebooting all the standard operating processes from a clean start. Restarting the processes from the beginning could well mean that whatever was causing the issue will work perfectly this time around.

In today's digital world, I could be running many different pieces of software at the same time. For example, I regularly have email applications open, on top of the many browser tabs and office applications I use for planning and creating resources. It can be difficult to keep track of what is open, and therefore what could be causing the issue. Is it one of the applications, many of which don't fully quit when they are closed, open in the background? Or perhaps an

automatic update is unable to progress and is spinning endlessly? Whatever is causing the issue, turning the laptop off and on again will stop all those background applications running.

Taking the step to switch something off and on again can feel a bit drastic at times. When I was a teacher, I would rarely shut my laptop down completely at the end of the day, as the extra time needed to reopen all the browser tabs and applications I needed once the laptop was turned back on felt like wasted time. However, once the principle behind turning something off and on again has been understood, more specific approaches can be taken. For example, instead of turning off the whole laptop, I might try closing specific applications to determine whether they could be causing the problem. It's the same principle, just being applied at a more localised level.

It doesn't matter where I start, whether it's at an application level or at the device level. What matters is that I have an understanding that by turning something off, or shutting something down, I am forcing that thing to restart all the processes from the beginning. This means that if the issue isn't resolved, I can now recognise that the problem could be more than just a process gone awry, and I might need to call in some additional support!





Why does this matter?

Time is precious. When I was a teacher, I led computing throughout my primary school. As a result, I was regularly interrupted while teaching because something computing-related in another classroom wasn't working as expected. This led to me spending a lot of time troubleshooting issues throughout the school — time that could have been better spent on other things, such as teaching or curriculum planning.

People are often afraid to try to fix issues with computing devices and applications because it is seen as difficult, or because they are afraid of doing more harm than good. However, in my experience, empowering other teachers and support staff to attempt their own resolution of their computing issues is a powerful strategy for a computing lead. It models resilience and simple problem-solving skills to learners. The learners in my class soon became accustomed to trying to fix their own issues before coming to an adult in the class! It can also support teachers in developing

“ EMPOWERING TEACHERS TO RESOLVE THEIR ISSUES IS A POWERFUL STRATEGY

their own subject knowledge about how a computer works, including the interaction between applications and devices; the role of a device in the wider network; and how the parts of a computer interact with each other. Building this subject knowledge will allow teachers to instruct with confidence, and ensure they are more able and prepared to address any alternative conceptions their learners have.

Removing the stigma that turning something off and on again is just a joke can ensure it becomes a valid troubleshooting step for educators and learners alike. And if something this simple can resolve an issue, perhaps it can begin to change perceptions that computing is inherently difficult. So tell me, have you tried turning it off and on again? **(HW)**



JOSH CROSSMAN

Josh is a programme coordinator at the Raspberry Pi Foundation, working across programmes such as the Teach Computing Curriculum and Hello World.



COMPUTING CURRICULUM FUNDAMENTALS

Why computing systems should be at the heart of computing curriculum design

Whether you plan lessons on a topic, develop curriculum content, or even write curriculum policy, you have to make choices. What are you going to include, and what is less of a priority? You have to consider time constraints and access to resources. You might have to consider prior learning and pupil interests. You probably also have to consider the wider curriculum context. Well, here is my first principle to help you: computing systems should be the foundation of your computing curriculum.

As a teacher, when I first began writing computing lesson plans for children aged nine to ten, I started with programming. This was a very visual entry into computing education, and children were excited to create projects that were familiar to them, such as games and animations. However, as my understanding of computing grew, I realised that something was missing.



SWAY GRANTHAM

Sway is a senior learning manager at the Raspberry Pi Foundation. She leads a team developing computing resources for primary teachers (@SwayGrantham).

A computing systems epiphany

Learners could explain what an algorithm was, as well as explain that a program was 'a set of instructions that runs on a computer to tell it what to do'. Both of these met the curriculum needs, but I wasn't convinced that learners could link these two concepts together. Could they connect what they were doing on a floor robot to the computing systems around them? Did they understand what a computer was? Well, I asked them, to see what they'd say.

According to my class, a computer was:

- A piece of technology
- A keyboard and a screen
- A search engine
- A machine used for work
- A metal brain
- A machine with a keyboard
- An information device
- Electric

This very simple question highlighted a wealth of alternative conceptions about programming and computing systems. The other commonality was describing the computer's function, as if to define it, we just need to know what it does. This view of a definition greatly reduces the potential a computer has for anything beyond personal use. My learners now had two discrete chunks of knowledge: how to program a floor robot, and that laptops were computers. However, this learning began to

seem disjointed without a bridge to connect the different chunks. Learners needed to have a concrete, conceptual understanding of what a computer is before they could start to comprehend the more abstract role of a program in that system.

Beyond the experiences of my young learners, we see examples of a lack of understanding about computing systems all the time in society. Many competent users of software are able to regularly complete the tasks they need to, but if something doesn't work one day, they do not know how to find a solution. Equally, many people enjoy exploring digital making projects, yet if they want to personalise the project, they don't know what they can or can't change. Knowledge of computing systems empowers people to take control of the technology they own, and not just consume it.

Planning computing content today

Both of these examples highlight the importance of introducing computing systems, both as life skills and as support for developing other areas of computing. More recently, my team and I have been creating 100 hours of curriculum content with a non-profit organisation called Amala, to give refugee learners, who may never have used technology before, enough understanding to build a website that encourages social change. While we know that it will need to include some foundational knowledge of computing



systems, we need to first consider the core content that learners must understand to achieve the end goal, such as:

- Web page creation
- HTML/CSS/JavaScript
- Project management
- Project development

These areas of learning are a great place to start as, undeniably, learners aren't going to be able to build a website without knowing the process of creating a website, the languages used to create web pages, or the project management skills needed to see a project from start to finish. This could be the entirety of the content, but instead, I encourage you to think back to those children who could program but didn't know what devices programs could run on. We need to connect the core content to that foundational content: how is building a website related to computing systems?

Prior knowledge

All learning is built on prior knowledge, even if that prior knowledge has been gained through life experience and not formal education. To build a website, we must know how to type and use a mouse. We need to know what a website is, why people use them, and what sort of media is found on them. Beyond that, we need to know how the files we are creating are shared with other people. We need to understand how a computer can communicate with another computer, and what the process is to make that happen. None of this learning is the core content of building a website, but if you tried to build a website without understanding these things, it would be difficult to do.



■ Learners need a concrete understanding of what a computer is before they can comprehend the role of a program in that system

As the learners in the Amala project might have no prior experience using technology, we needed to ensure that this foundational computing systems content was built into the learning sequence — things such as:

- Recognising digital devices
- Decomposing computing systems
- Digital painting (mouse skills)
- Combining text and images (desktop publishing)
- Networks and the internet
- Internet searching

By incorporating this content into the learning sequence, we ensure that learners do not just learn a process for creating a website. They understand the impact of the choices they make when building a website, they have the skills to implement their ideas, and they can connect their understanding to solve any unexpected challenges they come across

along the way. This more holistic approach should support knowledge transfer and offer a much broader range of opportunities.

Whatever your curriculum requires, you will have the core content you need to teach. This could be the requirements of your standardised curriculum or the specific project you're trying to build, or it could be the aspirations that you have for your students. However, rather than stopping at that part of your learning sequence, take a step back and consider the prior knowledge you're connecting to. I expect you will find that computing systems is what you need if you are to ensure learners' new knowledge has a solid foundation. **(HW)**

We will revisit the Amala project in future issues, so look out for how our learners are getting on in the pilot of the Using Online Technologies to Create Change course in Kakuma camp, Kenya.



WEAVING THE REAL-WORLD INTERNET

Andy Storey introduces activities to highlight how the internet works

Whether it is a news story about a distributed denial-of-service (DDoS) attack or a question about the latest social media trend, there is plenty about this wonderful subject to capture the interest of our pupils. Although there is no specific mention of the internet in England's computing curriculum, we do have a responsibility to ensure that our learners become digitally literate.

But without secure and practical knowledge of how the internet works, I'm not sure they can be active participants in a digital world. Here are some activities that will increase the breadth and depth of lower-secondary learners' digital literacy skills, while also laying the groundwork for later studies. **[HW]**



ANDY STOREY

Andy's interest in computing dates back to programming an Acorn Electron when he was eight years old. He is a recently qualified teacher, working in an oversubscribed secondary school in the south of England. He has worked in IT throughout his career in both the public and private sectors (@**mrandystorey**).

ACTIVITY NAME	DESCRIPTION
Learn how to google like a pro	A search engine such as Google is a very powerful tool, but most people don't really know how to use search engines properly. Here are some tips that your students might not know! <ul style="list-style-type: none"> • Use quotes around search terms to perform exact searches • Add a dash right before a word to exclude it from the search • Use the site: operator to restrict results to a particular website • Use two full stops in between two numbers to search for a range helloworld.cc/searchoperators is an excellent introduction to Google search operators and parameters, including great examples for some of the tips above.
Port scanner in Python	Live-code a basic port scanner, walking pupils through this tool that's used in penetration testing. 'Hacking' is a term that learners will regularly come across, so it's useful to be able to walk them through a rudimentary approach using Python. The code at helloworld.cc/portscanner is straightforward for pupils who have a basic understanding of Python, and reinforces basic programming constructs such as sequence, selection, and iteration. Be sure to either use this on your local machine or consult with your network manager before scanning live servers.
What is DNS?	Having a basic understanding of DNS (Domain Name System) will make a social engineering technique such as pharming much easier to understand and teach. The slides at helloworld.cc/DNS provide an introduction.
Undersea cables	Many internet connections have to cross oceans. Get pupils to find their way from their current location using fibre-optic cables running under the sea. Try submarinecablemap.com , an excellent resource that illustrates how all continents around the world are interconnected.
What happens when you type in a URL?	You can support pupils' understanding of why some websites are slow by explaining that when your browser requests a URL, the server responds with a copy of the web page (the HTTPS request/response cycle), and then does that all over again for every resource on that page. Get your class to compile a top-ten list of websites they use and run them through helloworld.cc/pagespeed . Then investigate how Google ranks them by performance, accessibility best practices, and other search engine optimisation (SEO) principles.
Create a web server	If students want to get more adventurous with their World Wide Web knowledge, Python has a built-in web server that can be used in class (helloworld.cc/HTTPservers). Students will see exactly what happens when they type in a URL on their own machine as the requests are displayed live. helloworld.cc/HTTPResource is an easy-to-understand resource that explains how to start a web server on your own computer.

THE POWER OF THE PICO

Tina Fountain shares her personal experience of using Raspberry Pi Pico to bring theory to life

Since I first discovered the power of the BBC micro:bit for engaging students, I have become a little obsessed with it. In fact, I have a reputation on the international circuit for being the 'queen of the micro:bit'. Over time, I have developed a wide range of ways to incorporate physical computing into my classroom, to help bring theory to life. For a student, adding power to something that they have programmed is a compelling experience. However, one frustrating aspect of micro:bits is their cost, meaning it's never been possible to allow students to take projects home or keep them on display. The introduction of the RP2040 chip and low-cost Raspberry Pi Picos have therefore piqued my interest. I must confess to being rather late to the Pico party, but I have quickly realised that these could be the answer to bringing the theory of computing systems to life.

With the exception of a single on-board LED, Pico has no inputs or outputs, meaning you need to add simple output components. Because of this, I decided to test out Pico

with my AS-level (aged 16–17) group first, before jumping in with younger students. I purchased 0.96 OLED screens, LEDs with built-in resistors for ease, push button modules, crocodile clips, and jumper wires. Each kit came to less than \$11, making it very cost-effective to have one per student. This is my school's first year of computing, and I have very few additional resources, so I have had to get quite creative in the set-up. We have managed to build some very rudimentary systems with paper clips, staples, glue guns, and copper tape. I would highly recommend purchasing breadboards and starter kits instead!

We started the year by building simple representations of a computer system with inputs and outputs, as we had already covered some of the basic theory. Learning about computers as you program them opens a lot of opportunities for challenging students to consider the bigger picture. This in turn allows them to make early connections between what they are programming and how the hardware will handle those processes. For example, you can discuss how the data will be held in RAM during execution, what processes are involved in producing the outputs, and looking at how improving the code they write impacts the processing. I love this more holistic approach to teaching theory.

My class then moved on to data representation, and again, a physical computing approach brought the topic to life. We built denary-to-binary converters using LEDs to represent the binary output,

and attached buttons to increase or decrease the denary value to be converted into binary. This allowed us to then explore programming libraries and look at how images can be represented using byte arrays. Using the OLED screens allowed students to create simple menus and expand their programs to cover a range of data representation topics. The students found the challenges exciting and experimented at their own pace, sharing ideas and learning from each other.

Having learnt the basics of programming Pico, I challenged students to work together to build and program a logic gate simulator. They were able to set up the build and program the gates in a 90-minute lesson. They enjoyed the challenge, and giving them free rein in the development meant they all had different ideas that they implemented. Testing each other's ideas also allowed students to see alternative ways of tackling the same problem. Building a basic simulator meant that students could easily get to grips with the concept of logic gates, discovering how a computer system works and how it executes instructions.

As a teacher, it is not possible to be an expert in every single computer science topic. Using Pico for the first time this year means I could let my students become the experts, with one particularly passionate student creating some wonderful projects in his own time, which he happily shares with the rest of the group. Perhaps one day I'll be the 'queen of Pico' — for now, though, I'll just encourage you to try it out to bring theory to life in your classroom! **(HW)**



TINA FOUNTAIN

Tina teaches computing in an international school in Madrid. She is also part of the steering committee for the South East Asia Computer Science Teachers' Association (@[TinaFountain7](#)).



HOW DOES THIS WORK?

The input–process–output model can support young learners to understand the technology around them

Sometimes, when you ask a question of young learners, you are pretty sure you know what their response will be. Other times, they take you by surprise: you blink, try to keep calm, and begin unpicking exactly where that answer came from. This is what happened when Sway asked a class of seven- to eight-year-olds how automatic doors worked. One child rationally explained that a person was watching the door on a camera, and when someone approached the door, they pressed a button and the doors opened. Simple.

While this is a perfectly reasonable explanation, it is not the correct one. Instead, it highlights that our learners often have no idea how information technology (IT) actually works, what it does, or why something is happening. This makes it seem magical at best, and sentient at worst. This article will

explore some common misconceptions that learners hold about computing systems, and how you can use the input–process–output (IPO) model to support learners in making sense of the IT around them.

The IPO model

All computers work with inputs, processes, and outputs (see Figure 1). All computers accept inputs, which are entered into or received by a computer. They can be generated in many ways, including by a user pressing a key on a keyboard, or a computer receiving a signal from another device. The process then determines what the computer does with that input. It can process the same input in different ways, depending on the program running. The output is how the computer finally presents the results of the process. It can return the results to the user

in many ways, such as displaying text on a screen, creating printed materials, or playing a sound from a speaker.

In today's connected world, it's easy to overlook the processes taking place in devices that learners don't immediately recognise as computer systems, such as pedestrian crossings or washing machines. This can lead to learners developing alternative conceptions about what is happening, making it harder for them to apply their understanding of programming or input and output devices as they gain more knowledge. We can't build knowledge on insecure foundations, so the sooner we identify these misconceptions, the better.

Does a lamp have a computer inside it?

As we start to pay attention to the world around us, we begin to recognise different groups of objects that have similar properties, such as natural or manufactured, and mechanical or electrical. However, as these objects become more complex, it can be hard to tell which groups they belong to. This ambiguity can make learners overgeneralise their understanding of how something works. Taking time to break this down with the IPO model allows learners to reflect on their assumptions.

Let's imagine a desk lamp. Does it have an input? Yes — I press a button to trigger what I want to happen. Does it have an output? Yes — the light turns on. Now comes the important part: is there a process? No — there is no program receiving data that the button has been pressed. Instead, the switch

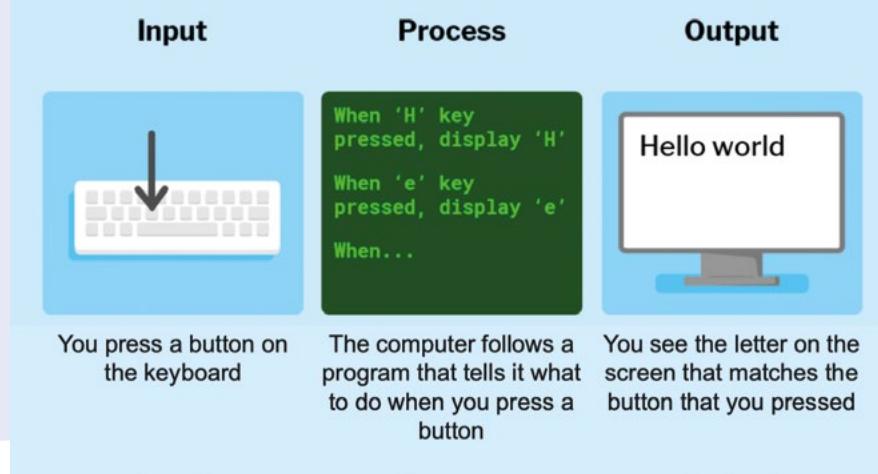


Figure 1 A simple example of the IPO model



on the lamp creates a circuit for the electricity to flow through, allowing the bulb to light. Therefore, most lights do not have computers inside of them.

Computers are really clever

The feeling we have that computers are magical, before we start to understand how they work, is often reinforced when the device can do something we do not know how to do ourselves. One of the most prevalent and early alternative conceptions that learners hold about computers is that they are 'really clever'.

To address this, let's consider looking for information on a website to answer a question. What is the input? Using the keyboard to type in keywords that tell the computer what I'd like to know. The search engine's computer then processes this data by running a program to find relevant information. What is the output? A website showing a list of other websites on my screen. Do I have the answer to my question?

stored locally on the device you are using, and what's uploaded to the internet. I've found this particularly challenging with certain apps on tablets that may also back up online.

To unpick this, it's important to start considering larger and more complex systems, such as ATMs. The input (the data from the keypad) and the output (the information displayed on the screen) are clear. However, much of the process is not happening on the computer within the ATM — it's using the internet. The computer in the ATM sends the input data through the internet to the server at the cardholder's bank, to check it's correct. This is the process. Then the server sends back the output data to show the outcome on the screen. The first data processed will check whether the PIN number is accurate, but each instruction after that will begin the process again. Even if learners can't accurately recognise what's happening on a device and what's happening online, having these IPO conversations can support them in thinking about what's

FURTHER RESOURCES

THE RASPBERRY PI FOUNDATION'S FREE ONLINE COURSES:

- *Teach Computing Systems and Networks to 5- to 11-year-olds:* helloworld.cc/systemscourse
- *Get Started Teaching Computing in Primary Schools: Preparing to teach 5–11 year olds:* helloworld.cc/primarycourse

FREE LESSONS AND ACTIVITIES IN THE TEACH COMPUTING CURRICULUM:

- Connecting computers (ages 7–8): helloworld.cc/tccsystems1
- Systems and searching (ages 9–10): helloworld.cc/tccsystems2



JOSH CROSSMAN AND SWAY GRANTHAM

Josh is a programme coordinator at the Raspberry Pi Foundation, working across programmes such as the Teach Computing Curriculum and Hello World. Sway is a senior learning manager at the Raspberry Pi Foundation. She leads a team developing computing resources for primary teachers. Josh and Sway are both former primary teachers (@SwayGrantham).

“ WITHOUT UNDERSTANDING HOW A SYSTEM WORKS, IT'S EASY TO MAKE ASSUMPTIONS

Most often, no. I now have to go to each web page and look for the answers I need. Taking learners through each step in the model highlights how much of the process is reliant on human interaction to work, and how computers are only as powerful as the humans that use and program them.

IPO takes place on one device

Without understanding how a system works, it can be very easy to make assumptions. One afternoon, the internet went down at my (Josh's) school. My class, however, didn't believe me! Why? Because the interactive whiteboard was still working. These assumptions become more important when we begin considering personal data, what's

happening before they create content and potentially share it online.

From programming, to collecting data from sensors, to recognising technology around us, the IPO model applies to almost all aspects of computing. Starting activities with the question, 'How does this work?' can evolve into learners recognising the many and varied IPO systems in the world around them. You can then get creative, letting learners invent imaginary systems to put the IPO model into practice (see helloworld.cc/tccsystems1). Initially, the processes will be assumptions, but as learners' experiences grow, these approaches become a chance for them to imagine the computer systems that will change the world. (HW)



COMPUTER NETWORKS: A FRESH APPROACH TO ONLINE SAFETY?

Sway Grantham argues that we should progress from teaching hard rules about online safety to developing informed decisions with knowledge of computer networks

Every day, teachers experience the implications of learners making mistakes as digital citizens. This is often explained through the idea of new technologies, and even though online safety education is now commonplace in the majority of schools, it is still almost expected that learners will make these mistakes. In the English national curriculum, though, it is mandatory to teach computer networks, and this topic area can give young people the skills, knowledge, and behaviours they need to make informed decisions that keep them safe online, if only we knew how to wield it!

Taking risks

To know how a networking education can help learners make informed decisions, we first need to understand how we learn to mitigate risks and keep ourselves safe. An example that's familiar to us all is crossing a road safely. When crossing a road, I need to use prior knowledge and experience: how long will it be before that car gets here? How far is it to cross the road? How quickly

can I move? I use this knowledge to make a choice: will I cross the road now, or should I wait? The more times I do this, the better I get at making the decision accurately. Even better, for the first few years of my life, I have the decisions modelled to me by an accompanying adult; they make sure I am not taking unnecessary risks. All these experiences help me to avoid a consequence I can easily imagine: me, or someone else, getting hurt.

If we break this process down, to learn how to take a risk and make an informed choice, we have:

- **Prior knowledge and experience:** what did I think about the last time I made this choice?
- **Consequences:** what will happen if this is the wrong choice?
- **Role models:** what did the trusted adult show me?

Prior knowledge and experience

Understanding how the internet works can empower students to think about their choices. However, when discussing online safety, we so often focus on hard rules and forget the why and the how. Why might it be risky to upload a photo to a social media

site? What happens when I upload that photo? Where does it go? Who owns it? Who can see it?

When we explain that the internet is not floating in the sky (or 'the cloud') and acknowledge that it comprises real-life computers stored and owned by people around the world, uploading a photo becomes more real. Depending on where you upload it, the photo may be stored in another country that has different laws about what the person who owns the computer (server) can do with it and how long they can keep it for. The photo will probably also be backed up to keep it safe in case there is a problem with that original computer. Again, it could be saved in another country which is subject to different rules.

Consequences

By this point, you can see learners starting to understand how beyond their control that photo now is. We can then follow this up with conversations about how, when you delete a photo from a profile, it will still be saved on some of those computers and in some of those backups. Even if you only uploaded it to a private social media account, it is still saved on someone else's





© christin late/EyEm/stock.adobe.com

To make informed decisions, we first need to understand how we learn to mitigate risks and keep ourselves safe

computer. That computer might be hacked, or its data (including your photo) might be stolen or sold, and then other people might also have access to it. The more learners understand about how data is transmitted through the internet, how websites are stored and accessed, and what they are agreeing to when they give someone else that photo, the more they are able to make informed choices.

points, are other examples that give learners insights that can empower them to change their online behaviour and provide a clearer connection for them between the action and the consequence.

Role models

Using technology should not be seen as an exclusively dangerous activity. Online communities, for example, allow people to

will be saved forever' to 'Do you mind if the fact that you [love this band] is public information?' or 'Does it matter if the other people who love the same band as you are a different age to you?', then learners can review each scenario more critically using their prior knowledge of how networks work and what the consequences of sharing that information could be.

Just like that parent who first started discussing with their children when it was and wasn't OK to cross the road, you can be that role model for the learners in your class. You can encourage them to think critically about how computer networks work, what they can and can't control, and whether information should be shared or not.

So now I encourage you to reflect: how do you teach online safety? Do you teach fixed rules that work in specific situations? Or is safety integrated with pupils' understanding of how computing systems and the internet work? Should it be? After all, convincing a child that they need to be careful crossing a road is going to be a challenge if they have never seen a car. [\(HW\)](#)

CONVINCING A CHILD THAT THEY NEED TO BE CAREFUL CROSSING A ROAD WILL BE A CHALLENGE IF THEY'VE NEVER SEEN A CAR

This is just one example of how learning about networks can give learners knowledge and experience about how to make informed choices when they use the internet. Learning how companies record data to put advertisements in a place where you're more likely to click on them, or how games track what time of day to offer double experience

connect with a wide range of people with fewer economic restraints. You can discuss a band you love with the ten other people in the world who have heard of them. You can feel that you belong and that there are other people like you. If we switch the narrative from 'Don't share', 'Don't talk to strangers', and 'Anything you say online



SWAY GRANTHAM

Sway is a senior learning manager at the Raspberry Pi Foundation. She leads a team developing computing resources for primary teachers ([@SwayGrantham](#)).



GET NETWORKING

Three people explain how they use networking in their jobs, and share tips for students interested in following a similar path

Providing students with context about why they are learning something is a valuable engagement tool. Here, we profile three individuals who use networking skills and knowledge in their jobs. You can use

these profiles to help inspire your students when introducing the topic of networks, helping them to understand the variety of career paths they could take with a grounding in the subject. **(HW)**

THE NETWORK ENGINEER



NAME: Jamie Lawrence
JOB ROLE: Network engineer at Arm Ltd., Cambridge, UK

WHAT DID YOU STUDY AND WHERE: BSc in digital and technology solutions (network engineering) at the University of Suffolk, UK.

DESCRIBE YOUR JOB AND HOW IT RELATES TO NETWORKING: I work in Arm's global networks team as a network engineer. We are responsible for designing, building, maintaining, and operating Arm's network across the world. This includes the networks within our offices, to ensure our staff can work (local area networks and campus area networks); the data centre networks that connect our storage, server, and compute systems; and the wide area network connecting our sites

together using internet circuits or private circuits (using a technology called Multiprotocol Label Switching).

My role largely encompasses working on projects to design and build new systems and services for the operation of the network. At the moment, this includes implementing a new monitoring system called Cisco ThousandEyes, helping to deploy a software-defined wide area network solution, and automating our network processes using a mixture of Python and Ansible coding languages. The network is the foundation of any IT solution, so our role ensures that Arm can operate and develop our technology for our customers.

WHAT DO YOU ENJOY ABOUT YOUR ROLE: I really enjoy working with the latest Cisco equipment and systems, which means I'm continually learning new technologies and skills. I also enjoy the mixture of hands-on and desk-based work, including visiting our data centres and other offices to perform maintenance

and installs. Finally, I get to work with some extremely talented engineers, which is certainly helping me on my professional journey.

DO YOU HAVE ANY ADVICE FOR STUDENTS WHO ARE INTERESTED IN A SIMILAR CAREER PATH: If you can, get some work experience within an IT department. At age 15, I did work experience at a university's IT department and I learnt a lot about the roles available, the processes involved in running an IT service, and the infrastructure behind it.

I would also say that customer service skills are a big part of IT: being able to communicate with a customer in non-technical language and then speaking technically with colleagues. If you're interested in networking, I would recommend Cisco Networking Academy (netacad.com). It has lots of free, short introductory courses, as well as its Packet Tracer program, where you can build and simulate networks.



THE ICT ENGINEER



NAME: Stuart Ramsay

JOB ROLE: ICT engineer at
Durham County Council, UK

WHAT DID YOU STUDY AND WHERE: BSc (Hons) in computing at the University of Sunderland, UK.

DESCRIBE YOUR JOB AND HOW YOU USE NETWORKING:

I work in Durham County Council's IT department, in the section that provides managed IT services for local schools. For the past six years, I have been based in a small team in one of the country's best state secondary schools (2000 users, 1000 devices, and 20 servers) looking after all aspects of its IT infrastructure and technical support. My prior experience in installing physical

networks (including copper, fibre-optic cabling, and associated hardware) led me to progress into not only maintaining and troubleshooting the schools' networks but being responsible for delivering larger-scale on-site infrastructure projects. This can include the configuration and installation of a completely new network with network switches, data segmentation/virtual local area networks, a replacement Wi-Fi network, web filtering, and firewall solutions. I've also replaced the ageing server farm with new servers based on a hypervisor cluster system and implemented both on-site- and cloud-based backups.

WHAT DO YOU ENJOY ABOUT YOUR ROLE: I really enjoy the variety of my role; no two days tend to be the same. Some days can be entirely desk-based, carrying out small administrative changes; other days can be completely hands-on with a practical

requirement to ensure there is no disruption to the day.

DO YOU HAVE ANY ADVICE FOR STUDENTS WHO ARE INTERESTED IN A SIMILAR CAREER PATH:

Have a firm grasp of the fundamentals, such as the OSI model and IP networking. It's impossible to know everything when running a network, so ask as many questions as you need. Also, take any opportunity to learn something new when you can, as you never know what doors it may open.

I would also recommend learning to work amongst people, and how to communicate effectively with others. This sort of soft skill is often overlooked in our profession. Finally, hands-on experience, vocational qualifications, and technical certifications are just as valuable for your learning and career progression as traditional degrees.

THE PROGRAM MANAGER



NAME: Jaclyn Lazarus

JOB ROLE: Technology development program manager at
Verizon, New Jersey, USA

WHAT DID YOU STUDY AND WHERE: BS in Biomedical Engineering at Rutgers University, New Brunswick, USA, and MS in Enterprise Project Management at Stevens Institute of Technology, Hoboken, USA.

DESCRIBE YOUR JOB AND HOW IT RELATES TO NETWORKING: I work in Verizon's technology and product development organisation as a program manager. In my role, I work with 5G technology to help develop products for sports and entertainment events.

Last year, a couple of my major projects were working with 5G technology to bring fans up close and personal to events such as the Super Bowl Halftime Show and the Oscars, using augmented reality (AR). My job is also to verify that the 5G network at these events is optimal for streaming.

WHAT DO YOU ENJOY ABOUT YOUR ROLE: What is exciting about my job is that there is no blueprint — our team is collaborative and creative. I am able to help create products that can impact the world. I have been able to bring fans close to the action at high-profile events such as the Super Bowl, the Oscars, and The Game Awards to name a few. I also enjoy learning about new technologies and working with extremely talented people every day.

DO YOU HAVE ANY ADVICE FOR STUDENTS WHO ARE INTERESTED IN A SIMILAR CAREER PATH: My first piece of advice is to take every opportunity you can get in school; get involved with research, clubs, and career services. Second, apply for internships as soon as possible to get professional experience. My first internship was unrelated to what I do today, but was important for getting my foot in the door and learning about how corporations operate. Finally, go with your gut! I decided to apply for Verizon even though my background was in Biomedical Engineering, which is not a traditional major for working in network engineering. I saw that Verizon offered a leadership development program and decided to go for it. Ten years later, I am still here, and still working with awesome people!



HAS THE CLOUD CHANGED THE WAY WE TEACH NETWORKING?

Ben Hall explores the cloud and how educators have adapted their approach to teaching networks

When I first started teaching computing, one of my go-to lessons was a network safari. This involved seeking out the school's network infrastructure and uncovering the purpose of each device and how they all linked together. At the time, the set-up at the school I worked in was pretty typical — a line from outside, a router, a central server, a switch, and an IT suite of around 30 desktop computers. There were some wireless access points, but the iPad revolution had yet to take place.

Exploring this was straightforward, and having the physical devices to show learners really helped: the server handled all the requests for documents, web requests went out via the router, all the computers were physically connected with wires via the switch, and software was installed on the computer you would be using. Teachers could further explain the relationship between each device with role plays and more abstract diagrams, but still the principles were fairly simple.

Gradually, things began to change: desktop devices became laptops, iPads appeared, and the increased load on the wireless network meant that wireless access

points suddenly popped up in all corners of the school. Students did more learning on web-based applications, and the previously huge server progressively shrank in size and was no longer physically connected to devices. At the same time, learners switched from saving all their work on a network server to saving some, and then eventually all of it, on 'the cloud'.

Addressing alternative conceptions

With these changes in mind, explaining the fundamentals of network architecture becomes harder; devices are hidden from sight, and the scale of the cloud is very difficult to comprehend. There are also a couple of pretty hefty alternative conceptions that many learners (and teachers) have which can get in the way:

1. The cloud is in the sky
2. When you can't access a website, the Wi-Fi is down

To address these alternative conceptions, we have to zoom out a bit. First, it can be useful to explore why decentralised IT infrastructure is a good idea. Using the traditional network model of a centralised server in a school, ask learners to consider the implications of a failure of that server. What could cause it, how much would it

cost to replace it, and who would need to pay for it? This could be a basis for a discussion around why more and more people pay companies such as Microsoft and Google to store data, and use applications on their servers instead of running and maintaining their own.

Once learners understand the benefits of decentralisation, you can then challenge the alternative conception that the name 'the cloud' brings — namely, that the cloud is **not** in the sky. In its simplest terms, the cloud is a network like any other — it's just a lot bigger and a lot more distributed. The equipment you would have previously found in a school is now part of the infrastructure of the internet, but on a much bigger scale. Cables under the ground and sea transport data to huge warehouses full of computers via routers and switches, just as the old server in a cupboard communicated with local-network computers.

The second alternative conception can be harder to address. For whatever reason, the term 'Wi-Fi' has become synonymous with 'network'. In reality, the Wi-Fi-connection part of a network is a tiny part — it's just the connection between your device and the wireless access point you are connected to. More often than not, when you can't access the network via Wi-Fi, the problem will be elsewhere. It could be that:





1. Your device is not connected to the wireless network (but the Wi-Fi is working)
2. The service you are trying to access is unavailable (but the Wi-Fi is working)
3. There is no internet connection to your network (but the Wi-Fi is working)
4. You have a power cut (the Wi-Fi is down, but so is everything else)

Only if the wireless access point is failing to send a signal can you legitimately say that the Wi-Fi is down. Once learners understand that Wi-Fi is a tiny part of a huge

page 20 for more), or you could modernise the network safari so that it works on a wider scale. If you choose the latter approach, there will be at least one device in your school that you can show; somewhere there will be a line in and there will be a router sending the data out to wireless access points. Beyond the walls of your school, you'll need to be a bit more creative!

One of my favourite videos explaining how the internet works is *A Packet's Tale* (helloworld.cc/packetstale). You could show learners how a request goes from



BEN HALL

Ben is a learning manager at the Raspberry Pi Foundation. He is a CAS Master Teacher and a Raspberry Pi Certified Educator (@hengehall).

such as Google (helloworld.cc/googledata), or take a virtual tour of a data centre (helloworld.cc/datacentretour).

However you choose to introduce these ideas to learners, the key is to remember that, although they are now harder to comprehend, the basics of networking have not changed since the first two computers were connected together. What has changed, and what learners really need to understand, is the size and reach of networks and the enormous possibilities they offer for everyone. (HW)

“ DEVICES ARE NOW HIDDEN AND THE SCALE OF THE CLOUD IS HARD TO COMPREHEND

infrastructure, you have a better chance of explaining how it all works together.

Bringing the cloud to life

To make cloud computing relatable, you can try a combination of approaches. You could start small, explaining the traditional school architecture using concrete resources (see

a local device out to a local hub, then a national hub, then under the sea, through another series of routers and hubs to its destination. You could also show learners submarinecablemap.com, which shows the comprehensive global network of undersea cables. You could examine maps that show the location of data centres of companies



SAFE NETWORKING

Mike Smith explores IP addresses and looks at what you can do to protect your devices

I am sure you have seen someone on a TV show use an email to find an IP (internet protocol) address, and then find someone and hack their computer. Is that possible? To explore this, we first need to look at what IP addresses are and cover some networking basics — and then we can look at what you should do to protect your devices.

Types of IP address

When you google, 'What is my IP address?', you will actually get the answer to 'What is my public IP address?', which is the IP address of the home router provided by your internet service provider (ISP). The most common types of IP address are public, private, and localhost.

142.250.178.4 is a public IP address; if you enter it into your browser address bar, you will find it belongs to Google. Your public IP address belongs to your home router and is used to communicate on the



MIKE SMITH

Mike is the CTO of remote.it. He has 30 years' experience in integrated circuit design, CAD, and systems. Mike was previously CTO at iReady and Metaram (sold to Google).

internet. If you have several devices using your internet connection, they all share the same public IP address.

In contrast, 10.0.0.1 is a private IP address. In fact, all IP addresses of the form 10.x.x.x or 192.168.x.x, where x can be between 0 and 255, are private IP addresses. Every device on your home network has its own private IP address assigned by your home router. These addresses work only on your local network — for example, in your home or school.

for about 4.3 billion devices. Now that we have surpassed that number of devices, we have to share private IP addresses. Your home router handles the translation of your internet traffic from a public IP address outside your network to a private IP address inside your network.

Wouldn't you always want a private IP address? You might think so, but you can't talk to computers outside your network this way. If you want to make a connection from inside your network to outside your

WOULD YOU LEAVE YOUR FRONT DOOR OPEN IN A SAFE NEIGHBOURHOOD? WHAT ABOUT IN A BAD NEIGHBOURHOOD?

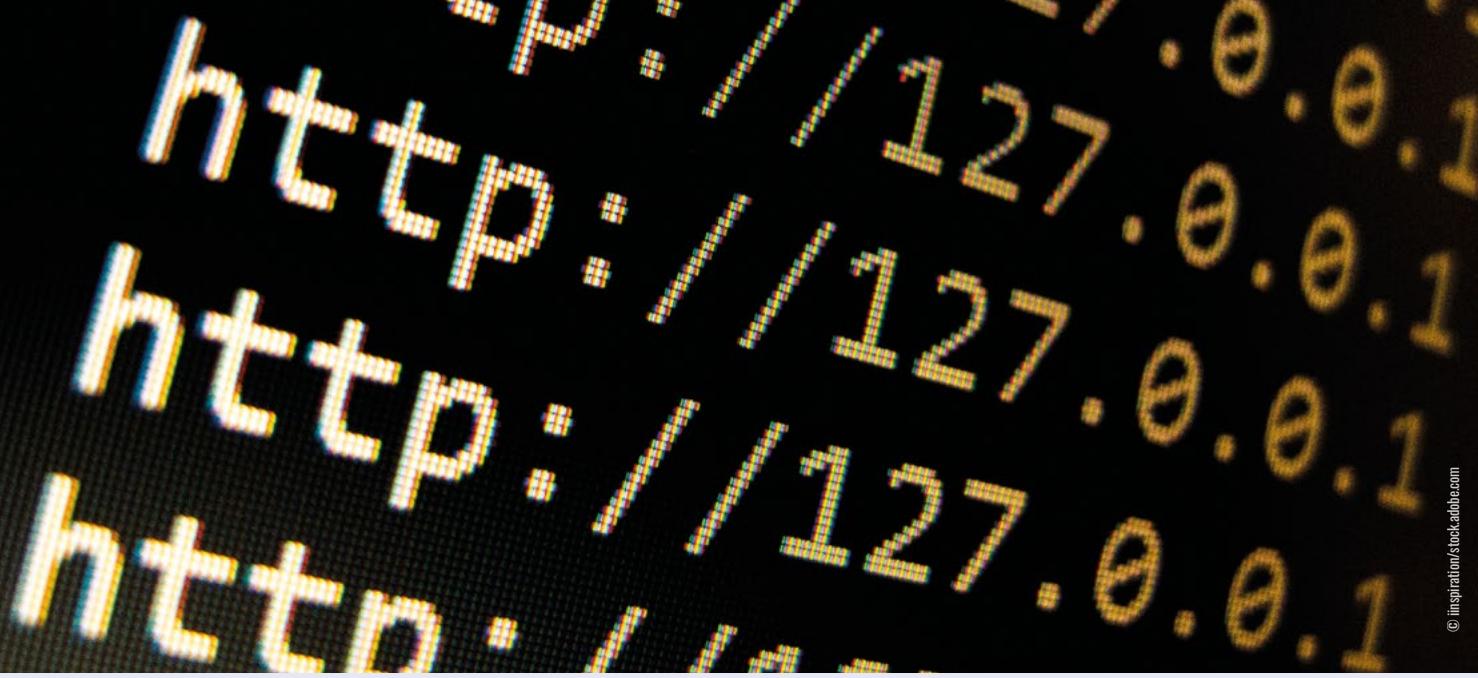
Finally, any 127.x.x.x IP address is a localhost IP address. Localhost IP addresses mean 'this machine', so that a computer can talk to itself. This can be used for testing programs without sending information over the internet; so that applications on a computer can talk to each other; or as a form of secure communication.

Public versus private

So, why do we need both private and public IP addresses? We don't actually have enough public IP addresses for the number of devices used in the world. In the 1980s, the internet was created with 32-bit IP addresses, which was enough

network, you send something through your router. Your router remembers where that traffic was sent, and when an answer comes back, it translates that traffic back to you on your computer. You have to send something for that to work, and nobody can send you anything from outside your network, because your computer has a private IP address. In fact, nobody can see your computer when it has a private IP address. It is possible to make your private IP address visible by changing a setting on your home router. This is called port forwarding, and you can do this to play a game online, or to let you or someone else access your computer remotely.





Protecting your devices

A port is part of an address, and can be used with an IP address to identify the kind of network traffic they receive. For example, secure web traffic (with the padlock symbol in your browser bar) uses port 443 and is the secure and encrypted version of the protocol that makes the World Wide Web work. The console gaming service Xbox Live uses multiple ports, including ports 88 and 3074. It is very common, though potentially unsafe, for someone playing games to forward the ports they use for gaming. Port forwarding is like opening the door between your computer and the internet. So, if you forgot something at home and wanted to connect to your home computer from school, you could port forward your home router to allow access to your home computer.

Isn't it bad to open the port forwarding door? Yes and no. Would you leave your front door open in a safe neighbourhood? What about in a bad neighbourhood? While some may tell you that it is safe to port forward, there are risks. With port forwarding, anyone on the internet can find your computer. In fact, if you port forward, a website called Shodan will list your computer for anyone to see. Would someone then make the effort to break into your computer and do something

bad? Maybe not, but as more and more people find ways to abuse the internet, port forwarding becomes less and less safe.

If you want to access your device remotely and safely, the simplest answer is not to port forward. If you do port forward, then make absolutely sure that you have changed your passwords from their default values, because those are now all that is stopping someone from accessing your computer. Alternatively, look for remote access or private network software that can solve the problem safely.

After installing appropriate software, check that your network is safe. You can perform a scan using a website such as ShieldsUP! (helloworld.cc/shieldsup) to see if your computer is visible from the internet. It will detect your public IP address and check if any ports are open and, thus, if your computer can be reached by someone on the internet. One or more of your ports might be open, for example, if you changed your port forward router settings.

The best result from a scan is that none of your ports are open and you are 'cloaked', or invisible to the internet. If one or more of your ports has to be open for some reason, you should check what is behind that port — it may be a game or another application. Unfortunately, there is no easy way to determine whether you have left

FURTHER RESOURCES

- helloworld.cc/IPaddress: a good explanation of an IP address from a security point of view
- portforward.com: how to port forward your home router, and why this is not necessarily a good thing!
- helloworld.cc/shodan: the title of this article says it all: 'Shodan: the scariest search engine on the internet'
- helloworld.cc/myIP: there are many websites that will show your public IP address, but this one will often track you to very near your home location — cause for you to take extra precautions!
- helloworld.cc/shieldsup: a website that scans your computer to see if you have any ports open

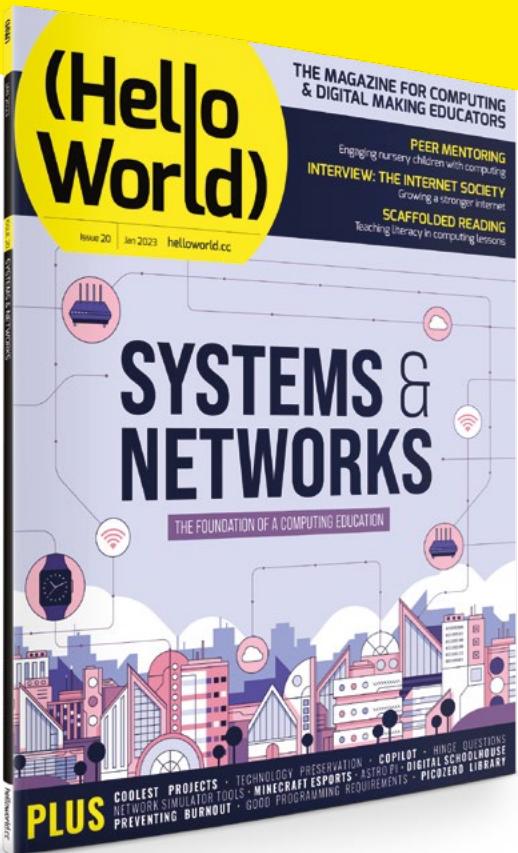
the password at the default setting, or even don't have one, so you should and must do that manually.

If you follow all these steps, you will not be the person on the TV show that can be traced and hacked. In fact, if you are successfully cloaked, there is no way for anyone to find you, and you are networking safely! **(HW)**



(HW)

SUBSCRIBE TODAY



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 four times a year

TO SUBSCRIBE VISIT:
helloworld.cc/subscribe

Not a UK-based educator?

- Buy a print copy from helloworld.cc/buy – we ship to over 50 countries
- Subscribe to receive the free PDF on the day it is released
- Read features and news at helloworld.cc



A BALANCING ACT

Michael Conterio provides tips to support learners with understanding equations

A common problem students face is forgetting equations. Even when they have memorised equations, learners rarely understand what they mean and, therefore, struggle if what they want to figure out does not map exactly to the equation they learnt. In both cases, thinking about what each value in an equation represents can help — especially if you then use that to compare the units of each value. This article will provide some tips, using the example of calculating the colour depth of a bitmap image.

The colour depth of an image is the number of bits used to represent the colour of a single pixel from that image. In other words, it is the number of bits for one pixel, with a unit of 'bits per pixel'. Bits per pixel simply means the number of bits in an image divided by the number of pixels in the image; you just have to work out which numbers of bits and pixels. Make sure they come from the same place — for example, the number of bits for an image and the number of pixels in that image.

Equations with multiple terms

Knowing the equation and units for a quantity such as colour depth can help you use that quantity to calculate others. You can do this by combining different quantities you know the units of. By balancing the units, you can build up equations containing multiple terms and check that they look sensible. For example, imagine that you wanted to get the file size of a bitmap file and you already knew the colour depth. As colour depth is given by the number of bits divided by the number of pixels, you would need to multiply the colour depth by a number of pixels to get a file size in bits:

```
file size [bits] = colour depth
[bits/pixels] * number of
pixels [pixels]
```

The units here balance:

```
bits = (bits/pixels) * pixels
```

Divide both sides by pixels:

```
bits/pixels = bits/pixels
```

This now shows that the equation is balanced, as both sides are clearly equal.

Sometimes you might need to combine more things in a chain to get what you want. For example, imagine that you wanted to upload a bitmap image, and would like to know the time this would take. In this scenario, imagine that you also had the colour depth, but hadn't calculated the size of the file. You'd start with an equation for upload time, in which the units balance. If you couldn't remember this, you could work it out using the units, as you did for colour depth:

```
upload time [seconds] = file
size [bits] / upload speed
[bits per second]
```

The units in this equation look like this:

```
seconds = bits/(bits/seconds)
```

Multiply both sides by (bits/seconds):

```
(bits/seconds) * seconds = bits
```

This is another way of saying:

```
bits = bits
```

The units always have to balance!

As you don't have the size of the file, but you have the colour depth and number of pixels, you can substitute the equation:

```
file size [bits] = colour depth
[bits/pixels] * number of
pixels [pixels]
```

... into the equation for upload time, to get an equation that doesn't use the file size:

```
upload time [seconds] =
(colour depth [bits/pixels]
* number of pixels) / upload
speed [bits per second]
```

You can check that the units match up:

```
seconds = (bits/pixels) *
pixels/(bits/seconds)
```

```
seconds = bits * (seconds/bits)
```

```
seconds = seconds
```

Of course, in all of these cases, you or your students need to start by thinking about what each of the terms means. However, by combining quantities and units like this, you can exercise your knowledge and become happier with the mathematical side of computer science. What techniques do you use to help develop students' understanding of computing terms and the mathematical side of computer science? We'd love to hear from you — tweet us at @HelloWorld_Edu. (HW)



MICHAEL CONTERIO

A former physicist, Michael now works as an online course production manager for the Raspberry Pi Foundation.

CROSS-AGE PEER MENTORING

Matt Holt shares how his primary school has developed a novel approach to teaching computing to nursery-aged children

In 2005, England and Wales introduced mandatory PPA (planning, preparation, and assessment) time for teachers. This means that 10 percent of teachers' time during the week should be away from their class, so that they have time to plan and mark during their working hours. While this was welcomed wholeheartedly by the rank and file, it became yet another logistics challenge for senior leadership. Many schools chose to employ specialist sports coaches or additional class teachers, or even used supply teachers, to fill the gap for learners. My primary school took a different approach and recruited a specialist ICT teacher, which is a fairly common model now, but was considered a novel solution in the early noughties.

As I had worked in the private sector as a software engineer before retraining as a teacher, the role seemed tailor-made for my

skill set, and I became that ICT teacher. The timetable was arranged to allow each year group, from Reception to Year 6 (learners aged four to eleven), two hours of discrete technology teaching, often, but not always, delivered in the ICT suite. However, it wasn't until about my fourth week in the post that I discovered that our school had a 36-place nursery! As a newly qualified teacher, it was only in the second half of my first teaching term that I gained enough confidence to question why the early-years children had been omitted from the ICT timetable. The answer was that there simply weren't enough hours in the week to accommodate the youngest of our children.

Nursery Partners Programme

It had already become clear that children in Reception were struggling with the fundamentals of traditional computing. Even back then, the learners were already well

established as a touchscreen generation and had limited or no experience with a keyboard or mouse.

I decided to give peer mentoring a try. I'd seen it used in other schools for reading, with excellent results, so why not use it for ICT? I was, and still am, fortunate to work in a progressive school. The idea was embraced by the nursery staff, head teacher, and perhaps most importantly, the older pupils, who couldn't wait to embark on the trial project. Today, the Nursery Partners Programme is an embedded feature of the teaching calendar, running once every few weeks.

Year 6 children (aged ten to eleven) are first given some mentoring training. They are coached on the practicalities of working with younger children, and how to motivate and encourage their mentees. This training puts a strong emphasis on relinquishing control to allow the nursery child to have a hands-on learning experience. At the next session, the early-years children and the older children are split into two groups and each younger child is partnered with an older buddy, who they will work with for the rest of the year. The two mixed-age groups then separate, one going to the nursery setting and the other to the technology suite. After around 45 minutes, the two groups swap places.

The goal of these sessions is twofold: to develop the younger child's confidence and technical skills, and to consolidate the older child's understanding of core concepts. This is particularly relevant to lower-ability mentors, who gain a fantastic sense of achievement and pride in being able to pass on their knowledge. Over the weeks, the mentors will be tasked with



The goal of the programme is to develop the younger child's technical skills and consolidate the older child's understanding of core concepts



The first session is always a drawing task in which the buddies draw portraits of each other on the computer

teaching their younger partners a range of practical computing skills, but the first session is always the same: a drawing task in which the buddies draw portraits of each other on the computer. This activity promotes dialogue between the children and gives the teacher a good opportunity to do some formative baseline assessment of the nursery child's ability to use the mouse. As the weeks progress, children work together on a range of technologies including tablets, programmable robots, animation, and filming, as well as some unplugged activities.

In the nursery setting

While children in the technology suite are getting to grips with the computers, the other group is busy in nursery. The nursery teacher provides some guidance, but the session is largely led by the younger children, who are always keen to show their older buddies around their domain! Typical activities include small-world play (imaginative play using toys), construction, painting, sand and water, reading, maths, and (far and away the most popular), outdoor play.

The one-to-one attention the nursery children receive is invaluable. Very often, younger children are more open and

engaged with another child (albeit an older one) than they ever are with adults in the setting. The older children love the responsibility that is placed on them. Many of them also attended our nursery and enjoy the chance to relive their youth!

At the end of the session, Year 6 digital leaders give the nursery children stickers before they return to their setting. I often ask the Year 6 children to evaluate the lesson and think about the next steps, such as other projects they could run with their buddies.

We launched this project to solve a logistics problem, but the Nursery Partners Programme has given us so much more. Nursery children get quality one-to-one learning from enthusiastic, technically skilled and relatable mentors — as one nursery child, Kyle, put it, "I like doing learning partners because the Year 6 children are funny and they help me on the computer." In turn, the older children learn coaching skills, patience, kindness, teamwork, and a huge sense of pride and self-worth. Mia, one of our Year 6 learners, told us: "I enjoy Nursery Partners because I want to be a teacher when I grow up."

As a Specialist Lead Educator in computing, I have had the pleasure of meeting many computing subject leaders



MATT HOLT

Matt is a specialist teacher in computing at Birkenshaw Church of England School in West Yorkshire, UK. Before joining the profession in 2003, he worked as a technical consultant and software engineer. Matt is also a Specialist Lead Educator for primary computing in the Kirklees Education Authority.

over the years, and have even managed to convince some of them to give the Nursery Partners Programme a go! I'd love to hear from anyone who is keen to try it, or is already doing this in their school. Who knows, you might even be inspiring the next generation to follow in your teaching footsteps! Feel free to reach out at matt.holt@kirkleeseducation.uk. (HW)

SETTING UP A MINECRAFT ESPORTS LEAGUE

Siobhán Morgan shares how she started an esports league in her primary school, where pupils competed to demonstrate their knowledge of a topic through Minecraft challenges

Last summer, I was inspired to set up a Minecraft esports league in my primary school after taking part in the online Microsoft esports Teacher Academy (helloworld.cc/esportsacademy) and watching a livestreamed esports final between two schools in the USA and South Africa. Esports is defined as online competitive gaming in which teams compete against each other in front of an audience. I initially set up our league as an extracurricular club, working towards streaming a live final. If that went well, I thought we could then move on to competing with other schools or doing more within our own school. I thought we could maybe even start house esports competitions, as we do with other, more traditional, sports.

Setting up the league and sessions

We created our Minecraft esports league with mixed year groups. I ran one half-

term with pupils aged seven to nine, and another half-term with pupils aged nine to eleven. This allowed pupils to collaborate with children they don't usually work with, and helped them to develop key skills such as negotiating with others and communicating their ideas effectively. During the first session, pupils chose their teams (with teams of around four, but this was flexible) and team names. We developed some rules and guidelines about how to work together. I explained that we would score the builds each week using a range of criteria, including scale, colour, effective use of blocks, and references to the build theme.

There was then a different theme for the build challenge each week. I informed the teams of the theme one week in advance, so that they could conduct any further research if necessary. The weekly themes covered topics that pupils had already learnt about in their curriculum lessons, or were relevant to seasonal events. For example, during Ramadan, pupils researched the festival of Eid and created models reflecting what they had learnt. One of the great things about Minecraft is that pupils can present their knowledge both through building models that represent the theme, and by using tools such as 'books' and 'signs' to display written information about the topic they have learnt about.

I downloaded the esports world 'Make & Model: Practice Plaza x12' from Minecraft Education (helloworld.cc/practiceplaza)

to run each session. Using an esports world, rather than asking each team to build in a standard Minecraft world, has a few advantages. Firstly, a timer is clearly displayed. The teacher sets the timer at the beginning of the challenge, which begins when you press the start button. Secondly, there are clear areas to build for each team, and they are all the same size, making the challenge much fairer. Finally, these worlds are prebuilt with an export button, so you can export finished designs as CAD models, or even 3D-print them.

At the end of each session, I assigned points to teams based on the agreed marking criteria, and we tracked and displayed each team's progress in a league. We also had time for reflections at the end of each session, which was an important part of improving teamwork skills.

Developing teamwork

The advantages of esports include helping to develop a range of twenty-first-century skills. While this is always a clear bonus, I also wanted to make explicit links to the curriculum, so that pupils could demonstrate their understanding of a topic at the same time as developing important skills such as effective collaboration and communication.

Initially, some members of the teams could be a bit domineering, and were not successfully communicating their ideas with their teammates or taking suggestions from others on board. A few sessions later, it was clear that team



Part of a nature garden built in a 'planet protectors' week

SET UP YOUR LEAGUE!

To try this out for yourself, you will need to:

1. Plan a series of challenges, possibly linked to your curriculum
2. Get pupils to select team members and choose a team name
3. Inform teams of the challenges in advance so they can do additional research if they choose
4. Score the builds using agreed success criteria after each challenge
5. Livestream the challenges to those not involved, to build up interest in your league

members had been developing these skills and were collaborating far more successfully. This became apparent in their builds, as their designs had clearly been produced collaboratively as a team, rather than as four individual pupils working independently in the same virtual space. At the end of each session, pupils shared their reflections, with many reflecting on the improvements they had made in their collaboration skills.

Streaming the final

After weeks of competing, we had two teams in the final, and they were eager to compete! The theme of the final focused on representing a historical event. One team selected the Great Fire of London, while the other selected World War II. Both teams went away to research and plan their builds before meeting at the end of term for the grand final of the inaugural season.

By the time we held the final in July 2022, all our staff had become extremely proficient at using Google Meet, and were more than happy to stream the event to their classes. This was great, as it opened up the event to the entire school community. We had pupils watching and cheering along in classrooms, while our younger pupils were excitedly watching together in one physical space.

I called upon two of my digital leaders (older pupils who are enthusiastic about technology) who had been keen to share their love and knowledge of Minecraft to



One team created a bee and a hive as part of a nature garden during the 'planet protectors' week

“ SOMETHING THAT STARTED WITH 50 PUPILS ENDED WITH A WHOLE SCHOOL OF 200 PUPILS BEING EXCITED BY ESPORTS ”

become 'shoutcasters' (commentators) for the final. This was particularly effective as they had a good understanding of building in Minecraft, as well as an understanding of the topics that the two teams were drawing upon in their builds. As well as a running commentary of the final, they interviewed teachers and members of the competing teams, which added an extra layer of excitement for everyone involved. By the end of the challenge, something that had started with 50 pupils had drawn a school of 200 pupils into the excitement of esports, and we hope it will inspire others to participate in our league next year. With the esports pilot having taken place at the recent Commonwealth Games, it's an exciting time in the world of competitive gaming, and with the many benefits it can bring to your pupils, it may be time for you to get involved too! (HW)



SIOBHÁN MORGAN

Siobhán is head of computer science and design technology at Exeter Junior School in the UK, and an associate lecturer at the University of Exeter. She is currently completing a PhD in technology-enhanced learning at Lancaster University ([@koduclassroom](#)).

COMMUNITY OUTREACH WITH DIGITAL SCHOOLHOUSE

Chris Lovell discusses his experiences of the Digital Schoolhouse programme, which engages primary-school pupils with computing through its community of Lead Schools

My school has recently been accepted as a Lead School member of the Digital Schoolhouse programme (digitalschoolhouse.org.uk). Digital Schoolhouse is a not-for-profit programme which uses play-based learning to engage primary-school pupils and teachers in the UK with computing. It offers free workshops, activities, and events, led by its community of Lead Schools. The resources are rooted in the English national curriculum for computing, and aim to equip pupils to use computational thinking and creativity to understand and change the world.

Joining the Digital Schoolhouse programme, with its focus on continuing professional development, has enabled me to develop my own teaching style and connect with computing teachers nationwide. The programme has also helped me bridge the gap between industry and education, with its strong focus on educating pupils about the careers available in the UK games industry.

Play-based learning

Creative workshops are at the heart of the programme. They are delivered by the Lead School together with the class

teacher from the participating primary school. In my school, lower-secondary pupils also support the workshop delivery, giving them an opportunity to discover and learn together with young people from various communities in our local area.

In my conversations with primary teachers, some have mentioned that they find programming challenging to teach. It's not a skill that these teachers necessarily experienced as part of their own education or training, and they have found that becoming a proficient coder takes time and persistence. I also remember my own fears when first training and the wide variety of IT problems that can occur in a lesson, no matter how well prepared we feel we are.

I think this is probably why some of the most popular workshops are Crazy Mazes, Get with the Algo-rhythm, and Just Dance with the Algorithm, which all offer creative and fun ways of teaching programming and coding. The Just Dance with the Algorithm one-day workshop is a particular favourite with pupils. Learners begin by creating and planning their own dance moves using pen and paper. They then refine and test out their moves by dancing together in groups. In the second half of the workshop, pupils take their learning into the Scratch programming environment, where they create sprites that dance in the sequences they have

TIPS FOR APPLYING TO BECOME A LEAD SCHOOL

When applying for the Digital Schoolhouse programme, ensure you can:

- Describe how becoming a Digital Schoolhouse will support your school in developing innovative computing provision
- Describe the existing resources your school has access to that will help you deliver the programme
- Describe your existing access to your local community of primary schools
- Show awareness of your strengths for establishing a network of schools
- Describe the challenges you anticipate when establishing your school as a Digital Schoolhouse



© Image courtesy of UKIE

ABOUT DIGITAL SCHOOLHOUSE

Digital Schoolhouse is delivered by the UK games industry trade body (UKIE) and backed by the video games industry and the UK government. Sponsors of the programme include Nintendo, PlayStation, EA, and SEGA. Through its 79 Lead Schools nationwide, the Digital Schoolhouse programme has reached over 130,000 pupils and over 11,000 teachers since its creation in 2014.

created. At the end of a recent workshop, one parent told me that their child had enjoyed the workshop so much that they were going to continue working on their algorithm at home.

Digital Schoolhouse's research has shown that pupils who participate in their workshops become much more confident with coding, and grow their understanding of computer networks and of how to be safe online (helloworld.cc/schoolhouseresearch). This research has also shown that the gap between boys' and girls' confidence in each of these areas is narrowed following a workshop. In my experience, pupils find the workshops

enjoyable, and excited pupils consistently ask me to return for a follow-up. I have observed that pupils' confidence with coding grows during a workshop, and pupils are able to explain their algorithms to me and the rest of the class by the end of the day.

Being a Lead School

As a Lead School in the programme, we receive regular training from leading computing education practitioners through Ingenuity Days. In our induction workshop with the team earlier this year, we were privileged to attend a training session by researcher Paul Curzon on semantic

variables, and making these concrete by associating them with everyday activities that learners are familiar with. At a future Ingenuity Day, we will hear how we can better prepare aspiring game makers for a career in the games industry, and learn to use some of the latest game software, such as the Construct 3 web-based game development engine (construct.net).

I believe that being a member of the programme has given me the tools to be more creative in my own teaching. I also feel a sense of reward for supporting local primary-school teachers in delivering the computing curriculum effectively. In time, I know that the programme will

PUPILS WHO PARTICIPATE IN DIGITAL SCHOOLHOUSE WORKSHOPS BECOME MORE CONFIDENT AT CODING

wave theory and its application to the workshops and to computing teaching in general. In the training session, Paul led an activity in which we applied semantic wave theory to an imagined workshop session. We discovered that the approach could help us improve explanations by unpacking and repacking abstract concepts such as algorithms or

enable me to become a better teacher, by observing the variety of teaching and learning approaches adopted in our local community, and by being connected to some of the leading computing teachers across the UK. I encourage computing teachers, coordinators, and leaders at primary and secondary schools to find out more about the programme today!

You can find out more about how to join the programme at digitalschoolhouse.org.uk/apply or by emailing the team directly at dsh@ukie.org.uk. (HW)



Digital Schoolhouse's workshops reduce the gap in confidence between girls and boys



CHRIS LOVELL

Chris is the head of computing at Ashfold School in the UK, where he is also a lead teacher in the Digital Schoolhouse programme (@ashfoldcomp).



■ Children working on their designs

MAKECODE: GAME ON!

Peter Rutherford shares a STEAM project he carried out with primary pupils, designing and building arcade machines and developing retro-style games in MakeCode

I have been teaching Year 6 (pupils aged ten to eleven) for several years, and I've learnt that after the pupils take their SATs (formal exams in England) in May, it is useful to have an extended cross-curricular project to keep their attention up until the summer holiday!

Previously, we sent a group of Lego astronauts to an altitude of 35,000m using a weather balloon filled with helium. This project was a huge undertaking, and there was a lot of learning for both myself and the pupils. We had to apply for permission for the launch from the civil aviation authority, and then had to contact air traffic control on the morning of the launch to get clearance.

Our launch slot was set for 5.30am, and children and parents (some still in pyjamas) came to watch. As the balloon rose through the atmosphere, it expanded until it finally burst. Our Lego astronauts then returned to Earth with a parachute, and thanks to a GPS tracker and the power of Raspberry Pi, we could locate it and recover the on-board GoPro camera. I had never seen a group of young students so engaged in their schoolwork. To replicate this success, at the end of the last academic year, we designed and built retro arcade machines and games to play on them.

The idea for the arcade project came about as the cohort of pupils I had last year

had a strong interest in retro video games and Japanese culture. We had an old Nintendo 64 set up in the classroom and the children were slowly making progress through The Legend of Zelda when they had a few minutes of free time here and there. Through my own research into the MakeCode platform, I discovered that some people had built functional arcade machines using Raspberry Pi and some fairly cheap components. I was inspired, and got to work planning the project.

Building excitement

To launch the project, I arranged a class visit to Four Quarters East, a retro arcade

bar in Hackney Wick, London, which is packed with vintage arcade machines and games consoles. The manager kindly agreed to open up early for us and the children spent the afternoon playing on old Pac-Man and Space Invaders machines, competing for high scores and reviewing the games they had played. They also had the opportunity to look at how the cabinets casing the machines had been designed. We returned to school full of enthusiasm for the project and bursting with plans for our own games and machines.

We later spent one afternoon looking at the work of the French street artist Invader. He creates mosaics inspired by the Space Invaders game and hides them across different cities around the world. The children loved his work, and some had even seen some of his mosaics close to our school in London. After this introduction, we created our own paper mosaics in the same style and 'invaded' our school site with them. This added to the excitement around the project, as suddenly there were computer game characters popping up all over the school.

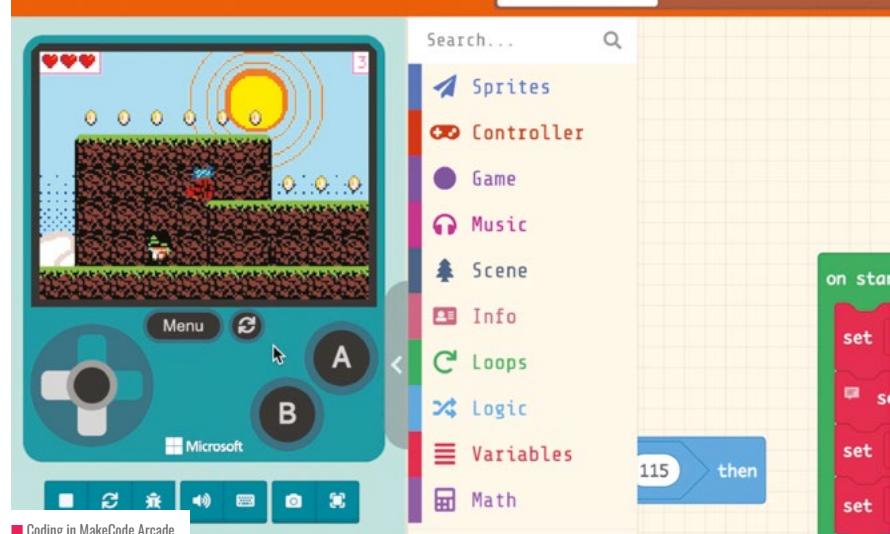
Video game history

After whetting learners' appetites, the next step was to research the history of video games. We went back to the dawn of the industry and looked at pioneers of gaming such as Tomohiro Nishikado, who developed Space Invaders. It was fascinating to see his original sketchbooks. Here, he planned his code and designed his characters by mapping out the pixels based on his initial drawings — something we would revisit later when designing our own game characters.

A parent of one student also put me in touch with her brother, who designed characters and scenes for several computer games in the eighties and nineties. He kindly agreed to hold a video call with the



■ Building the cabinets



class to talk about his work and answer their questions. One game he worked on was the original Jurassic Park game for Amiga, and he showed us his design process for some of the dinosaurs in the game. The children were fascinated by his work and career, and could then play the game and see his work in action using Raspberry Pis running RetroPie.

would like to make based on their interests and the games they had enjoyed playing at the arcade. The hardest aspect of this was trying to contain their excitement! I had to explain that their games would not be rivalling the latest PlayStation or Xbox releases, but that they could make something which was fun and that people wanted to keep playing. Once they knew

“THE CHILDREN WERE SO ENTHUSIASTIC – THEY SPENT HUGE AMOUNTS OF THEIR FREE TIME WORKING ON THEIR GAMES”

Our research into video game history also led us to the work of software engineer Jerry Lawson, who developed the technology to store a game on a removable cartridge. Prior to this, gamers needed to buy a new console every time they wanted to play a new game. Lawson's work thus paved the way for modern-day consoles. Jerry Lawson unfortunately passed away in 2011, but the children wrote letters to his family to show their appreciation for his work, and we received a friendly response from his son.

Research and planning

At this point, I split the children into small groups and they created their own fictional games studios. The groups all came up with a name and logo for their studio and started discussing the types of games they

the type of game they wanted to make, they began to map out how it would work and what it would look like by sketching their backdrops and characters. I spent some time working with each group to help them figure out how their ideas could translate into code based on the limitations of the platform.

Coding in MakeCode Arcade

At this stage, as we began coding our games, the project started to become very real. We used the MakeCode platform (makecode.microbit.org), which was new to my students. Over their years in primary school, though, they had developed their coding skills using Scratch, and so the block-based system wasn't too unfamiliar. To introduce them to MakeCode, we used BBC micro:bits. They learnt how





■ Some of the design competition entries

► to program the micro:bits' push buttons to display different images, and how to take readings from the on-board sensors and display them. Some students even programmed their micro:bits to send and receive messages via Bluetooth.

We then moved over to MakeCode Arcade (arcade.makecode.com). The children worked through some of the Skillmaps before starting their own projects. Some of our less experienced coders began by adapting the example games, while the more experienced children started games from scratch. They created their assets based on their original sketches and then started to code their games.

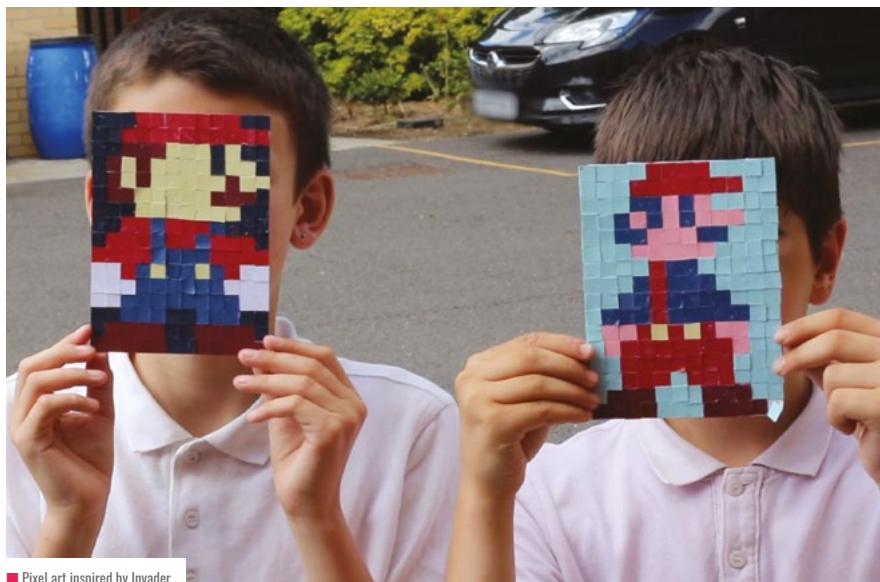
To continue bringing the project to life, we next looked at how a good soundtrack can enhance a video game and listened to some famous examples. As a class, we learned how to play the Super Mario Bros. theme tune using Boomwhackers (colour-coded plastic tubes tuned to a musical pitch by length). The children then composed their own soundtracks using glockenspiels before recreating them in MakeCode. Some children who had created level-based games figured out that they could increase the tempo of their soundtracks each time the player progressed to the next level, to add tension and enhance the player's enjoyment of the game.

Designing and building the machine

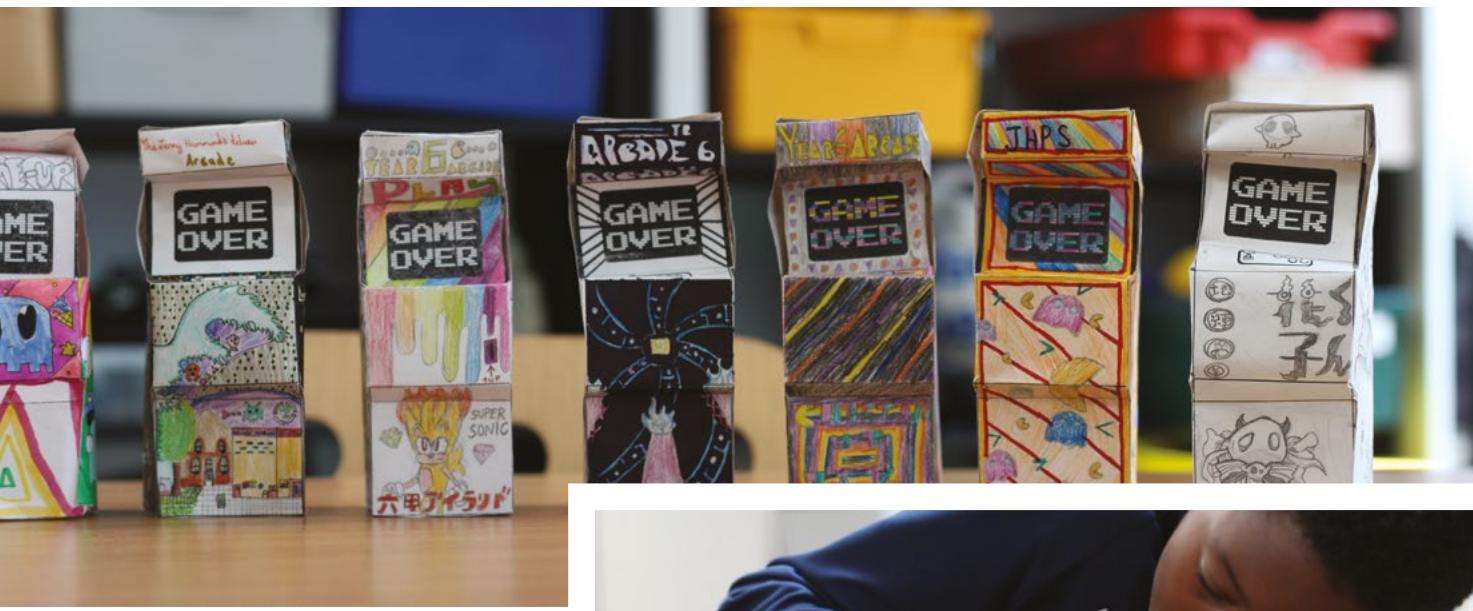
We ran a whole-school competition to design the cabinet that would hold the machine the children would run their games on. We gave the children the option to either submit a 2D design or create a 3D model of their design from a net. The submissions were incredible, and the process of deciding the winner was so tough that I decided to combine some of the designs and make two machines. With the winners announced, I got to work ordering the materials and components ready for construction.

To work out how big the cabinets needed to be, the children measured the height, eye level, and elbow height of children from across our school and calculated the averages. We then used these measurements to decide the height and positions of the screens and control panels. We built the cabinets from MDF. I cut out all the panels using a circular saw and a jigsaw, and then the students helped to screw them together. We mounted the screens and drilled holes for the joysticks and push buttons.

We invited the children who had won our design competition to come and paint their designs onto the cabinets. We gave them a couple of coats of emulsion paint in their chosen background colours and then used POSCA pens to add details. The finished products were very different, but both were equally striking.



■ Pixel art inspired by Invader



Raspberry Pi wiring

The children then faced the challenge of wiring up Raspberry Pis to the joystick and button in the control panel, and the monitor. This proved to be quite tricky. For this project, we used Raspberry Pi Zeros, as their functionality is more than adequate and this allowed us to keep costs down. We installed the Arcade.Cardboard disk image from the MakeCode Arcade website onto our SD cards (helloworld.cc/cardboard). We then used the pre-installed RPI Configurator program to ensure that we wired up our buttons to correspond to the correct functions. Once the wiring was complete, we were able to test our machines using some of the example games from MakeCode. It was great to see the pupils' reactions, as we now had two functional arcade machines!

Time to play

The last step was for the children to test and debug their games before downloading their code and transferring it onto the SD cards so that they could play the games on the arcade machines. All the games were fantastic, and most importantly, very playable. We had recreations of classic arcade games, platform games with multiple levels, games based on football and basketball, and even games set in our own school featuring staff members as enemies. On the first day, we put both of the machines in the school playground and



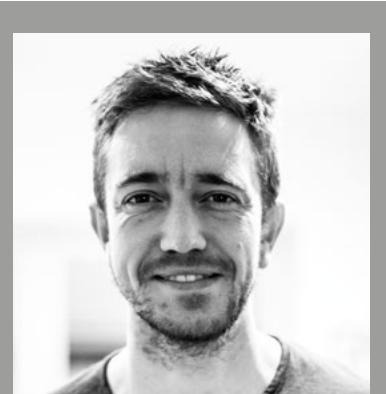
■ Wiring up Raspberry Pi

had huge queues of children wanting to play during their break times. The machines now move around the school on rotation, so that all of the children get a chance to play on them and become inspired to create their own games.

The project turned out better than I could possibly have imagined. The children were so enthusiastic about it that they spent huge amounts of their free time working on their games. Most importantly, the amount of learning that took place throughout the project was incredibly satisfying. The project covered age-appropriate national curriculum objectives from computing, science, maths, English, music, art, and design technology. Plus, the children could see how the skills they had learnt in class were useful in a real-life situation.

They also developed their collaboration skills: I even had children arranging video calls with each other so that they could continue to work on their games together at the weekend. What's more, they developed resilience when things

went wrong, and learnt to work methodically to solve any issues that arose. The MakeCode platform made this project possible, and I would be more than happy to help other teachers carry out a similar project with their pupils! [\(HW\)](#)



PETER RUTHERFORD

Peter is the computing lead and a Year 6 teacher at the Jenny Hammond Primary School in East London, UK.

#CSK8 Podcast

with
Jared O'Leary

PREVENTING BURNOUT

Jared O'Leary summarises how guests on the #CSK8 Podcast recommend preventing burnout in computer science education

Educators in the USA and elsewhere are struggling right now. With relatively low pay for the skills needed to teach computer science, an ever-increasing list of responsibilities and risks placed on educators, and a work week that stretches beyond 40 hours, it's no wonder that so many computing educators are burned out.

In my interviews with more than 50 computer science educators and scholars on the #CSK8 Podcast, I frequently ask guests how they attempt to prevent burnout. This article uses Asian Efficiency's TEA framework (helloworld.cc/TEA) to discuss how guests are intentional with their time, energy, and attention.

Time

One way to reflect on how you're spending your time is to spend a week setting an hour-long timer at the start of each day and pausing at the end of each hour to write down what you were doing and how you felt in the previous hour. This activity reveals not only how you're spending your time throughout your week, but how you feel when engaging in different activities.

Once you've established a baseline, you can refine your schedule by planning for both work and leisure. Many guests report using the Pomodoro Technique, in which you work for a set period before taking a scheduled break to do

something energising. For example, I often work for 50 minutes and then take a 10-minute break to walk, read, meditate, or play the drums. Regular breaks allow me to metaphorically press F5, refreshing my mind and body throughout the day.

Another approach is to set clear boundaries between work and leisure by signalling when you're switching from one to the other. For example, do 20 minutes of exercise at the end of your workday to signal to yourself that you don't need to think about work for the rest of the evening. If you notice you're thinking about work during your leisure time, add the words 'no' or 'not right now' to your

vocabulary. As education consultant and podcast guest Justin Cannady says, "By saying 'yes' to everything, you're really diluting everything." It's OK to block off your personal time and say no to requests for work meetings during that time. When work invades your dedicated leisure time, plan to make up for your lost leisure; and take holidays as actual holidays, instead of trying to catch up on projects. Regularly taking time to step back and metaphorically sharpen your mental axe can not only help with getting more done than if you hadn't taken time to rest, but it can also help you to feel refreshed and energised when teaching.

Energy

Many podcast guests mention being intentional in restoring their energy, for example, by walking, playing squash, swimming, cycling, doing martial arts, jogging, powerlifting, or performing aerial yoga! There are so many ways to have fun while staying active, so find something that works for you and schedule it into your day. Educators also mention restoring their energy through the foods they eat. Because everyone has different nutritional needs and goals, check in with a healthcare professional to learn more about improving your energy levels through nutrition.

Some guests say they find it easier to stay focused at certain times of the day than others. I'm most energised and focused in the mornings, so I get up early and focus on my most important task at the start of the day. I save a running list of tasks that are suitable for when I have lower energy later in the day but want to stay productive (such as checking emails).

Another approach that some guests use to refresh their energy levels is socialising more. For example, educators can join their local Computer Science Teachers Association (CSTA) chapter to connect with other computer science teachers, which can really rekindle your passion for teaching. Another approach is to socialise with groups of people who share an interest in something you're passionate about in your free time. Mitch Resnick, Lego Papert Professor of Learning Research at the Massachusetts Institute of Technology,

restores his energy by focusing on the effect of his work with Scratch: "There's a lot of satisfaction I get from when I see the impact the work is having with young people around the world."

Attention

Podcast guests frequently mention that if you put too much pressure on the little details of the job, it can take away your time, energy, and attention from helping kids. For example, Khalia Braswell, computer scientist, educator, and technologist, noticed that when she checked her email at the start of the day "it would kind of ruin my morning if I'm like trying to respond to something I wasn't anticipating". Finding ways to disconnect from things that distract you by turning off notifications, checking email no more than twice a day on weekdays and not

attention; for example, close the office door behind you when you're done working for the day. If space is limited, you can modify your environment to differentiate between work and leisure modes. For example, I turn my computer monitor in one direction when working from home and in another direction when playing video games. It sounds silly, but this simple trick works surprisingly well.

We can't help others effectively if we don't take care of ourselves first; this is true both of teaching and life outside of teaching. By focusing on being intentional with your time, energy, and attention, you might be able to prevent burnout. However, it's important to realise that if there were a one-size-fits-all answer, we'd all have six-pack abs and be millionaires. Find a set of tools that work for you and model for your colleagues and students how you take

CONCENTRATING ON THE LITTLE DETAILS CAN TAKE AWAY YOUR TIME, ENERGY, AND ATTENTION FROM HELPING KIDS

at all on weekends (the world won't end, I promise), and disengaging from tools designed to consume your attention (like mindlessly swiping on social media) can all help with maintaining your attention by limiting distractions.

Several guests touch upon the positive impact that meditation has on them. I, too, have been meditating for several years, but I'd like to reframe it as 'attention training' for those who are sceptical. Rather than thinking of it as attempting to spend 10–20 minutes in a pure state of thoughtless nirvana, I'd recommend spending that time focusing your attention on one sensation (such as your breath) or rotating through a variety of senses (switching your focus every two minutes between what you see, hear, and feel, for example). This kind of daily attention training is an excellent way to practise being present and focused.

Guests who work from home recommend physically separating your workspace from your leisure space so you can be more intentional about what grabs your

care of yourself by striving for consistent, incremental improvement over an extended period of time.

If you'd like to listen to the full interviews, as well as other episodes on a variety of topics related to computer science education, search for the #CSK8 Podcast on your podcast app or visit JaredOLeary.com. (HW)



JARED O'LEARY

Jared creates free content for drummers, gamers, and computer science educators at JaredOLeary.com.

IMAGES, QUESTIONS, AND STUDENT UNDERSTANDING

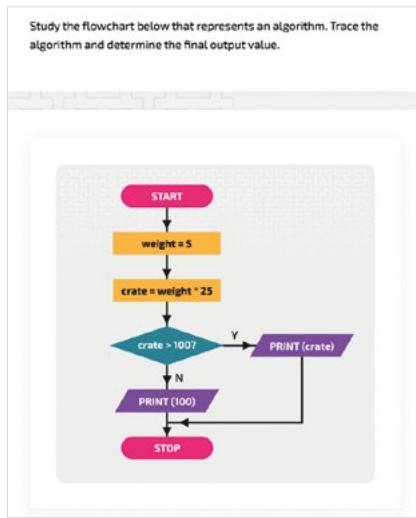
Laura Holborow considers how we can use images within cloze questions to deepen students' understanding of computer science theory

In a traditional cloze question, students complete a passage of text by filling in words that have been deliberately removed. As Ben Garside explored in issue 19 of Hello World, the Raspberry Pi Foundation has produced a wide variety of cloze questions that go beyond this typical approach (helloworld.cc/19). More recently, the content team at the Foundation has been further developing cloze questions by incorporating images. In this article, I will explore the benefits of using images within questions and look at the implications that need to be considered.

Why use images within questions?

Images are powerful methods of communication that can be used to:

- Improve a learner's overall comprehension of a topic



■ Figure 1

- Enable a learner to recall information effectively
- Boost engagement and enjoyment in learning

By using the right images, content can be made more relevant to the learner, and difficult concepts can be communicated more effectively. For example, think about asking learners to trace the output of an algorithm. We could provide some code, and that would enable them to answer the question. However, some learners may struggle to interpret the code and would not be able to answer the question. To address this, we could provide an image of a flowchart that represents the algorithm, which may make the question easier to answer (Figure 1).

In research into the effects of images in computer-based assessment, it was discovered that learners who were given questions that included images were more successful than those who were given questions without images. The research concluded that "to a great extent, the use of images in the questionnaires helps students to select the correct answer" (helloworld.cc/juan2015). Another team of researchers who were studying pictures in test items also acknowledged that "pictures in the stem and in the answer options increased the correctness with which students responded to the test items" (helloworld.cc/sass2012).

When is it appropriate?

Not all questions lend themselves to the use of images, and it is important that the image is used to support the learner in answering the question, rather than being

■ Figure 2

a decorative illustration. Take the topic of common searching and sorting algorithms, for example. When I teach these in the classroom, I use sets of playing cards that I give out to learners and model how the algorithm would work by swapping and organising the cards. Learners then work through questions or activities using their cards to consolidate their understanding. We could replicate this type of learning activity on a computer-based assessment using cloze questions with images (see Figure 2).

Another situation in which images can be used to support learning is when the image supports learners' understanding or recollection of a topic. For example, in the question in Figure 3, we are testing knowledge of user interfaces. By providing the learner with thumbnail images of the interfaces, we hope they will be able to recognise the features and drag and drop the correct name label into the table.

To allow the user to interact with the computer, an operating system provides a user interface (UI). There are various forms that a UI can take.

Match the UI to the image that represents it by dragging and dropping the image into the correct position in the table.

Command line interface (CLI)	
Natural language interface	
Graphical user interface (GUI)	

Items:

■ Figure 3

By using images within cloze questions, we can also replicate how learners would typically be expected to answer a question within a formal exam. For instance, in some assessments, students may be asked to complete a table in a specific way by adding a tick or cross to indicate their answer, or in the case of a truth table, a value. For example, the question in Figure 4 about

which accessibility is the most important. Within the Raspberry Pi Foundation, we are committed to ensuring that our resources are accessible for all learners, and images can create some accessibility issues. It is therefore important to ask yourself why you're including an image in the question. What's its purpose? If you can't think of a good reason, do you need to include it at all?

If a learner has a visual impairment, using an image in a question may mean that the question becomes impossible for them to answer. To check that a question with an image is accessible, ask yourself whether learners can answer it without seeing the image, and what additional support you should provide if the image is inaccessible. For example, you could ensure that the image contains detailed and helpful alt text, which can be inserted as an attribute in an HTML document to tell viewers the contents of an image, and can be read out using a screen reader. Avoid including any information in the image that is not contained either in the alt text or the question itself. You will also need to consider the colours and contrast used within the image; for example, if you have a circuit diagram that contains only red and green wires, that's going to be hard for someone with deuteranomaly (colour blindness) to make sense of.

On the flip side, providing images within questions may improve accessibility for some learners. By representing information in an image or diagram, you may reduce the cognitive load required to answer the



LAURA HOLBOROW

Laura is a learning manager in the secondary education team at the Raspberry Pi Foundation. She previously taught computer science and ICT for 13 years to learners aged 11–16 in the UK. She is particularly passionate about encouraging as many children as possible to study computer science.

that a learner will answer the question correctly. Images can also improve learners' engagement, and when developing learning materials, it is important to provide a range of resource types to keep learners motivated. At the Raspberry Pi Foundation, we are still in the early stages of using and developing images in cloze questions, so it will be really interesting to see how successful learners are at answering these questions and how this compares to their performance with text-based cloze questions. (HW)

The table below identifies actions of an operating system. Each action is managed by a different function of an operating system: memory, processor, or file.

Drag and drop the tick icon to identify what actions are managed by each function and drag and drop the cross icon to identify what actions are not managed by each function.

Action	Memory management	Processor management	File management
Delete a file			
Load data from secondary storage into the main memory			
Determine which portion of memory should be available for a process			

Items:



■ Figure 4

operating systems requires the learner to place a cross or a tick in the correct position in the table, to identify which actions are managed by functions within the operating system. Images in cloze questions are therefore important in effectively preparing learners for the variety of different question types they will come across in assessments.

Considerations

There are some important considerations when using images within questions, of

question, or reduce the amount of text needed to ask the question and hence improve readability. Outside of accessibility considerations, it's also important to think about how learners are interacting with the content and what digital devices they are using to access it. Does the content work on all screen sizes? Are the images still clear when they are smaller?

Images are an important part of learning, and research shows that using images within questions can improve the chances



THE ASYNCHRONOUS COMPUTING CLASSROOM

Kuri DiFede shares why she implemented asynchronous learning with her secondary learners, and offers tips you can apply to your own classroom

We offer digital asynchronous classes at our school. These allow students to take ownership of their learning and take classes that are otherwise prevented by traditional scheduling. I will admit that at first, I was extremely hesitant to teach asynchronously. As a proud computer science teacher, I work hard to perfect my instruction and interaction with students in class. I build relationships with each student, and tailor assessments to their needs. I thought it would be impossible to replicate that experience digitally. But what swayed me towards offering

asynchronous learning was the learners. I thought of those who could not fit computer science into their schedule, and those who would benefit from having more time to complete course material than traditional classes offered. For these students, asynchronous learning offered clear benefits.

Our initial implementation

We started with nine students aged 13. They had all studied computer science throughout their elementary education, but had no room in their school day to continue their computing studies. The

students signed up for the course and their parents all gave permission. The students had also shown in other classes that they could be responsible when doing work outside the classroom.

We worked with an organisation called OYOclass (oyoclass.com) to help us supplement learning. OYO has a platform that's part learning management system and part integrated development environment (IDE). This means that learners can write code and turn it in on the same platform. OYO also sets coding challenges that are checked by the platform's mentor community. This was

extremely helpful when getting started, because we could save time on both the creation of the resources and the grading.

OYO worked with us to create an entire pathway for their *Intro to Python* course, adapting some of the curriculum and instruction to meet our needs. In addition to the course materials and challenges, learners had access to mentors so that they could ask questions. The challenges are set up to allow unlimited submissions. This lets students ‘fail forward’ and continue to iterate until they understand the material. Learners could access the entire pathway straight away, and they were instructed to complete the course at their own pace, with a few check-ins throughout the semester.

Making changes

Even the most studious learners can struggle when given complete autonomy over their time management. We found

TIPS FOR ASYNCHRONOUS TEACHING

- **Meet semi-regularly:** focus on the course, build relationships, and ask questions
- **Assign suggested deadlines:** keep learners accountable and on track
- **Start small, and don't stop iterating:** look for feedback on how to improve the instruction and assessment each year
- **Use an online or easy-to-use IDE:** make set-up as easy as possible, particularly as learners may be on different devices at home

optional; I would pull learners out of one class a week, rotating the periods, and there was about 90 percent participation each week.

During this time, we restored focus on the material, to stop learners from ignoring the class for weeks at a time. We built relationships and worked together. Students collaborated and could ask me or their peers questions. As a teacher, I could

In the beginning, my mindset about asynchronous courses was that learners would miss the support a teacher can provide. In my implementation, I have found that the key to a successful asynchronous program is to have an active facilitator who spends time creating relationships with learners, grading assignments, coordinating meetings, and periodically reviewing the curriculum. My administration counted my asynchronous courses like a synchronous course in determining my workload, which has really helped to make the programme more successful. Overall, I couldn't be more pleased with the result of adding asynchronous courses to our offerings, and I'm happy to have been proven wrong! (HW)

EVEN THE MOST STUDIOUS LEARNERS CAN STRUGGLE WHEN GIVEN COMPLETE AUTONOMY OVER TIME MANAGEMENT

students were prioritising work from their other classes, and this led to a lot of computer science work being left until the end of the year. When I probed deeper, I found that students would get stuck on a problem, and instead of asking for help, would procrastinate for another day. At the end of the semester, many students were then stressed, and I could see their love for computer science draining away. This is something that I wanted to change.

In our second year, we offered *Enterprise Python* asynchronously, and eight of the nine original students returned. We offered *Intro to Python* to another 22 students aged 13–17. We continued to partner with OYO, but we made a big change by meeting with students weekly, rather than just a few times a semester. This can be difficult, as the whole point of asynchronous learning is that you are **not** meeting regularly. These meetings were

see more effectively what was and wasn't working with instruction and assessment, and make changes that positively affected students.

Looking to the future

Now, in our third year of asynchronous teaching, we have added a *Web Development* course and advertised the courses to more students. In addition, 14 previously asynchronous learners are currently enrolled in synchronous courses, and we have 54 students enrolled in asynchronous courses this school year. We have revised the asynchronous courses to add more explanatory material, as well as to streamline some of the assessments that best show mastery of the material. We have also added suggested due dates for students, to support them with staying on track, and learners will continue to check in with me weekly.



KURI DIFEDE

Kuri is a computer science teacher leader at Mineola High School on Long Island, USA. She has been a K-12 educator for eleven years. Kuri is passionate about supporting underrepresented groups in CS, and about providing a variety of CS experiences in K-12 classrooms.

<p>Planning My Day! Below you will answer questions to plan your work for today's class!</p> <p>What is my goal today? (Ex. All 1.1.3 activities, To finish all the practices for section 1.3, To be able to do the practices from 1.2.4 without having to look back on the notes,etc) <u>finish the first three tasks un 1.2.2</u></p> <p>How will I know when I have reached my goal? * <u>when I have 3 points in the exercise</u></p> <p>What can I do to achieve the goal? Will I need any supports? * <u>try to focus as much as I can</u></p> <p>Monitoring My Work: Take a break to think about your progress!</p> <p>Am I sure I know what I am doing? *</p> <ul style="list-style-type: none"> <input type="radio"/> Not sure <input type="radio"/> Kind of sure <input type="radio"/> Idk, but what I am doing is working <input checked="" type="radio"/> Definitely sure <p>Am I making good progress toward my goal? *</p> <ul style="list-style-type: none"> <input type="radio"/> Not really <input type="radio"/> Kind of, I could do better <input checked="" type="radio"/> Oh yeah! 	<p>How am I feeling? *</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Interested <input checked="" type="checkbox"/> Focused <input type="checkbox"/> Motivated <input type="checkbox"/> Off-task <input type="checkbox"/> Need a break <input type="checkbox"/> Stressed Out <input type="checkbox"/> Proud <input checked="" type="checkbox"/> Let me go back to work! <input type="checkbox"/> Sad <input type="checkbox"/> Confused <input type="checkbox"/> Struggling <input type="checkbox"/> Other: _____ <p>Evaluating My Progress Reflecting on your progress today!</p> <p>How well did I achieve my goal or master what I set out to learn? * <u>I did achieve my goal, and went even further after that</u></p> <p>What were the most important points I learned today? * <u>that overlapping shapes can be very useful in creating a picture</u></p> <p>What can I do differently next time? What kept me from achieving my goals or what worked really well? * <u>maybe set my goal higher, I achieved this goal very fast and even when further beyond</u></p> <p>What questions do I have for Mrs. B? * <u>none</u></p>
--	--

■ An example of the Google Form that students fill in throughout the lesson

SELF-REGULATION SKILLS

Victoria Berkowitz shares how she built student agency and self-regulation skills through her self-paced game maker class

Does the idea of self-paced learning sound awesome in theory, but thinking about how you would actually implement it freaks you out a little bit? That is exactly how I felt after I found an amazing platform that would allow my high-school designing animations and games class to be a self-paced course (helloworld.cc/CMUplatform). This platform offered all the wonders of pupil agency and differentiation, but I was worried about how this might affect my classroom management. How would there be structure? What would I do for learners who struggled? How would I monitor my learners' progress?

I was also taking a course on assessment, which highlighted the importance of self-

regulation — in this context, monitoring and controlling one's ability to learn — and how it is a vital skill for students' success. I had a self-paced course on my hands, but with no meaningful way to monitor my learners and their achievements, so introducing self-regulation tools and techniques seemed to be the perfect solution.

How did I do it?

I created a Google Form for my learners, broken up into three chunks: planning your day, monitoring your work, and evaluating your progress. Each of these three sections contains prompts such as, "What is my goal, and how will I know when I have achieved it?", "Am I making good progress toward my goal?", and "What can I do differently next

time?" I teach a 42-minute period, so I take the first five minutes of class to have my learners plan their day out. Then, midway through the class, I prompt them to take a three-minute break from their work to monitor their progress. The last four minutes of class are for them to reflect. I ask my learners to submit their Google Form at the end of each class, to help me monitor their progress and provide me with important insights into their abilities and challenges.

Some wins

These insights, paired with self-paced learning, have allowed me to spend more time conferencing (having one-on-one chats with pupils) and getting to know my learners more personally than traditional

teaching would have allowed me to. I could see how many of my learners struggled with content and self-regulatory skills, such as creating goals or identifying the key points of their learning. I could also see that prompting my students to reflect at different points in their learning helped tremendously, particularly by giving them the opportunity to change their approach halfway through the period. I have had learners admit they were off-task during the first half of the period, but then took the second half to turn it around and work toward accomplishing their set goals. Additionally, this chunking of time helped my learners break their big goal down into smaller ones. Specifically,

a lot more focused and goal-oriented, even outside of class.”

Some hurdles

While this strategy brought about significant benefits for my learners, I definitely faced challenges trying to fine-tune my teaching to the needs of the class. An early challenge was figuring out exactly how much time to provide for planning, monitoring, and reflecting. I started out with five minutes for each and adjusted this after a few weeks because my learners became better at answering the questions.

Another challenge I faced was getting them to produce meaningful answers. In

interaction. I would promote collaboration and camaraderie — aspects that might ordinarily be missing from a self-paced course — by pairing up a struggling learner with another learner who had overcome a similar struggle. I would also do Fun Fridays, when we did group work or partner work that helped learners to bond and have fun.

These strategies helped my learners to rely more on their peers. It also gave them a new resource to draw upon when they were stuck, which ultimately helped them to better achieve their goals that had been set within the Google Form. Camaraderie in the classroom is important because we cannot always do things alone. Self-paced learning does not have to mean isolated learning.

I would encourage you to take the risk of trying out self-paced learning, especially with embedded self-regulation. There is no easy formula to follow, but having an environment in which learners observe their teachers modelling improvements and changes helps them to do the same. I will need to keep fine-tuning my approach, but knowing that I have set up my kids to be better lifelong learners makes it worth it. **(HW)**

I WOULD TALK THROUGH LEARNERS' EXPERIENCES AND HELP THEM IDENTIFY WHAT THEY HAD LEARNT THAT DAY

my learner with ADHD improved his time management skills because he could make smaller, more tangible goals.

One of the best wins was seeing that when my learners had met their goals for the time period, they learnt to then take a well-deserved break. I was constantly checking in with them, and I was delighted when one learner said he was taking a small break because he had worked intensely to finish the set coding challenges. This ability to pace yourself and know when to take and enjoy breaks is a vital skill that I wish I had mastered before attending college!

While these changes were happening as I prompted and changed the structure of our class, I also made sure to be explicit about how we as a class were building our self-regulation skills. I would explain the benefits and research behind self-regulation, so my learners knew these Google Forms weren't just for keeping tabs on them. I would periodically ask my learners to reflect on how they had improved. One learner memorably responded, “When you said the thing about how the daily check-ins help with the future, I noticed how I've become

the beginning, I got a lot of responses along the lines of “IDK” or “I learnt the stuff.” However, I needed my learners to create specific goals with thoughtful reflections, and I helped them to do so by conferencing and modelling my thought processes. I would talk through learners' experiences and help them identify what they had learnt that day. A conferencing conversation might go something like this:

T: What do you think you learnt today?
L: I'm not sure.
T: Can you explain to me what you did today?
L: Oh yeah, I took the examples to code my own stars and then I did a coding challenge where I had to code the star on my own. I had to remember where I would put the numbers for the plotting and the colour for the filling in of the star.
T: Oh, I see. From this description, I hear you learnt how to code stars with the appropriate inputs.

My biggest challenge with implementing a self-paced course was creating class



VICTORIA BERKOWITZ

Victoria is a software developer turned computer science teacher. She works at Mineola High School in the USA, where she teaches Game Maker, Robotics, and AP CS Principles. She loves all things teaching, whether it be in public schools, volunteering, or children's ministry (@v_berkowitz).



© Daisy Daisy/stock.adobe.com

WHAT WORKS FOR GIRLS IN COMPUTING?

The **SCARI Computing** project team shares insights from teachers about how they are encouraging more girls to study computing

The SCARI — Subject Choice, Attainment, and Representation in Computing — project is researching how gender diversity in computing can be improved. Currently, only around 20 percent of young people in the UK who choose GCSE computer science (for students aged 14–16) are girls. However,

some schools appear to do better than others in terms of numbers of students, as well as the proportion of girls choosing the subject. The SCARI Computing project team collected reflections from computing teachers in some of these schools to share good practice. Teacher names and school

details have been omitted in order to preserve the ethical integrity of the research. This research is funded by the Nuffield Foundation. You can follow our social media and website for updates to this ongoing research (@scaricomp, scaricomp.org/home, scaricomp@kcl.ac.uk). (HW)

A SUPPORTIVE TEAM

Teacher A, Head of Computer Science

"Future jobs using computing skills are for all, not an elite selection. We need to show students how vital the subject is by showing how many industries it impacts: cassette videos no longer exist as we are streaming, cameras are being replaced by mobile phones, customer services are being taken over by artificial intelligence programs, and many more."

"Our department regularly speaks with students to see how they feel, asks for suggestions, and acts on them. For example, we have recently created a student team to represent the computing department, which is equally represented by boys and girls. This team handles promotion of the subject, delivering clubs to lower years at lunchtime and after school, and supporting the delivery of material to upskill parents in computing. These tasks provide wonderful

opportunities and are a very good addition to their CVs. "I am also fortunate to have a wonderful senior leadership team who support ideas to encourage all students into computing. They have supported me with parent workshops, virtual reality headset use, 3D printers in the classroom, promotion of computing leadership roles, and competitions with rewards. The team actively expresses how important computing is for today's society."

CREATIVE COMPUTING

Teacher B, Head of Department for Computing

"I believe there are several factors that are contributing to the lack of students choosing to study Computer Science GCSE, including the amount of funding available to schools to invest in IT suites and resources for extracurricular activities. There also isn't clear information about which careers are available to students

if they pursue a pathway into the industry. I also believe that the GCSE specification does not enable enough opportunities for students to be creative. Personally, I would like to see some more creative aspects added to syllabuses, such as minor game development.

I think it's important to demonstrate just how much of the process of developing algorithms is filled with making mistakes and reading errors. I have seen smiles

after working with a group of girls in the classroom and them witnessing that even as an experienced teacher, I am prone to making the simplest mistakes, like not declaring a variable or missing a semicolon somewhere. From this, they learn to focus on the fun aspect of problem-solving and not letting their mistakes discourage them from being the amazing computer scientists they can be."

CULTURALLY RELEVANT

Teacher C, Curriculum Leader for Computing

"Although we haven't quite cracked it yet, we're trying to listen to students to see how we can do more of what inspires girls to enjoy computer science and less of what puts them off. We need to reward creativity and reflection, as well as the quantity of work produced. In some ways, the media is getting better at challenging gender stereotypes in tech roles, but subliminal gender messages in advertising are still divisive and damaging. We need to counter those with persistent and personalised positivity so that all students leave the room believing it when we tell them they can thrive

in computing. The examples I use while in lessons arise out of my own (limited) life experience, so it really helps to know what students are interested in and motivated by so that I can adapt the activities to be more relevant to them. For example, while learning about the connotations of colour for effective digital media production, my class listened to a Ukrainian student passionately describe what the blue and yellow on her flag signify.

"There also needs to be an increase in the curriculum time dedicated to computing. There's so much to cover that creativity can get squeezed out to make room for all the content. Ofsted, the Department

for Education, and other national stakeholders in England should promote effective and inclusive computing provision by celebrating the benefits of students being digitally literate, computationally competent, and IT-savvy in wider society. The digital literacy and information technology strands of computing should be given equal emphasis to computer science at Key Stage 3 [students aged 11–14] and there should be a richer menu of progression opportunities at Key Stage 4 [students aged 14–16] than just GCSE computer science. I want students to be able to take pride in what they create, rather than just tick off what they've done."

CHALLENGING STEREOTYPES

Teacher D, Head of Computing

"A risk of having a limited range of young people in computing is creating a belief that it's only suited for specific groups. Computer science has a reputation for being hard, and students fear failing. If this message isn't addressed, I think some students will opt for subjects that they find easier; subjects that they believe have a higher guarantee of success. There can be quite strong misconceptions about the GCSE — that it is hard, or that it is more for boys — which prevent many

students from choosing the subject. It seems that girls generally don't want to be one of the only girls, or even the only girl, in a class, so how the options are divided is also important. If computing is offered in one strand and there are more popular subjects within this band, girl numbers are going to be a lot lower than if it was offered in more than one strand. Computing teachers, as we know, are hard to recruit. If a department has a high turnover and is unstable with teachers, I think this affects students' choices, as they don't trust they will have a teacher throughout their GCSE."

"Family can also have an influence, and contacting home to celebrate achievement helps motivate students. I've always had a department policy for teachers to contact at least two students a week, which seems to have worked well. We also introduce role models students can relate to, for example, women who have contributed significantly to the field (Margaret Hamilton, Katherine Johnson, the ENIAC 6, and so on). This helps to show that girls can and should strive (and thrive) in computing, and challenges the notion that it is a boy subject."

DEFINING REQUIREMENTS

Patricia Tanner outlines the importance of figuring out what your requirements are when you code a program, before figuring out how you write the logic

Teaching students to program a solution involves so much more than simply teaching a programming language. Instructors also need to teach students how to break down the presented problem, not only into tasks the program must perform, or requirements, but into *good* requirements.

Let's take a problem, such as creating a program to convert temperature data from Fahrenheit to Celsius and vice versa. This program needs the following:

1. The type of conversion (F to C, or C to F)
2. The formula for converting the data in either direction
3. Input from the user of the temperature to be converted
4. The output showing the converted answer

Within these steps, though, more analysis must be done to truly solve the problem and create high data quality. Questions must be asked, such as:

GOOD REQUIREMENTS

Good requirements are:

- Complete
- Unambiguous
- Clearly defined in scope:
 - What the program will do
 - What the program will not do

1. Do you want to have one program that does only one type of conversion, or do you want the user to tell the program which type of conversion to perform?
2. Do you want the program to run until the user decides to stop it, or perform only the conversion selected before ending?
3. What type of message do you want to give to the user once the conversion is completed?
4. Will you use any error checking to ensure that your inputs are realistic and conform to the program's instructions?

A program that does not consider any of the above questions is often disappointing to the end user and can be considered a failure. Researchers Thayer and Dorfman wrote: "It is a well-documented belief that the failure to develop and document good requirements specifications is the major cause of software development failures" (helloworld.cc/thayer1997).

Problem-solving

To develop good requirements, I advise students always to know what tasks and rules are needed to solve the problem before writing any program logic or code. Every program, even the most basic, is based on input, processes, and output. Firstly, students must understand what information (input) they need to solve the problem, so they can start creating the list of variables needed to solve it. Using the example of converting Celsius to Fahrenheit, the beginning list of variables starts with the type of conversion and the

temperature to be converted. There may be more variables for other tasks within this program, but these two variables are necessary inputs to perform the task.

Next, I recommend looking at any outputs from the program. Continuing with our temperature converter, what is needed to display the converted temperature (the output in this case)? Another variable will be needed for this step.

Finally, the most challenging of the steps: the processes. This is where students begin to think through the steps necessary to record a temperature, determine the type of conversion required, and then perform the conversion to generate the output. More variables will be created at this stage. When teaching this subject, my students must be able to answer these types of question before they begin coding.

When considering program requirements, it is critical to decide carefully what types of information the program should save for use at another time. Programs that have saved information that is not understandable often lead to more problems as the information moves to other programs and/or systems. For example, the temperature-converting program is not intended to be used for scientific research. Therefore, it is not necessary to allow the program to convert and record extremes of temperature ranges. Additionally, any other calculations, such as averages or statistics with the extreme data, would produce incorrect results. Early exposure to understanding data quality gives students a head start in realising the importance of data consistency, timeliness, and accuracy.

INPUT, PROCESS, AND OUTPUT (IPO) CHART

Input	Process	Output
<p>1. Beginning principal 2. Interest rate 3. Number of years 4. Number of times interest is compounded yearly</p> <p><i>[Inputs needed for program]</i></p>	<p>Amount = $P * (1 + I/M)^{N*M}$</p> <p><i>[Calculation for interest rate, i.e. the processes to be included in the program]</i></p>	<p>Ending principal plus interest</p> <p><i>[Outputs from the program needed for the user]</i></p>

Figure 1 Defining the problem with inputs, processes, and outputs

What are good requirements?

Simply having requirements for a program is not enough; we need good requirements. Every requirement should be:

1. Complete
2. Unambiguous
3. Within a defined scope

When a developer is beginning to create a new program or system, they must keep their attention focused on eliciting good requirements from their intended user community. Communicating with potential end users and developing an understanding of the issues surrounding the problem is a critical step — defining what the program will do results from the requirements. This

This chart is a guide for the development of the program; these are not necessarily the requirements. Rather, the chart helps to ensure that the programmer doesn't forget the steps and rules. The problem analysis chart shown in **Figure 1** would create the following requirements:

1. The amount of principal must be > 1
2. Interest rate cannot be $> 50\%$
3. Number of years must be > 1 and number of years cannot be > 70
4. Compound interval must be > 0 and < 366
5. Amount in savings must be $>$ principal after calculation is completed
6. Any range values for the variables listed must be approved by the user

4. The output should display:
 - a. Original principal
 - b. Interest earned each year
 - c. New principal amount at the end of each year
 - d. Final total of principal, plus total interest earned

Your programming logic will be affected by all the requirements. Good requirements equal good programs and save development time in the long run — an important lesson to teach your students early in their computing education! **(HW)**



PATRICIA TANNER

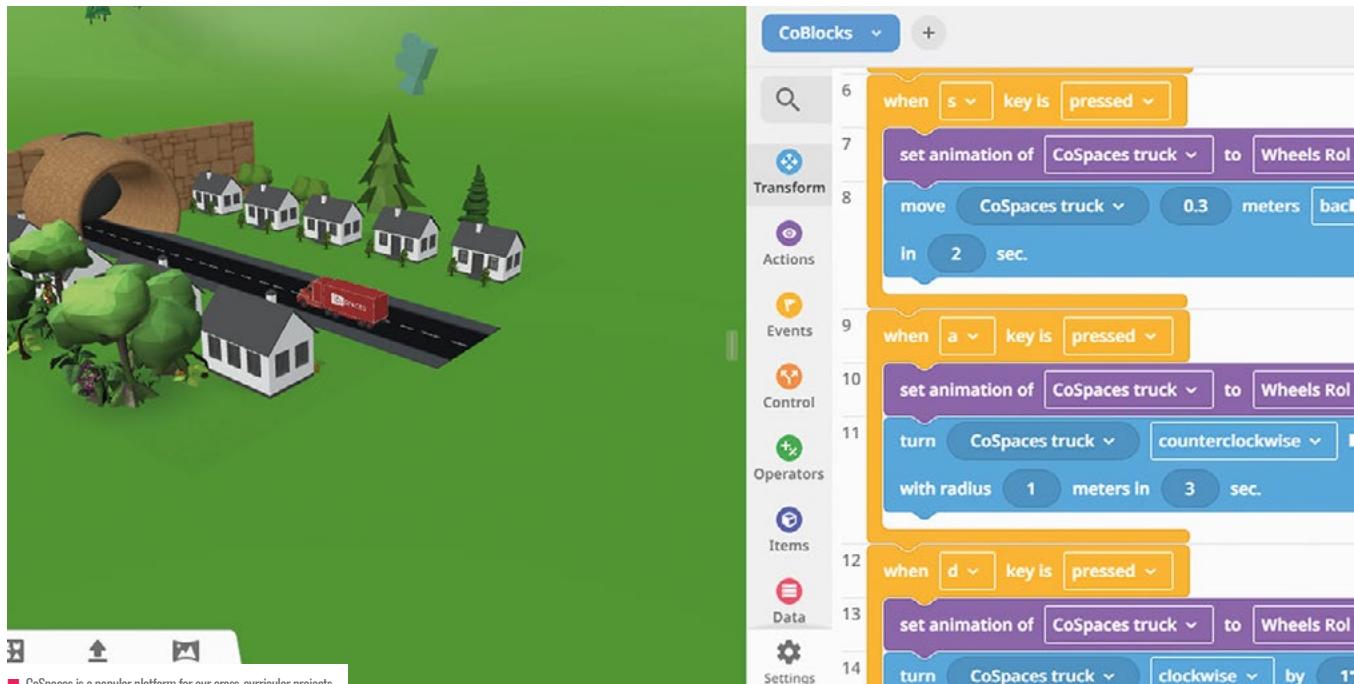
Since the early 1980s, Patricia has loved helping people of all ages learn about computing. She currently teaches at Indiana Institute of Technology, USA, as an adjunct professor. Her classes include problem-solving for programmers, information ethics, and networking and communications.

is not the time for figuring out how the program should work.

Figure 1 shows the next step: developing program logic through a problem analysis chart, using the example of calculating a principal and interest loan. The breakdown of the problem into inputs, outputs, and processes is a tremendous help in developing program logic. As the programmer completes the chart information, the program's rules become clearer.

There may be more requirements for this program, such as:

1. Use the currency format — display only two decimal places in the output
2. Allow the user to repeat the running of the program until they indicate that they want to stop running savings scenarios
3. If data entered is outside the range limits, ask the user to either exit the program, start the program over, or correct the error



CODING GOES CROSS-CURRICULAR

Shashi Krishna shares how his school is integrating coding into different subjects

Not so long ago, integrating technology into another subject meant building specific units around an identified technological skill. While this is still somewhat true, edtech tools in our classrooms have empowered students and subject teachers with embedded opportunities to make cross-curricular learning a more meaningful exercise. Last year, I presented a workshop during one of our internal professional development days to showcase the computing platforms we were using.

The history teachers found the visual appeal of one such platform, CoSpaces, relevant to their upcoming unit. Thus the discussion about integrating coding through interdisciplinary projects across the middle-school curriculum began.

Extending learning

We started by extending a unit from another subject with programming tasks, and students have benefited from solidifying concepts from both subjects. For example, Year 8 (aged 12–13) extended

their history unit on the transatlantic slave trade by building code-controlled interactive virtual reality (VR) simulations with CoSpaces. While some students chose to animate jury trials of servants who wished to challenge their contracts, others decided to showcase the experiences of captives who arrived on Western shores based on their journal entries. Students learnt how to use selection statements, loops, and custom functions to control the flow of their narratives. They also coded information and choice panels to allow users to engage with the visuals.

In another project, Year 7 students (aged 11–12) built mobile apps to advertise hard-to-sell products they had devised in their media studies class, such as garlic potato ice cream and a car with a maximum speed of 30mph. The apps contained interactive features, such as using drop-downs to allow users to view the product in different colours and sizes, links to custom-made marketing video clips, and an inbuilt click-and-collect game using variables that led to sales discounts. Students also included icons, logos,



Students coded a walk-through of one of Aberdeen's landmarks, Mercat Cross

RECOMMENDATIONS

- Having cross-curricular samples can help other teachers to think about the relevance of computing in their subject areas
- Reach out to leadership about providing ways for such initiatives to be planned and executed
- Build a resource bank of dos and don'ts (and people to seek out!) for others to consider when planning a unit
- Share quality student work with the entire community for a common understanding of the outcomes

and other visuals in their apps, which they had created as part of their media studies project. Year 8 students have also just completed their designs for their VR project, in which they coded a walk-through of one of Aberdeen's landmarks, Mercat Cross, with eco-, business-, and tourist-friendly upgrades. Users are given choice panels to navigate the space and enjoy the sights and sounds of this

for work in other subjects, such as health sciences. One student built a recycling quiz app using JavaScript in which different products are displayed with a timer and the user chooses which recycling bin they should go in, thereby gaining or losing points. Another student created a multiple-choice quiz on environmental topics. Most recently, a student wrote a quadratic equation formula-solver using Python



■ Sample student work created for media studies classes using App Lab and CoSpaces

“NEXT, WE’LL BUILD A CHOOSE-YOUR-OWN ADVENTURE GAME IN JAVASCRIPT USING ‘LORD OF THE FLIES’ AS INSPIRATION

reimagined city icon through programmed sequences. Next semester, we plan to extend literature units by examining ways to build a choose-your-own-adventure game in JavaScript with *Lord of the Flies* as a source of inspiration.

Benefits and challenges

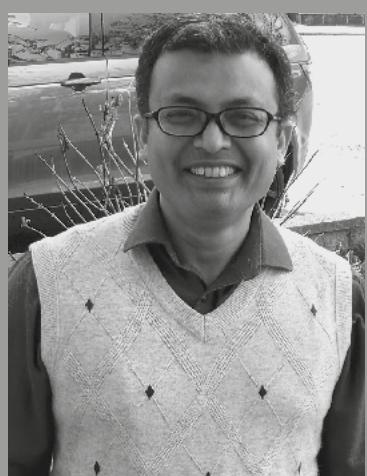
Despite the varied nature of these projects, the focus remains on students being able to convert data into information, which is the core of computing. By using programming constructs such as variables, selection, loops, and data lists, students are creating solutions that have the potential to become a resource for teachers when they revisit these topics with a new group. These custom programs can also act as a guide for those students as they engage with the work of their predecessors.

Some students have benefited from the cross-curricular projects so much that they have chosen to use their computing skills

for maths class. All these students found building a dynamic computer program more interesting than creating static visuals such as slide shows or posters to demonstrate their knowledge of these subjects.

These projects have also presented problems to solve. These include scheduling timelines for such units, and allocating planning time for teachers. To address these challenges, my school has a dedicated cross-curricular Strategic Initiative Group (SIG), which explores ways of embedding such initiatives deeper into the curriculum. Currently, we aim to run one cross-curricular computing project at the end of each term for each lower-secondary year group. Each project will be a case study aimed at a larger resource bank for teachers outside the SIG to explore further.

Programming plays such an essential role in building the technology central to our lives, and giving students an opportunity to



SHASHI KRISHNA

Shashi is a middle- and high-school computer science teacher at International School Aberdeen, UK. He has spent more than two decades in international schools teaching computer science and IT at different levels, including IB DP/MYP and IGCSE (@skrish2017).

let those skills find relevance and purpose outside the computing classroom feels like a natural next stage. 



■ Singing and dancing the *Hokey Cokey* is an established method of teaching children the basics of an algorithm

COMPUTATIONAL THINKING FOR PRESCHOOLERS

Hannah Hagon and colleagues discuss how we can teach and assess computational thinking using England's existing early years framework

England's national curriculum for computing for children aged five to seven years includes programming and coding aims, such as understanding algorithms; developing and using logical reasoning; creating simple programs using unambiguous instructions; and debugging errors (helloworld.cc/ks1&2). Researchers have suggested that preschool children (aged two to four years) can also learn some of the basics of computational thinking (CT), or problem-solving skills, so that this later formal learning can build on these experiences and knowledge (helloworld.cc/lee2022).

Many aspects of CT are already being taught in preschool. For example, we looked at England's early years foundation stage (EYFS) framework (helloworld.cc/EYFS) to see if any themes crossed over with CT themes (see **Table 1**). The framework has overarching early learning goals (ELGs) and there are clear links to CT within each of the goals. For example, singing and dancing together in the Hokey Cokey is an established method of teaching young children the basics of an algorithm (helloworld.cc/bers2014) and is recommended in ELGs about communication and language; ELGs

about literacy; and ELGs in personal, social, health, and economic (PSHE) education about interacting together. In addition, there are many opportunities for cross-curricular learning that support CT within early years education, such as ordering and sequencing in maths.

However, we believe that the focus needs to be more on how we can encourage children to think critically and problem-solve, and how practitioners can teach CT using guided play activities. We believe that CT and broader critical-thinking skills can be taught in preschools through unplugged or non-digital activities. So, how

EYFS PRIME EARLY LEARNING GOALS THAT CROSS OVER WITH CT ABILITIES

Early learning goals	Relevant CT abilities that can be taught at preschool
Communication and language	Interactions with others; self-commentary; listening; commenting; asking questions; answering questions; language development; vocabulary development
Physical	Singing and dancing; directionality; sequences of movement; robot dancing; fine motor skills; balance
PSHE	Friendships; interpersonal skills; modelling actions; language and behaviour; appropriate reactions; celebrating achievements

■ Table 1

can children in the early years progress from no knowledge to a level that infant teachers can work upon? What prerequisite themes, skills, and concepts will support formalised education? And how can infant teachers assess these skills?

Guided play and CT

In nursery and preschool environments, practitioners often teach through guided play — a range of playful activities between adults and children that give children the freedom to explore a learning objective,

but in a more systematic way. Guided unplugged play activities that provide an ideal basis for CT can take different approaches, including asking children to order a sequence and then working out together whether it was correct. For example, you could ask children to mix up icing or cut out biscuit shapes. Children may use logical reasoning (what equipment do you need?) or work out a sequence (what do you do first?). In this way, guided play can provide the processes and procedures required in many tasks and situations.

Table 2 shows an example of a lesson plan for bathing a baby doll in which CT concepts are highlighted. In this example, there can be a number of learning objectives, such as communication and collaboration; carrying out given instructions; logical reasoning and sequencing; and identifying and debugging. Teachers will have their own priorities according to the children in their groups or classes, and their ages.

Using familiar play items and activities can give adults an opportunity, through guided play, to extend children's learning and make the links to CT more explicit by introducing critical thinking, learning through debugging, making mistakes, and listening to each other. For EYFS-aged children, guided play brings a different element of interest to the children; they learn these concepts almost through osmosis. Children are focused on completing the task, but they are also learning the foundations for what will be expected in their first year of primary school.



LESSON PLAN FOR BATHING A BABY

Equipment provided: bath or bowl, jug of water, towel, flannel, soap, doll (if possible, one for each child)

Distractors: book, pen, fork (for example)

Lesson section	Activity	CT concepts and ELGs
Introduction	<ul style="list-style-type: none"> ■ Tell the children they are going to bathe a baby doll ■ Ask them about their own bath time: what do they do? ■ Ask them to choose the equipment they need (providing some distractors), and put these on the table 	<ul style="list-style-type: none"> ■ [Communication and language]: commenting, self-commentary, asking and answering questions ■ [Logic]: using logical reasoning to think about what equipment is necessary for the task ■ [Deconstruction]: abstraction, deconstruction to focus on what is needed, and removing what is surplus to requirements ■ [Cross-curricular, maths]: counting the equipment items needed
Main activity	<ul style="list-style-type: none"> ■ Check that the children have chosen the correct equipment ■ Ask what they do first; make sure water is in the bath and that the flannel is wet before it is soaped ■ Ask questions such as 'Why do we use soap?' ■ Dry the doll ■ Children put the equipment away as directed 	<ul style="list-style-type: none"> ■ [PSHE]: helping each other to choose equipment ■ [Debugging]: identifying errors and debugging ■ [Communication and language]: answering questions ■ [Cross-curricular, science]: identifying body parts ■ [Communication and language]: listening and organising ■ [Cross-curricular, maths]: how many objects can you carry?
Follow-up activity	<ul style="list-style-type: none"> ■ Ask the children, 'What items did you need to bathe the doll?' ■ Ask the children, 'How did you bathe the baby?' ■ Use pictures to show the equipment and sequence the task 	<ul style="list-style-type: none"> ■ [Sequence]: working out the sequence; recording the sequence by ordering the pictures (simple programming)

■ Table 2



■ Two children bathing their dolls in a guided play activity

► CT assessment

Assessing CT skills is not easy. How can you assess a child's critical or logical thinking? There is simply not much research in this area. There are no guidelines on what children are expected to know, or definitive benchmarks or milestones on when they are expected to know things. One approach is a scoring system based on how much adult intervention is required with specific activities, or whether a child has achieved simple success criteria — for example, has the child continued with a repeating pattern correctly? Another approach is simply talking to pupils and listening to their responses: how do they feel? Did they enjoy it? Did they learn new things? Can they talk to us about what they learnt? While such talk might not explicitly be a CT concept, this process of metacognition is valuable.

Assessment more generally in preschool is usually based on observation, and is often recorded in individual student

records with photos of particular activities and a short, written description to note the child's development stage. For more formal or summative assessments, activities can be set up so that individual children can be more accurately assessed using a holistic viewpoint across several ELGs. This can be seen in the lesson plan in **Table 2**, where a number of the CT concepts cross over with ELGs. We can also ask children to self-assess their abilities by completing a task using an 'I can...' list of learning outcomes.

The picture shows two children from a small group aged two to four carrying out the 'bathing a baby' activity. The children had to choose the equipment they needed to carry out the activity, and it was noted that this group chose everything correctly, except that they needed water (see lesson plan). The two-year-old girl (at the top of the photo) needed some help from an adult to wash the doll, whereas the four-year-old showed that she could work independently. The younger child might have found it difficult to hold the doll and wash it because her fine motor skills had not yet developed, or she may not have been very confident, whereas the older child showed that she could wipe the soap on a flannel.

This kind of guided play activity fits into the specific EYFS learning area 'Understanding the world', which involves guiding children to make sense of their physical world and their community. It would therefore not be expected that a younger child would know as much as an older child. These are the kind of assessments that a teacher who knows the children needs to make, so that they can make a 'best fit' judgement for the children's individual records. England's government guidance states that preschool staff may also complete charts of achievement for each child based on two assessment targets: emerging skills (not yet met the EYFS expected level of ability) and expected level of skills (met the EYFS level). Using guided play activities such as the 'bathing a baby' activity means that children can be assessed for their ELGs alongside CT concepts involving problem-solving and critical thinking.

We hope these suggestions are helpful to practitioners, and that they will aid the

transition of CT skills from preschool or nursery settings to the first year of primary school, therefore providing a solid foundation that will help support this important ability. **(HW)**



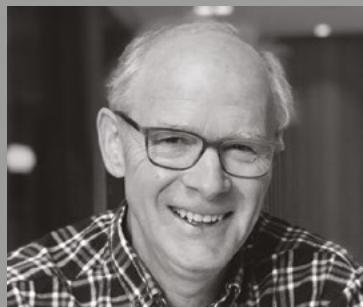
HANNAH HAGON

Hannah works in legal IT but has a passion for supporting children, parents, and teachers with screen-free activities to develop skills that will help fill the skills gap (unpluggedtots.com).



VALERIE CRITTEN

Valerie was an IT teacher in a special school and now works as a supervisor on the EdD programme at the Open University.



DAVID MESSER

David is emeritus professor of education at the Open University and visiting professor at City, University of London.

FURTHER READING

- helloworld.cc/13: turn to page 57 to read our article about introducing unplugged coding to preschool children
- helloworld.cc/critten2022: read more of our research looking at teaching programming to preschool children through guided play activities

COMPUTING SKILLS IN THE ENGLISH CLASSROOM

Jill Felbeck-Jones outlines how she integrates computational thinking and digital literacy skills into her English language arts classes

The skills we learn through the study of computing — computational thinking, ethical reasoning, and digital literacy, to name a few — are vital for all students in a world permeated by cell phones, computers, artificial intelligence, and a whole host of other technologies. However, students cannot learn all that they need to function in our technological society through stand-alone computer science classes.

According to a 2020 report by CSforAll (helloworld.cc/CSforAll2020), integrating computational thinking skills into other subjects improves student achievement in assessments in those other subjects. Educators see it as a “metacognitive strategy that deepens learning opportunities and helps students think systematically about their problem-solving approaches”. In the English language arts (ELA — reading, writing, speaking, and listening in English) classes I lead for students aged 11–14, I integrate many computer science standards so that my students learn the skills necessary to be successful citizens and stewards of technology. Here are a few of my strategies.

Flowcharts and the writing process

Algorithmic thinking, or step-by-step thinking, can be developed during writing planning exercises. I help my class to create flowcharts during the revising and editing stage of their writing process (see image). This approach allows them to break down the complex step-by-step actions they must follow when writing an essay. This algorithmic process opens students' eyes to

the metacognition (understanding of one's own thought processes) necessary to revise and edit their writing.

Respectful online interaction

In ELA classes, students develop communication skills, and I extend this to developing their online communication skills — a key aspect of digital literacy. To become good online citizens, students need to practise posting online and responding to people's comments and posts. One way that I provide an opportunity to do this is through Padlet (padlet.com). In this online bulletin board application, students can respond to questions posed during the class, comment on their learning, and learn to respond respectfully to others' posts. I teach them how to respond using the TAG structure: tell the original poster something about their post that you like or agree with; ask them a question; and then give them feedback or suggestions. I encourage students to add citations from sources to back up their ideas, too.

Laws and ethics

All students need to learn how to use the information on the World Wide Web legally, and how to govern their online behaviour ethically. While we develop research skills in my class, students learn the legal aspects of using someone else's work on the internet, and how to correctly quote and cite authors and sources. They also become aware of the social impact of using intellectual property, as they want to protect their creations.

In ELA classes, students use research on a topic to then write argumentative

papers, learning to provide evidence to back up their ideas. However, a Socratic seminar can take an argumentative essay to a more complex level by having students speak and debate about an issue. I often choose ethical technology concerns such as artificial intelligence, driverless cars, and facial recognition. To create excitement and enthusiasm for the Socratic seminar, students vote as a class for the topic they will research. They then read three articles on the subject, take notes, and finally debate the issue, citing the facts from the sources. Students learn to identify issues of concern, such as bias, malicious intent, and prejudice, while discussing possible solutions to technological dilemmas.

By integrating computer science standards in reading and writing classes, we teach students to systematically problem-solve and use these skills while completing everyday assignments, instead of siloing them to stand-alone coding classes. In this way, we can create more well-rounded digital citizens who we hope can improve technology for all. **(HW)**



An example flowchart used during the revising and editing stage of the writing process

SCAFFOLDED READING IN COMPUTING

Rachel Arthur shares some ideas on how we can teach literacy in computing lessons to improve pupils' understanding

We talk about digital literacy in computing all the time, and discuss the skills and knowledge required to be an effective, safe, and discerning user of a range of computer systems (helloworld.cc/dlreport). But how often do we consider the general literacy levels of the pupils we teach? In computing lessons, low literacy levels can impact pupils' ability to access learning. This can be because of the amount of technical language we use, the homonyms within that technical language, or even the frustration of syntax errors made purely because of a spelling mistake. In this article, I introduce the Scaffolded Reading Experience (SRE) approach, which you can use to improve literacy in

IMPROVING LITERACY THROUGH ON-SCREEN READING

In computing more than any other subject, pupils will have the opportunity to read digitally. A common misconception is that digital reading isn't as good as reading on paper or from a book. However, a report from the UK's National Literacy Trust found that for some pupils, reading on-screen may be more beneficial:

"Those with low reading engagement are more likely than those with high reading engagement to consume reading materials on screen – potentially providing opportunities to better engage them with reading in the future. For example, one in four disengaged boy readers said that they read fiction on screen compared to just one in ten of their more engaged peers (25.4% vs 9.8%)" (helloworld.cc/littrust2019).

your computing classes, in turn improving pupils' ability to access and engage with the subject.

The SRE is a framework developed by researchers Graves and Graves to support reading in lessons (helloworld.cc/graves1994). Although it may seem like an obvious structure to use, it's important to consider each of its stages when planning reading tasks in your lessons, so you can support the needs of all your pupils. The SRE is broken into three stages:

1. Pre-reading
2. During reading
3. Post-reading

We will go through each of these stages using the example of teaching cybersecurity to a lower-secondary class, imagining that you have shared a newspaper article about phishing.

Pre-reading

The role of pre-reading is to develop interest, recall knowledge, and build confidence. This can support students in accessing the reading they're about to do and ultimately reduces cognitive load. Some common activities in the pre-reading stage could include:

Explaining the focus of the reading: in this example, we are looking at a newspaper article describing a phishing attack. Before you begin teaching the lesson, you need to decide the purpose of reading the article: should your pupils focus on understanding how a phishing attack takes place? Or who is most

likely to fall victim? Or how to prevent a phishing attack? Once you have made this decision, you must communicate the purpose clearly to your class.

Whole-class recall: once the class understands the focus of the text, you can ask them to recall what they already know about the topic. This is a great opportunity to explore the technical vocabulary together before applying it.

Whole-class prediction: you can next ask the learners what they think might pop up in the text. By doing this, you're helping them to establish links between the purpose of the reading and the knowledge they already have on this topic.

Linking to real life: find a way to make the article relevant to your class. In our example, think about how a phishing scam would impact the pupils in your classroom. Where are they likely to come across one? Do they know anyone who has been a victim of one? This helps to make the article relevant and increases pupils' sense of belonging, which also increases girls' engagement in computing (helloworld.cc/child2021).

Additional literacy or computing focus: you might ask pupils to look for certain technical vocabulary or focus on another subject-related question, depending on your scheme of work and learning objectives. For example, what are the signs of a phishing scam, and can they identify them in the article? This can



■ Scaffolded reading can help boost literacy skills and improve students' understanding of computing

help pupils establish links between the learning taking place in your lessons.

During reading

Now comes the actual reading of the text, which you can supercharge by reading out loud as a class. During reading, you can:

- Pause, discuss key vocabulary, and ensure students don't confuse technical terms.
- Pull apart some key ideas in the article to help students understand the text.
- Ask elaborative questions, which will build students' understanding of the key language. These could be examples such as:
 - What would you do in this situation?
 - What could you do to prevent a phishing attack?
 - What advice would you give your friends to prevent this from happening to them?
- Show pupils what to do when they come across new vocabulary they do not understand. Making this a shared experience will normalise discussing words we don't understand and remove any shame around it.

Post-reading

The final step is to pull together all the learning that students have acquired from the text through an activity, the focus of which will depend on the purpose of the reading. In our example, the purpose of the reading was to use a concrete example to support pupils' understanding of a phishing attack. Here are some suggestions:

- Ask your class to find key vocabulary in the text. There is no getting away from the amount of complex technical vocabulary in computing lessons, and pupils' success often depends on their understanding of this language. Unpack the vocabulary using metaphors and simple language. Ensure that you then repack the language back to the technical terms, to develop pupils' literacy and technical understanding (this is called completing a semantic wave, which you can read more about in *The Big Book of Computing Pedagogy*, helloworld.cc/bigbook).
- Ask your class questions, and ask them to provide a quote from the text to show how they gained that information.
- Ask students to write a response to the text or a follow-up article on how to prevent a phishing attack in the future.

According to the UK's National Literacy Trust, a third of businesses are unsatisfied with young people's literacy skills when they enter the workforce. A similar number have organised remedial training for young recruits to improve their basic skills, including literacy and communication (helloworld.cc/natlittrust). Hopefully, using this approach to scaffolded reading in the computing classroom will help boost both your students' literacy skills and their engagement with and understanding of computing itself. Thanks to Emma King and Ben Davey for their support with researching the SRE approach. (HN)



RACHEL ARTHUR

Rachel is head of computing programme design at Teach First and she leads on the computing curriculum for Teach First programmes nationally. Before this, she was a computing teacher and assistant head teacher (@MsArthurCompSci).

USING AI-POWERED DEVELOPER TOOLS FOR TEACHING PROGRAMMING

Andrew Rice discusses the impact on teaching and learning of GitHub Copilot, an AI pair programmer that uses machine learning to write up to 40 percent of your code for you

Integrated development environments (IDEs) are the backbone of software development. An IDE gives a programmer tools to make their job easier: it can highlight mistakes in code, provide mechanisms for exploring code, and even complete code for you. Until recently, the ability to write code with AI was limited to targeted tasks such as completing a function name or filling in a piece of syntax such as a curly bracket, but today's AI tools can suggest entire lines or even blocks of code. These tools use a data-driven approach to predict what might be useful, based upon a training set containing millions of lines of publicly available code written by accomplished programmers. These include contributors to large open-source projects and even commercial companies that share their code online.

```
1 # compute modular exponentiation of a ^ b mod c
2 def modexp(a, b, c):
3     if b == 0:
4         return 1
5     elif b % 2 == 0:
6         return modexp(a, b / 2, c) ** 2 % c
7     else:
8         return a * modexp(a, b - 1, c) % c
```

GitHub Copilot

Github Copilot (helloworld.cc/copilot) is an AI pair programmer that can be used by both professional and hobbyist software developers as part of their programming workflow. It runs in an IDE (such as Visual Studio Code), and when the opportunity arises, it predicts what's coming next and suggests it in your code as ghost text (see the grey text in **Figure 1**). Users can accept this text by pressing tab, or ignore it by continuing to type. At GitHub, we think this is a game changer. Our data shows that up to 40 percent of the text produced in a programming session can be generated by Copilot, rather than typed by the developer, particularly in popular programming languages such as Python. When we surveyed Copilot users, they reported feeling more fulfilled with their jobs, less frustrated when coding, and able to focus

on more satisfying work (helloworld.cc/copilotimpact). I work at GitHub and use GitHub Copilot every day, and I'm regularly delighted by the things it can do.

A tool that provides suggestions for coding problems could really lower the barriers to entry for programming and give people a more rewarding experience as they learn. However, with disruptive, cutting-edge technologies such as AI come challenges. For example, researchers found that Codex, the AI model behind Copilot, generated solutions that placed it among the top 25 percent of students in an undergraduate cohort in an exam (helloworld.cc/finnie2022). The researchers wrote: "This technology should be of great interest to all computing educators. What we are dealing with is a freely available program that can take casually defined English language problem specifications, much like typical exam questions, and return often-correct, well-structured code that could pass as human-written." So what does this mean for teaching and learning? How should we teach programming in a world where there are tools that write the program for you?

PRIMM and AI

Some might argue that we should try to prevent learners from using AI pair programmers until they have mastered the subject. I would say that, instead, we have an opportunity to accelerate learning and lower barriers to entry. I've recently been introduced to the PRIMM (Predict–Run–Investigate–Modify–Make) approach to

■ Figure 1 GitHub Copilot suggesting an entire function body (see the grey text)

FREE FOR STUDENTS AND TEACHERS

GitHub proudly supports educators and students around the world. It offers multiple benefits to teachers, including free cloud compute via GitHub Codespaces; GitHub Classroom; and now, GitHub Copilot.

Visit education.github.com to find out more.

teaching programming. PRIMM works in stages: 1) discuss the program and **Predict** what it will do; 2) **Run** the program to test the prediction and discuss it; 3) undertake specific activities (e.g. tracing) to **Investigate** the code; 4) **Modify** the program to change

these tools during graded assessments. It's important at this stage to get students to commit to their interpretation of the code, so that they can challenge AI-generated answers later.

Run: we don't need the AI to run the code for us!

Investigate: in this stage, we typically begin by asking whether the program behaved as expected and then explore from there. Paul Denny, a researcher exploring this area, generated code explanations to help students (helloworld.cc/sarsa2022). He shared with me, "We have subsequently found that many of the code explanations, from line-by-line descriptions to high-level summaries, are very accurate and often textbook-like and perceived by students

It is totally open to whatever problem you want to solve and will make suggestions to help you. It can also provide a starting point, tailored to the problem at hand, from which to improve the program. Sometimes these suggestions are incorrect, so they do need to be thought about, but I think that is also a great skill for learners to practise. Design remains important, and the benefit of the tool is that it allows students to explore the implications of a design much more quickly.

At GitHub, we are continuing to explore the use of AI and developer tools such as code review. We can use similar technology to provide suggestions for improving a piece of code, generate a summary of a code change, or propose a strategy to solve an issue. If we can get this right, it will be great for educational use and in professional settings. The capability of these tools is only going to grow, and although they offer challenges to how we teach programming, they also offer the chance to do things in new ways. (HW)

THIS TOOL COULD REALLY LOWER THE BARRIERS TO ENTRY FOR PROGRAMMING

the functionality; 5) **Make** a new program that uses the same structures but solves a new problem. Here are my initial thoughts on how AI-powered tools might help both teachers and students, using the PRIMM framework as a structure:

Predict: the initial code educators give to students needs to be complete and well-written, and developing this is a time-consuming task. Teachers might also want to provide alternative versions to help with comprehension, and AI tools are well suited to help with this task. Even though AI pair programmers will be common by the time our students enter the workplace, they won't absolve a programmer of the need to understand a program. This requires thinking critically about the suggestions made by AI tools, and limiting the use of

to be helpful." Questions involving tracing and debugging are likely to require more student effort, as they seem to be harder for the AI tool to solve than questions asking for an explanation or annotation.

Modify: this stage helps students take ownership of a program by making amendments and increasing its functionality. There is a risk that students accept suggestions from Copilot without understanding them. Here, we may fall back on classroom techniques of written, verbal, or graphical explanations to explain their thinking and relate to the abstract concepts.

Make: this stage has the potential to be hugely rewarding, as long as students can make progress on their chosen problem. The AI could be extremely beneficial at this stage.



ANDREW RICE

Andrew is a principal researcher at GitHub, working on AI-powered developer tools with the company's R&D arm, GitHub Next (githubnext.com). He is a professor in the Department of Computer Science and Technology at the University of Cambridge, a member of the Raspberry Pi Foundation, and a founder of the Isaac Computer Science and Isaac Physics projects.

TECHNOLOGY PRESERVATION

Neil Rickus and **Anjali Das** discuss how older technology can be used to develop pupils' knowledge and understanding of computing concepts

With technology continuing to develop at a rapid pace, older technologies can easily be forgotten. The history that underpins an object's birth and evolution, though, is key to understanding its current place in society. In this article, we examine the importance of technology preservation as a vital way of telling future generations the story and impact of what has become such an important part of our lives.

What is technology preservation?

By its commercial nature, technology is often not made to last. Whether it be a device as a whole, or the parts that make up computing systems, technology is physically prone to damage. Storage media, such as punched tape, floppy disks, hard disks, and CDs, are clear examples of this. A common form of degradation affecting LaserDiscs, a storage medium from the late 1970s, for instance, is disc rot. Oxidation spots appear on the surface of the disc and render it unreadable. Similarly, the software that helps these machines work is at risk. Although it is not a physical object and cannot technically be damaged, software is stored on hardware and is therefore vulnerable to damage over time.

Conservation methods are therefore employed to prevent or limit damage to such physical objects. Environmental conditions — for example, relative humidity, ultraviolet radiation, and the temperature at which an object is stored — need to be carefully considered. Conservation scientists are currently carrying out research on this topic, and conservators focus on how to preserve the plastic that goes into a lot of these objects, such as by wearing nitrile gloves when handling

objects, which prevents dust and sweat from damaging them.

Links across the curriculum

Studying older technologies can support the development of students' knowledge and understanding of computing and other STEM subjects, providing an engaging context in which to introduce key concepts.

Teachers in the UK leading GCSE and A-level computer science classes (for pupils aged 14–18) could discuss the preservation of storage media, including its reliability and durability, when examining the benefits and drawbacks of different media, along with the need for devices to access the media. Older technology that is beyond repair can be used to examine a computer's components, such as the hard drive, memory, and processor, to enable pupils to view parts of the machine they

are studying in class. Many pupils find handling components particularly beneficial in developing their understanding of how digital devices function.

The US-based Computer Science Teachers Association (CSTA) K–12 Standards (for pupils aged 5–18) contain a number of statements that could be incorporated into discussions around technology preservation (helloworld.cc/CSTAsstandards), in particular:

■ **Grades K–2:** compare how people live and work before and after the implementation or adoption of new computing technology

■ **Grades 3–5:** discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices



Vintage technology can be used to develop children's understanding of the computing curriculum and discuss the need for technology preservation



■ Video games stored in acid-free boxes and protected in preservation-grade bags

These statements could enable children to examine and use older technology to help them better understand its impact, including making culturally relevant links, for example, understanding the way in which technology has altered communication methods in their local communities, or sharing appropriate role models related to certain technologies. In addition, students could explore the convergence of different technologies, such as how smartphones have, for many individuals, reduced the need for multiple devices.

For older children, the CSTA Standards contain the following related statements:

- **Grades 6–8:** compare tradeoffs associated with computing technologies that affect people's everyday activities and career options
- **Grades 11–12:** predict how computational innovations that have revolutionized aspects of our culture might evolve

These statements enable students to examine technology in more detail, with older technology giving children an opportunity to see how it has developed over time, as well as predicting how it might develop in the future.

If the statements within your computing national curriculum offer limited opportunities to explore technology preservation, you could make links with the topics studied in other subjects. For instance, in history and geography, the technology, or the people involved with

the technology, could form part of an extended project. Pupils could also examine the environmental impact of disposing of technology, making links to recycling and the Right to Repair movement (repair.eu). Finally, the environmental and handling conditions of old technology can also provide links to science and geography, along with providing real-world contexts for scientific investigations. Ultimately, though, it might just take a little creativity to link this context to your computing curriculum — for example, you could discuss how old technology is physically moved; pupils could produce algorithms for how items can be safely carried from one location to another.

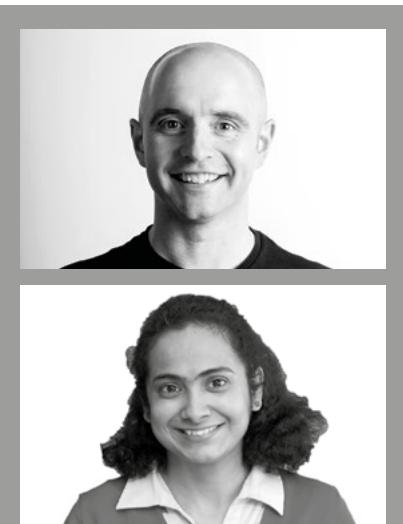
Opportunities for enrichment activities

Technology preservation can also be examined in a number of settings away from the classroom. Maker spaces allow people to investigate older technology, including repairing non-functioning items or upgrading older hardware to enable it to operate with more recent technologies. For example, within home computers, failing floppy disk drives can be replaced with SD card readers, or antique radios can be modified to work with streaming audio services. Working with technology in this manner can enable students to develop practical skills, as well as developing their understanding of how devices function.

Computer museums preserve our digital heritage by displaying examples of older technologies and telling the stories that accompany them. One such example is The Centre for Computing History in Cambridge, UK, which showcases the historical, social, and cultural impact of personal computing in our lives (computinghistory.org.uk). For instance, one article on the museum's website recently examined the importance of video game preservation, particularly its role in showcasing popular culture of the time (helloworld.cc/gamepreservation). The article discusses what it means to preserve a video game, and considers both physical and digital games, as well as the marketing that surrounds them. Educational groups visiting the museum experience hands-on learning, with strong links to England's national curriculum for computing, wrapped up in a historical context.

Museums and other organisations also use computers and video game consoles to enable children to have hands-on experiences with technology. For example, The Code Show works with schools to enhance the taught curriculum by bringing old technology to schools to examine how it has developed over time (thecodeshow.info). These experiences could even introduce programming concepts to children; through considering the programs used in a game, pupils can discuss the associated algorithms and how programming concepts have been implemented, then go on to implement programs within a block-based programming environment.

Are you incorporating, or considering incorporating, technology preservation into your schemes of work? Please get in touch with us on Twitter: [@computingchamps](https://twitter.com/computingchamps) (Neil) and [@MuseumsInspire](https://twitter.com/MuseumsInspire) or [@computermuseum](https://twitter.com/computermuseum) (Anjali). (HW)



**NEIL RICKUS
AND ANJALI DAS**

Neil is a senior lecturer in computing education at the University of Hertfordshire in the UK and the founder of Computing Champions ([@computingchamps](https://twitter.com/computingchamps)). Anjali is the head of learning at The Centre for Computing History in Cambridge, UK, where she leads the development and delivery of a variety of learning opportunities focusing on computing education for young people ([@computermuseum](https://twitter.com/computermuseum)).



DEVELOPING A SENSE OF PROGRAMMING

Alex Hadwen-Bennett shares some key findings from his PhD research exploring how visually impaired learners develop their understanding of programming concepts

My recent research has explored how visually impaired learners develop an understanding of key programming concepts when working with physical programming languages such as Code Jumper. In this article, by considering the processes through which these students become increasingly sophisticated programmers, I also explore the implications of this research for how we view progression in programming for all learners.

My research drew on Vygotsky's distinction between an individual's sense of a concept and the externally defined meaning of that concept (helloworld.cc/vygotsky1978). The meaning of a concept

is culturally defined and usually relatively fixed, whereas sense is unique to the individual, shaped by their experiences, and relatively fluid in nature. In my work, I demonstrated how visually impaired learners develop their own unique sense of programming concepts, shaped by a variety of factors, including their experiences, programming tools, and activity design. A learner's sense of a concept may differ from the agreed definition, but, nonetheless, traces of this definition are expressed in actions, gestures, and words.

The role of tools

The programming tools learners use when developing their sense of a concept

become a part of their sense. It could be argued that when a learner transitions to a different programming tool, for example, when moving from a block-based to a text-based language, they simply leave the old tool behind, like a scaffold that can be taken away. My research suggests that, in reality, the original tool remains a part of their sense. This was demonstrated in my research through learners' use of gestures which emerged during their interactions with the original tool. For example, when answering questions about repetition, many of them produced a looping gesture, seemingly influenced by the form loops take in Code Jumper. The re-expression of these gestures indicates that learners

were drawing upon, or even re-enacting, the original tool when they applied their sense to the new tool. In this way, one set of programming tools does not replace another; rather, the introduction of new tools seems to enrich a learner's sense of programming concepts.

With this in mind, we need to be careful how we frame different tools in relation to each other. For example, there has been much discussion regarding the transition from block-based to text-based languages in recent years, with the implication that block-based languages are less sophisticated. This could lead learners to believe that the sense they developed when using a block-based language is less advanced. Different programming tools

ways, many of which may not usually be considered when evaluating a learner's progression. In programming education, the successful creation of a program to solve a given problem is often used as a measure of progress. In addition to this measure, though, the learners in my study also employed gestures that indicated their understanding of the flow of control within the program. For example, they touched each instruction in turn, in the sequence in which they would be executed. They also demonstrated their understanding by making connections to their experiences from other domains — for example, one learner compared selection to a junction on a railway track, while others made a link between nested loops and multiplication.

can also be re-enacted, impacting on the learner's engagement, confidence, satisfaction, and enjoyment.

A key influence on affect in learning situations is problem design. In my study, it became clear that when we design problems so that learners can relate to them, they are more likely to persevere in the resolution process, even when quite complex programming techniques are involved. When they were building something they cared about, they frequently treated false steps as amusing, rather than frustrating, and they remembered their results with pleasure and even pride when they re-encountered similar challenges. The role that affect plays in sense is therefore continually evolving as the learner engages in new activities.

Perhaps these results can also apply to teachers; their unique sense of a concept will have emerged from their own experiences. My research led me to reflect on my individual sense of different concepts and how I felt about the process by which I had learnt them. This has impacted on my teaching. As I reflected on the affective side of my own sense of a concept, I became more aware of how aspects of the negative and positive experiences that accompanied my own conceptual development were re-felt as I taught the concept in question. I would recommend that such reflections might play a fundamental role in the development of inclusive approaches to programming. [\(HW\)](#)

VISUALLY IMPAIRED LEARNERS CAN EXPRESS THEIR SENSE IN VARIOUS WAYS

have different strengths, and different types of problem can be solved with different tools. This does not necessarily mean that a particular tool should be considered more advanced than another; rather, they have different features that make them more or less suitable for certain types of problem.

My investigations have indicated that it would be beneficial to give learners access to a range of programming representations to work with, to create a more inclusive environment in which to learn to program. In such a scenario, you may have some students working with physical representations and others working with visual or textual representations. Physical representations in education are often associated with younger learners or those with special educational needs. However, there is nothing to suggest that sense developed using a physical tool is any less sophisticated than that developed using textual representations. We therefore need to consider carefully how we frame such tools.

Expressions of sense

My research demonstrated that visually impaired learners can express their sense and understanding in numerous

The expressions of sense that learners produce will vary, and the form of the programming tool they use, whether it be physical, visual, or textual, will influence them. I recommend that computing educators remain open-minded regarding how their learners may express their understanding of a concept. Educators should not assume that learners do not understand a concept if they do not express their understanding in the expected, conventional ways.

The importance of affect

It is also important to consider the role that affect (feelings or emotions) plays in the development of sense. Teachers often consider the importance of creating engaging activities to improve motivation. However, my research suggested that affect should not be reduced only to motivation; rather, the feelings we experience during the development of our sense of a concept become part of the sense itself. As discussed earlier, the tools used in the development of the sense of a concept can be re-enacted internally when the learner encounters the concept in the future. In a similar manner, the affect associated with the activity of programming

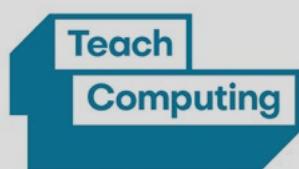


ALEX HADWEN-BENNETT

Alex is a lecturer in computing education at King's College London, where he undertakes research focusing on inclusive computing education and the pedagogy of making ([@drhbcomputing](#)).

The following lesson plan is taken from the six-lesson ‘Networks’ unit of the Teach Computing Curriculum (TCC), aimed at learners aged eleven to twelve. This unit asks students to imagine a world without computer networks; there would be no more YouTube, Google, instant messaging, online shopping, or Netflix. It begins by defining a network and addressing the benefits of networking, before covering how data is transmitted across networks using protocols.

ABOUT THE TCC



The Teach Computing Curriculum is part of England’s National Centre for Computing Education and supports students aged five to sixteen.

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.

NETWORKS AND PROTOCOLS

Introduce learners to computer networks and the concept of a protocol

AGE RANGE

11–12 years

OBJECTIVES

- ✓ Define what a computer network is and explain how data is transmitted between computers across networks
- ✓ Define ‘protocol’ and provide examples of non-networking protocols

This is the first lesson in the TCC’s ‘Networks’ unit for students aged eleven to twelve. In this lesson, learners will start thinking about the history of different communication methods, along with the meaning of the word ‘protocol’. (B&W)

ACTIVITY 1: DEVICES CONNECTED TO THE INTERNET 3 MINUTES

Ask learners how many devices are connected to the internet; ask them to write their answers on a sticky note. You will not give students an answer until the end of the lesson — they can revise their response at any point during the lesson, but they should attempt to answer the question at the start.

ACTIVITY 2: HISTORIC COMMUNICATION METHODS

5–10 MINUTES

As an introduction to this topic, get learners thinking about the history of different communication methods. Display Figure 1 to students, which shows five communication methods, five images, and five years. Ask learners, either in groups or individually, to identify which method goes with which image and year.

You could share some interesting facts about each communication method to engage learners early on. For example:

Carrier pigeon: carrier or homing pigeons are birds that have been trained to find their way back home over great distances. Carrier pigeons were historically used to send short messages on a small piece of paper attached to the leg of the bird. They were used by armies to communicate in battle. Some pigeons can fly at over 60mph. By 1167, a regular carrier pigeon communication service had been created between Baghdad and Syria. Over



Figure 1 Historic communication methods

ACTIVITY 2: HISTORIC COMMUNICATION METHODS (CONT.)

54,000 pigeons were used in World Wars I and II to relay secret messages.

Semaphore: semaphore was first introduced in France in 1792, when the country was embroiled in a revolution. The French government needed a way to send and receive messages quickly. Claude Chappe (along with his four brothers) created a system that was made of towers placed less than 20 miles apart. Each tower had two moving wooden arms and a crossbar. The position of these arms and the crossbar could be changed, and 196 different symbols were created based on these differing positions. A signal displayed by one tower would be observed by an operator using a telescope at the next tower, who would then make the same signal for the next tower — and so the message was sent from tower to tower. Operators needed to regularly check their nearest towers and recreate the signal as quickly as possible. It was a very tiring job.

Telegrams: in 1837, two sets of inventors simultaneously developed an electrical telegraph: Wheatstone and Cooke in England, and Samuel Morse in the United States. Morse went on to develop an alphabet using dots and dashes that became known as 'Morse code'. By 1861, this Morse telegraph system connected America's West and East Coasts.

Telephone: Alexander Graham Bell is commonly credited as the inventor of the telephone, though many individuals contributed to the devices that we use today. Alexander Graham Bell filed for the patent of the telephone system in 1876.

Email: the very first version of what we now know as email was created in 1965 at the Massachusetts Institute of Technology (MIT). This allowed users to share files and messages on a central disk on a server, logging in from remote machines. In 2023, more than 347.3 billion emails will be sent daily. At least half of them will be spam.

ACTIVITY 3: INTRODUCTION TO COMPUTER NETWORKS 5 MINUTES

Share the following definition of a computer network with learners: 'A computer network is when two or more computers are connected together to allow them to communicate.'

Ask students to think about what daily activities use computer networks and share their ideas in pairs. You might want to give learners examples, or ask them to split their answers between different times of the day.



ACTIVITY 4: A BRIEF HISTORY OF COMPUTERS AND NETWORKS 5–8 MINUTES

Explain to students how computers have evolved significantly over the last 70 years. Discuss the following timeline with learners, which shows some of the key milestones in the growth of computer networks:

- **1950–1970:** mainframe computers grew in popularity. These were single computers that supported multiple users or terminals connecting to them at the same time. These machines were huge and expensive.
- **1969:** the internet was brought online. It was initially called the ARPANET, and it was a project run

by the Advanced Research Projects Agency (ARPA) to connect four universities in the USA together.

- **1974–1977:** the evolution of smaller circuits, microchips, and smaller components such as keyboards and mice, as well as a reduction in cost, enabled the first PCs to be built and purchased by individuals.
- **1989:** a British scientist called Tim Berners-Lee invented the World Wide Web. The original purpose of the web was to allow the automatic sharing of information between scientists in different universities around the world.

■ **1999:** Nokia brought out a revolutionary new mobile phone called the 7110. This was the first mobile phone in the world that could access information from the internet. The mobile phone thus became a global communication device. Users could check weather reports as well as email. However, all information was text-only.

Now ask your learners the same question as at the beginning of the lesson: ‘How many devices are connected to the internet?’ Ask students to review their answers in pairs — if the learners’ answers differ from the start of the lesson, ask them to amend them.

ACTIVITY 5: A MESSAGE TO AUSTRALIA 10 MINUTES

Message transmission without using computer networks

Ask learners to discuss how they might send a message from their location to a person in Australia without using computer networks. Ask them to try to think of multiple methods, as well as the information that they would need to get their message to their intended recipient.

The most straightforward discussions might focus on an individual writing a message and then travelling from the UK to Australia to deliver it (via plane or boat, for example). More complex discussions might involve using more than one person in different countries on the way to Australia, who can each travel a short distance before passing the message on to the next person. For this scenario, the recipient’s name and address would need to be included — this would be a rule for the communication to work. Discussions may also include communication methods mentioned in Activity 2.

Message transmission using computer networks

Now ask learners to discuss how they might send a message from their location to a person in Australia with the use of computer networks. Ask them to think of multiple methods, along with the information they would need to get the message to the intended recipient.

Discussions will most likely be based around email, but could include instant messaging, FaceTime or Skype. Most forms of computer-based communication would require both the sender’s email address and the recipient’s email address. This would be one of the rules of the communication.

Message transmission using letters

Ask the learners to think about what happens when a letter is sent (which may have been a common answer when discussing sending a message between the UK and Australia without computer networks). With luck, learners will discuss

using post office depots that are located at key intervals along the journey. The letter is passed from depot to depot until it reaches its final destination.

Message transmission using email

A common answer to the second scenario, in which computer networks existed, should have been ‘email’. A common misconception is that messages are transmitted directly from one machine to another across a network or the internet. Learners may believe that when an email is sent, it simply ‘enters the cloud’ and then appears at the recipient’s machine, or that it travels directly along a cable from point A to point B. Neither is correct.

Instead, a message may pass through many intermediary devices along the way, which then pass the message on. This is exactly the same as how a letter may travel across a country or the globe, going from one post office depot to another until it reaches the recipient.

ACTIVITY 6: PROTOCOLS 5 MINUTES

Describe the meaning of the word 'protocol' in this context: 'All methods of communication need rules in place in order to pass on a message successfully. These sets of rules are called protocols.'

Then ask learners the following questions:

- What protocol exists for meeting someone new?
- Is this the same in all countries?

RELEVANT LINKS

TCC 'Networks and protocols' lesson:
helloworld.cc/networkslesson

ACTIVITY 7: CLIMBER AND BELAYER PROTOCOL 10 MINUTES

Give learners the series of standard commands (i.e. a non-networked protocol) between a climber and a belayer, and ask them to put them in the correct order to ensure the safe descent of the climber (**Table 1**). The answers are given in the fourth column of the table.

Command	Person	Meaning	Order number
Belay on	Belayer	The rope is secured and slack taken up so it is tight.	(2)
Climbing	Climber	I wish to climb.	(3)
Off belay	Climber	I wish to be detached from the rope. The climber only says this command when it is safe to do so at their destination.	(6)
Belay off	Belayer	The belay rope has been disconnected.	(7)
Climb on	Belayer	I am happy for you to climb.	(4)
Slack	Climber	I need some slack rope in order to climb up.	(5)
On belay?	Climber	Is the rope secured?	(1)

■ Table 1

ACTIVITY 8: COMPUTER PROTOCOLS 10 MINUTES

At this point in the lesson, learners need to make the connection between non-networked protocols for communication and networking protocols. They need to understand that, just as we have rules that we follow to ensure our communications are successful, so a computer has rules for communicating with another computer across a network. For example, we must use a postal address when we post a letter. The format of this address follows rules.

It must include a house name or number, a street, a city, and a postcode. Similarly, computing rules exist when using email addresses and web addresses.

Ask learners to answer the below questions in a think-pair-share format:

Question: what rules exist for an email address?

Possible answers:

- We need to use an @ symbol

- The email address must be unique

Question: what parts of this web address show a protocol being used?
<http://www.bbc.co.uk>

Possible answers:

- All website addresses start with http:// followed by www
- All website addresses are unique
- They use dots to separate each part of the address

ACTIVITY 9: DEVICES CONNECTED TO THE INTERNET 5 MINUTES

Ask learners if they would like to amend the answer they gave at the start of the lesson asking how many devices are connected to the internet. Then ask some learners to share their responses to this question and

report whether their answer changed over the course of the lesson. If their answer did change, ask them why.

Reveal the answer of 27 billion+. Learners may find this figure surprising, given that

not every person on the planet has access to the internet. You may wish to discuss the fact that many of these devices, such as smart meters, are not used by individuals but are part of the internet of things (IoT).

AGE RANGE

14–19 years

REQUIREMENTS

- Raspberry Pi (any model with current Raspberry Pi OS)
- A used/redundant wireless router
- An ethernet cable
- A USB memory stick
- Windows 10 laptop with Wi-Fi access

EXPERIMENTAL LAN

This project shows you how to build an experimental LAN that students can play and learn with, causing no technical harm

Students will learn how to make a simple LAN (local area network) using Raspberry Pi and an old/used domestic wireless router. They will set up Raspberry Pi as a web server and connect it to the wireless router. This activity will help them understand the

relationship between a web server and a web browser. They will also have an introduction to the TCP/IP network protocols that make the internet work. In the next issue, I will share some fun follow-up activities in which students hack into Raspberry Pi. **(HW)**

ACTIVITY 1: SETTING UP PI AS A WEB SERVER 30 MINUTES

Depending on the time available, you could set up Raspberry Pi as a web server before the lesson, or do this in class while you explain the process step by step:

1. Connect Raspberry Pi to the internet using an available network's Wi-Fi service.

2. Make sure Raspberry Pi is up to date by opening a terminal window and entering:

```
sudo apt-get update  
sudo apt-get upgrade
```

3. When that has finished (it may take a while), install the Apache web server

```
software: sudo apt install  
apache2 -y
```

4. Check that the web server software is running by obtaining the IP address of Raspberry Pi. To do this, enter:

```
hostname -I
```

5. Enter the returned IP address (e.g. 192.168.0.1) into the address bar of a laptop connected to the same Wi-Fi network. It should load the Apache2 Debian default page. This demonstrates that Raspberry Pi is successfully acting as a web server and serving a static HTML page to the laptop web browser.

ACTIVITY 2: BUILDING THE LAN 30 MINUTES

Using the Raspberry Pi web server, an ethernet cable, the wireless router, and a laptop, you can now build a LAN that your learners can safely experiment with.

Power up the wireless router. Turn off the Wi-Fi on Raspberry Pi and plug it into the router using the ethernet cable. On the laptop, connect to the router's Wi-Fi service. If you are asked for a password, you may have to reset the router to its default settings. Usually there is a small hole in the case into which you can insert a small pin.

Wireless LAN adapter WiFi:	
Connection-specific DNS Suffix:	: Belkin
Description	: Intel(R) Wi-Fi 6 AX200 160MHz
Physical Address	: 04:ED:33:E5:2B:03
DHCP Enabled	: Yes
Autoconfiguration Enabled	: Yes
Link-layer Link-Local Address	: 192.168.0.100 :9dd1:8d:8346:791a%8{Preferred}
IPV6 Address	: 192.168.2.3%3{Preferred}
Subnet Mask	: 255.255.255.0
Lease Obtained	: 07 December 2022 22:46:55
Lease Expires	: 07 January 2023 19:15:44
Dhcpv4 Client DUID	: 192.168.2.1
DHCPv6 Client DUID	: 91541634
DNS Servers	: 192.168.2.1
NetBIOS over Tcpip	: Enabled

Figure 1

Once connected, open a command window on the laptop (Windows key + R) then type cmd and enter: **Ipconfig/all**

CHRIS WALKER

Chris is a technology tutor at Tonbridge School, Kent, UK, in their digital creativity department. He has twelve years' experience tutoring and supporting GCSE computer science, specialising in physical computing resources.

ACTIVITY 2: BUILDING THE LAN 30 MINUTES (CONT.)

This will display the IP settings allocated to our laptop by the router. Note the IP address of the default gateway, which is the IP address of our router (**Figure 1**).

Enter the address (192.168.2.1, in this example) into a browser address bar on the laptop. This should load the admin window of the router and will now give you access to the Wi-Fi router set-up utility.

To make any changes, you will need to log in to the router. You will need to know the router password if it is not blank or 'admin'. If this does not work, you may have to reset the router (as before, there should be a reset pinhole on the side of the router) so that it adopts its default settings.

Next, connect Raspberry Pi to the wireless router using the ethernet cable. The router will recognise that it has a new device connected to it and will use the DHCP (dynamic host configuration protocol) to allocate Raspberry Pi an IP address.

With the router set-up utility, open the DHCP client list option and you should see a list of all the devices connected to the

router. In this example, Raspberry Pi has been allocated IP address 192.168.2.2 and the laptop has an IP address of 192.168.2.3 (**Figure 2**).

At this point, you may wish to remove the password for the Wi-Fi by finding the wireless security tab and disabling the security. If you now enter the IP address of Raspberry Pi into a browser address bar on the laptop, you should see the default Apache web server home page.

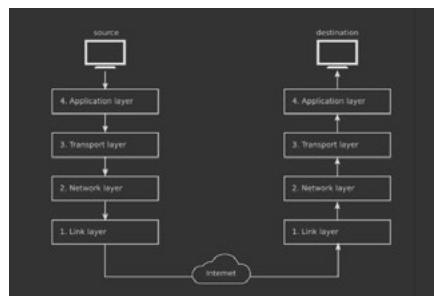
Your LAN is now working and serving a web page from the Pi web server, over the ethernet cable to the router, which routes the page to the laptop over a Wi-Fi connection!

Router Setup Utility		
LAN > DHCP Client List		
IP Address	Host Name	MAC Address
192.168.2.2	raspberrypi	b8:27:eb:c3:30:b2
192.168.2.3	CDW-X360	04:ed:33:e5:2b:03

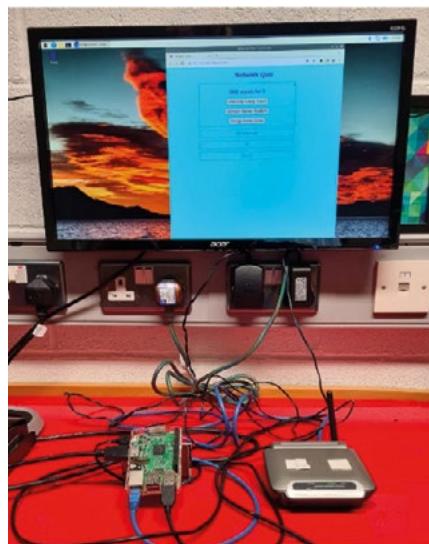
■ Figure 2

ACTIVITY 3: EXPLORING TCP/IP NETWORK PROTOCOLS 30 MINUTES

Using this model LAN, you can now explore how TCP/IP protocols apply to the model using the four-layer TCP/IP stack. You can then ask students to identify the network components in our LAN (web browser/server, Wi-Fi router, ethernet cable), and how they relate to the TCP/IP stack (application layer, transport and internet layers, and link layer).



■ The four-layer TCP/IP stack in operation



■ Our test LAN showing the Raspberry Pi web server, the Wi-Fi router, and a web browser displaying a page served from our Raspberry Pi

DIFFERENTIATION

To challenge students further, you could explore some of the router options. For example:

- Change the SSID (router Wi-Fi name)
- Change the IP lease time

Remember, if it all goes wrong you can press the reset button and restore the router default settings.

ASSESSMENT

To check students' understanding of the relationship between their LAN and TCP/IP protocols, you could present them with some multiple-choice questions:

What layer of the TCP/IP stack does the HTTP web browser and web server represent?

- Application
- Internet
- Transport
- Link

Which item represents the link layer?

- Wireless router
- Ethernet cable
- Pi web server
- Web browser

One of the components in our LAN is a wireless router. In our demonstration model, it also acts as an ISP to our connected devices. **What range of services does an ISP provide?**

- Access to the internet
- Firewall
- Antivirus software
- Web server

PREPARING A BACKUP PLAN

Alan O'Donohoe describes a range of backup activities teachers can fall back on when lessons can't proceed as planned

In those incredibly stressful teaching moments when technology fails you, what's the best strategy? Do you choose to wait in the hope that you can quickly resolve the issue, dive into fixing the problem, or abort your plans altogether? A technological issue may last only a few seconds, but time has an annoying habit of dragging painfully slowly as you try to diagnose and fix the problem while maintaining a sense of calm in the classroom. The goal of this article is to expand the range of lesson backup strategies you can rely on

when lessons can't proceed as planned. The guide describes some tried-and-tested activities you can make use of with minimum preparation and zero reliance on tech.

Readers are advised to trial these activities with teaching groups at the earliest opportunity — don't wait for things to go wrong before trying them out! This will give you a less stressful opportunity to judge how well they work for you and allow you to refine and develop them. You may discover a particular favourite which you keep up your sleeve, ready to implement at

ELEVEN

This activity is loosely based on a game called Nim. Like Hack Attack (see page 86), you can play it anytime and anywhere, and it requires no resources. It's a problem-solving game that requires participants to apply logical deduction and pattern recognition. It is suitable for two or more players. Each individual round lasts about two minutes. As multiple rounds are played, players can develop and implement strategies.

Goal: players need to avoid being the player who is forced to say 'eleven'.

Instructions: all players stand up until they are eliminated from play, and the last player standing is the winner. The moderator points to the first player, who plays their move, followed by each subsequent player in turn. The first player can say up to three numbers in sequence, starting with number one — for example, 'one' or 'one, two, three'. The next player continues the sequence — for example, 'two' or 'four, five'. On their turn, any player can say up to three numbers at a time, but they must avoid reaching the number eleven. Players whose turn it is to say 'eleven' must

do so and must sit down after being eliminated, at which point the next round begins again at the start of the sequence. In the final round, the two remaining players battle it out to see who the champion will be.

Variations: the teacher can introduce many variations into the game, and can also invite the class to predict likely outcomes. For example, you can divide the class into teams. In a two-team game, each team member takes their turn against a member of the opposing team. Teams can either score points for each win, or team members can be eliminated when they lose, until all the team members are out.

As learners quickly become deeply engaged in the game, they will need reminding to apply their skills, such as pattern recognition, logic, and prediction. There are other learning connections that can be made in the game, including abstraction, decomposition, algorithms, and flow control.

The advantage of this game, compared to Hack Attack, is that it's a fast-moving game that can be played in a few minutes. The rules are very easy to pick up, and students may also have already played something similar before.

- Three pieces of information, two are true.
- Work together to decide which is false.
- Allowed one question each.
- Strategy, Collaboration, Logic.

[Breakfast](#) [Click](#) [About](#)



ONE FALSE, TWO TRUE

This activity requires a bit of prior planning, but like the others listed, requires no additional resources and provides plenty of practical opportunities for students to apply computational thinking skills to an unplugged scenario.

Goal: students are challenged to identify which presented fact is false.

Instructions: the teacher starts by informing the class that they will present three brief facts about themselves, only two of which are true. For example, ‘I used to work for GCHQ, I appeared on BBC News, and I appear in a famous rock star’s autobiography.’ Students are then asked to collaborate in small groups to agree which fact might be false and propose some questions for the teacher to answer. The teacher will take one question from each group. When answering the questions, warn students you will answer truthfully, but that you might avoid answering certain questions, particularly if you are asked, ‘Which one is false?’

You should remind the class to approach this challenge from a computer science perspective in order to establish facts, for example by applying logical reasoning and abstraction.

When I play this game, I encourage students to demonstrate acute listening skills during the questioning, as well as computational thinking skills, and I praise them when they do. Sometimes, students are so focused on asking their own question that they don’t hear the same question being asked by another student and answered. Some students will ask interesting questions that are not relevant (such as ‘How old are you?’), in which case they need reminding of the goal of the activity!

It’s important to reveal at the end which of the facts was false and, if time allows, to provide some feedback on the types of question the class asked. You can then ask students to devise their own three facts so that they can practise on each other. Discuss strategies with students, for example how they might best include two unbelievable truths as well as a plausible falsehood.

TILE ALGORITHMS

Goal: learners develop, test, and evaluate a turtle-style algorithm to be used in their classroom.

Instructions:

Design: ask all students in the class, without physically moving, to choose a start position for themselves in the room. They could, for example, identify either a floor carpet tile or a ceiling tile, but a piece of paper will also work. Then ask them to develop an algorithm, using only two movement controls, that will describe a path from their current position to the exit door.

They can use both of the following movement controls:

- FORWARD n, where ‘n’ is the number of tiles to move. FORWARD 2, for example, moves two tiles in the direction the player is facing.
- CLOCKWISE n, where ‘n’ represents one 90-degree clockwise rotation. CLOCKWISE 4, for example, rotates the player completely so that they are facing in the same direction as they started.

Testing: ask students to test their algorithms. It can be a little chaotic if you let them all test their algorithms at the same time. Instead, ask one student to follow instructions while another reads out the algorithm.

Prediction: ask students to predict what would happen if a student seated in another part of the class followed their algorithm. Inevitably, they only succeed if they start at the algorithm designer’s origin.

Extension: ask students to work in groups. Explain that adding one more movement control to their algorithm could mean they could navigate from any position in the room to the exit. Students need to decide what the additional control is, and then develop and test an algorithm that uses it. The teacher may need to provide additional support by suggesting a control that can sense, such as collision detection. Remind students how, in addition to sequence, an algorithm might include conditionals and iteration. The additional control needed could be ‘if touching object’ or ‘while not touching object’. This isn’t completely flawless, as some room layouts could mean a person following the algorithm will be trapped in a corner forever!

HACK ATTACK

> This activity is based on a modification of Werewolf, a game that has proved to be popular among the tech community (see helloworld.cc/werewolfgame for more on this). It lends itself to being played anytime and anywhere, with no special requirements for equipment. It's a deeply engaging problem-solving game that requires participants to apply computational concepts such as logical deduction and pattern recognition. It can be played with groups of different sizes, and each game can last 15 minutes or longer, depending on the number of players and how much discussion the moderator allows.

Goal: players collaborate to identify and then eliminate cyberthreat actors from their online community.

Instructions:

Night: at the start of the game, the teacher is the moderator and all other players take on the role of network users. The moderator instructs all users to log off the network and go to sleep, as it is night. Users must all close their eyes and put their heads down and listen attentively for suspicious-sounding activity. As users 'sleep', the moderator secretly nominates three players at random by gently tapping on their heads. They become cyberthreat actors (or hackers) who then wake up in silence and make themselves known to each other. Next, without speaking, they select one innocent user to be the victim of a coordinated cyberattack, which they communicate to the moderator before sleeping again.

Day: the moderator announces the arrival of daytime, and as all users wake from their sleep, the user who was targeted in the attack is identified to all as the victim. The moderator explains that the victim can no longer communicate with anyone, except to utter their final words, "I've been hacked." This player is out of the game now; they need to keep silent for the remainder of the game and are strictly forbidden from communicating with other players.

The moderator warns users that the hackers are cunning and deceptive. While they remain undetected, they pose a constant threat and will continue to target more innocent users. If challenged, they will protest their innocence and use lies and tricks to preserve their anonymity.

Discussion and elimination: users are urged to use logical reasoning and pattern recognition (computational thinking skills) to deduce which members of the group are likely to be the hackers. For example, focusing on pattern recognition, educators could prompt students to discuss questions such as: are any players behaving differently during the game? Which players appear to be most nervous? Has a player provided unnecessary detail in their answers? Have any night-time noises made you suspect a player? Users then nominate potential hackers and explain their reasoning. The moderator puts it to a vote and the player with the most votes is removed. The moderator explains: 'All their devices are confiscated and network connections terminated, meaning they can no longer communicate with others.' This player is now eliminated and their true identity is revealed. Unfortunately, this may result in an innocent user being eliminated, so users need to be careful who they accuse!

Night again: the process is repeated and the next innocent user is targeted by the hackers. If the users apply their computational thinking skills, they will eventually identify and eliminate all threat actors and their community will be safe. Learners will need reminding to apply their computational thinking skills. There are other learning connections that can be made in the game, including abstraction, decomposition, and algorithms. When the moderator is describing the activity, they should use terminology related to cybersecurity, such as coordinated, denial of service, traffic analysis, etc.

Extensions:

- You can introduce users with elevated roles and certain powers, such as ethical hackers (to whom moderators can secretly reveal the true identity of a single player during the night) and firewalls (which can secretly protect one player each night from an attack — either themselves or another player).
- Classes that are familiar with the game can propose changing some of the game mechanics and predict the outcomes. For example, you can increase the number of threat actors, or create additional roles. You could even ask learners to design, test, and evaluate an algorithm for an existing or a new player role. I have seen some students become so heavily invested in the game that they want to discuss their observations, theories, and hypotheses for many days after a game has stopped!

■ You can take students on virtual tours of places with computer science relevance, like a visit to the Alan Turing statue in Manchester, UK





OTHER SUGGESTIONS

Deconstruction: challenge students to create an annotated diagram of a piece of technology or system to explain the key components and the connections between them (for example, a desktop PC, a mobile device, a classroom or school network, or the World Wide Web). Refer to examples of diagrams such as the London Underground map.

Textbooks: see the 'Emergency rations' section on page 91 of Hello World issue 17. The guide suggests an approach to using textbooks that could be used in case of emergency.

CSunplugged.org: this free website contains a wealth of activities that don't require technology. Note, though, that you will need the internet to access them!

Quiz: some teachers write their own quiz resources; others keep a book handy with some general knowledge questions they can quickly access when required.

A virtual tour: during the lockdown periods, I created some virtual tours to locations with links to computing (helloworld.cc/exapditions). Each of these trips lasts 15 minutes, or longer if teachers use the questions included, and sites include Silicon Valley and the Alan Turing Memorial in Manchester. They do require an internet connection, video, and audio, so they may not be suitable for every tech failure.

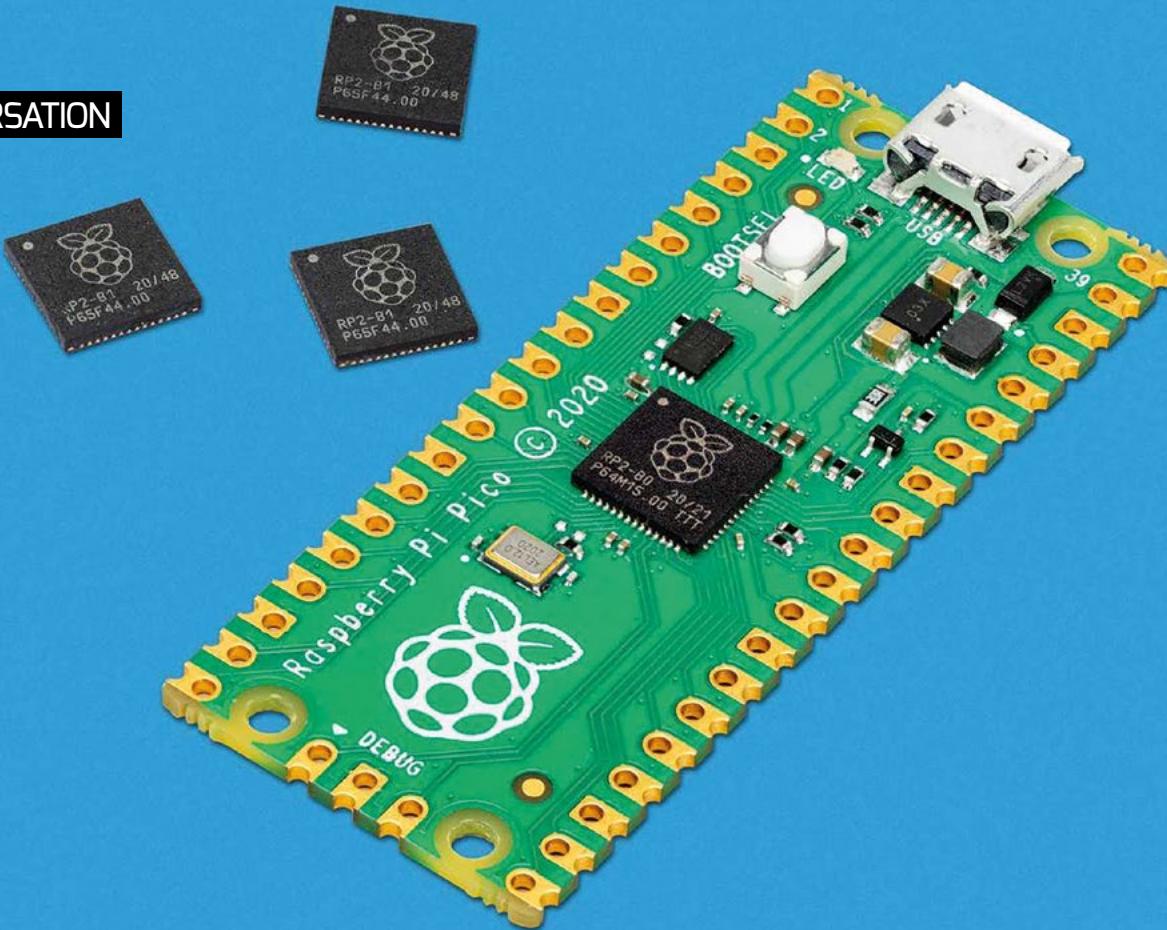
the first sign that things are about to go awry. It's likely that your favourite backup strategy will also become your learners' favourite — so watch out for children deliberately trying to sabotage the tech just so they can enjoy more of the same!

Some of the described activities have their origins in Victorian-style parlour games; I have modernised them and adapted them to help children develop computational thinking skills and apply computer science concepts to unplugged scenarios. All have been used successfully with children aged 5 to 18 years and above, some of which I'd prepared as a Plan B. [\(HW\)](#)



ALAN O'DONOHUE

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).



ZEROING IN ON ACCESSIBLE SOFTWARE

Laura Sach catches up with **Martin O'Hanlon** to find out what he learnt when creating the picozero libraries

Raspberry Pi Pico is a tiny and affordable microcontroller device that can be programmed using MicroPython. It costs only \$4 (about £3), so it's an ideal introductory device to physical computing for beginners. Martin has been working on a Python library called picozero (helloworld.cc/picozero) to simplify access to Raspberry Pi Pico's hardware and allow more people to use it for digital making projects involving physical components such as LEDs and buttons.

Laura: Why did you create picozero?

Martin: One aim of the Raspberry Pi Foundation is to democratise digital learning

and make it accessible. I have taught many workshops using the gpiozero library [an equivalent for the larger Raspberry Pi, helloworld.cc/gpiozero], and when you see someone light their first LED using a program they have written, it's a brilliant feeling. That level of accessibility was lacking for the Pico, and so picozero was born.

Smooth installation

Martin: I want to talk about the challenges we put in the way of others using our software. To make software accessible, it's important that as a developer, I understand the constraints a user has in being able to use it. Broadly speaking, the constraints a beginner may face may be environmental

or psychological — they might use a device where they don't have admin rights, or it might just be the good old-fashioned "Wow, that's hard; I don't have time for that."

Laura: Yes, I remember being frustrated as a teacher when I wanted to use the Python package Pygame and the instructions told me to "install it with pip". I had no idea what pip was, and I was too embarrassed to ask.

Martin: I think it's really important to have clear installation instructions, because this is where most people will turn off. If you create a barrier to getting your software installed that is too hard to overcome, no one will ever see the cool thing you created. I also

think that providing alternative installation options is important, in case someone can't use a particular method.

This was a particular challenge with picozero, as you have the complications of uploading code to a microcontroller. This

Picozero is designed to hide a lot of the complexity from the beginner — a light having a 'value' isn't a very intuitive concept. Instead, you tell the LED to turn on(), which is exactly how you would describe the instruction in English.

IF THERE'S AN ESTABLISHED WAY THAT PEOPLE ARE USED TO WORKING, WORK WITH IT AND DON'T REINVENT THE WHEEL

was made simpler by providing a version of picozero on pypi.org [the Python Package Index] and providing instructions for downloading directly from there.

Create a clear and consistent API

Laura: I'm sure there are plenty of people out there who aren't familiar with the term API — could you explain what this means?

Martin: An API [application programming interface] is a bunch of predefined things you can ask software to do. For example, with picozero, one method (a 'thing' you can ask the software to do) is to turn an LED on, but you don't need to know the ins and outs of how the software does that.

Without picozero, the command that turns on an LED is called 'value':

```
led = Pin(25, Pin.OUT)
led.value(1)
```

In picozero, the command is called 'on':

```
led = LED(25)
led.on()
```

Laura: How did you decide what names to use? Is it hard to figure out how to make things user-friendly?

Martin: I would definitely say that if there's an established way that people are used to working, then work with it and don't reinvent the wheel. How you name things can make a huge difference to the way in which your software is used and perceived. To make it accessible, you need to think about how your target users will interpret a name — give the method a name that **they** will understand, not one you do.

Laura: Ah yes. I occasionally had some very confused students who wrote a print statement in Python and expected a piece of paper to come out of the printer.

Martin: Exactly. There is a Python library called Tkinter that has a method named 'deiconify', which was so confusing that I even had a T-shirt made up with that written on it. It just means to display a minimised window, but the word is pretty meaningless to most people.

Ongoing support

Laura: So, you've got your beginner programmer over the initial hurdle and hidden some of the complexity from them. How do you keep looking after them following that initial rush of excitement?

Martin: Good documentation is the tool that you use to help people understand and use your product. Just as with the initial hurdle of installation, if the documentation isn't friendly, or doesn't answer the questions people have, they will give up. As well as good documentation, the Raspberry Pi Foundation also provides ideas for projects you can create using a Pico to help beginners build up confidence — we even test them on our staff!

You can try out Pico projects today at helloworld.cc/picoprojects, and discover the picozero beginner-friendly library for using common electronics components with Raspberry Pi Pico at helloworld.cc/picozero. (HW)



**MARTIN O'HANLON
AND LAURA SACH**

Martin and Laura are both senior learning managers at the Raspberry Pi Foundation. As a child, Martin wanted to be a scientist, an astronaut, or a snowboarder. Laura wanted to be Batman, but is probably better suited to being Batman's accountant.

OTHER ZERO LIBRARIES

Did you know there's a suite of Python libraries specifically designed for learners, called the 'zero' libraries? They're so called because one of their aims is to need zero boilerplate code to get going:

- **gpiozero** (helloworld.cc/gpiozero)
- **guizero** (helloworld.cc/guizero)
- **Pygame Zero** (helloworld.cc/pygamezero)
- **NetworkZero** (helloworld.cc/networkzero)



MY JOURNEY FROM DISCRETE COMPUTING TO TECH INTEGRATION

Emily Jones shares her experience of leading ICT and computing in both the UK and Thailand, and how her practice has almost come full circle

Towards the end of my teacher training, I attended my first job interview. I was extremely nervous, especially as I realised that one candidate was already on placement at the school and it looked like he was lined up for the job. However, the head teacher recalled me to his office as soon as the interview had finished. He had heard that I knew how to copy and paste an image from the internet and asked if I could show him how to do it. Back in 1999, it wasn't quite as simple as right-click/copy, and he seemed to think I was some kind of genius! It was this skill that gave me the edge over the placement candidate and led to me being appointed as ICT coordinator, a role that would remain with me, in its various forms, throughout my career.

My computing journey

Over the past 23 years, I have led on a range of initiatives, from the installation of the first networked computer suites, to the roll-out of England's 2014 computing curriculum. Prior to 2014, I worked as both a primary-class teacher and a subject coordinator (and, in many ways, as a technician too!). Much of my role involved advising other teachers on how to embed ICT across the curriculum via staff training, modelled lessons, and team teaching. However, with the introduction of the new curriculum, many staff started to shy away from the subject, as it became yet another thing to learn on top of all the other initiatives, such as teaching French and the ukulele (not together, but it felt at the time as if this might be next!). I therefore stopped being a class teacher and taught computing discretely as a specialist, while also travelling around my local area, advising other schools on how to implement the new curriculum.

During this time, I was fortunate enough to be appointed as the first female Computing at School master teacher in my region, and it was here that my passion for the subject grew. Fast-forward to the present day, and I am now working at Harrow International School Bangkok, alongside my husband,

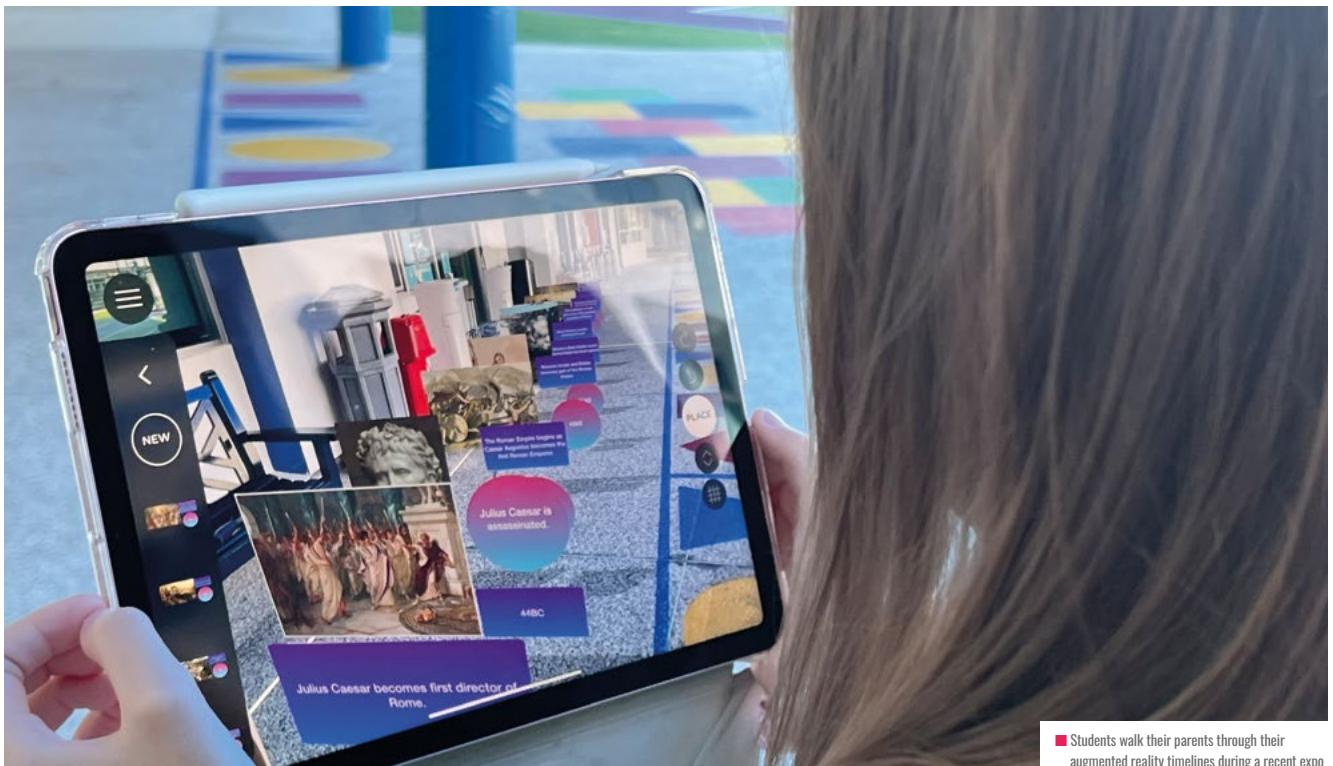
the aforementioned placement student! As head of lower-school digital technologies, I have a fantastic job and feel privileged to lead this ever-evolving subject in a school with amazing facilities, students, and resources.

Discrete versus integrated approaches

Historically, computing at Harrow has been taught discretely across the lower school, with students aged four to ten attending a one-hour lesson each week. Curriculum content has covered a mixture of computer science, digital literacy, and information technology. Although the discrete model appeared to work well, I always had an underlying concern that pupils might not be applying their learning across the wider curriculum. Feedback from staff reflected this, and the situation was further exacerbated by the deskilling of teachers, who weren't keeping up with technology changes as computing continued to be taught in a bubble.

A CLEAR VISION AND STAFF BUY-IN ARE KEY TO SUCCESSFUL INTEGRATION

In 2021, I was granted the opportunity to explore a model of greater tech integration. We are currently rewriting the entire lower-school curriculum, so this has provided the perfect opportunity for embedding computing objectives in a meaningful way. For example, Year 2 (aged 6–7)'s topic was 'Food for thought', which lent itself well to recipes, instructions, and algorithms, while over in Year 4 (aged 8–9), their enquiry question was 'How can we be responsible consumers?' which linked nicely to building and coding a sustainable house in CoSpaces.



■ Students walk their parents through their augmented reality timelines during a recent expo

I have also continued to teach computer science discretely, but instead of my lessons being used to release teachers for planning and preparation purposes, they are now an opportunity for staff to observe, team-teach, and upskill. Although most teachers are not yet ready to go it alone, many have commented on the applicability of computer science across other curriculum areas. For example, one colleague told me they will never teach angles in the same way again after witnessing how quickly students gain an understanding of this concept when learning via robots and sprites.

In terms of the digital literacy and IT strands of the curriculum, these are now integrated into other subjects. Each week, I attend year group planning meetings where we discuss ideas for tech integration. For example, Year 4 were recently researching the Romans and were planning to make a timeline. Rather than using traditional cutting and sticking methods, the pupils instead created timelines in Keynote and projected them in augmented reality via AR Makr. To teach students and teachers these skills, we booked a large auditorium, and I provided a demo. I also created how-to guides and made myself available so that staff could book me for the first lesson if they needed additional support. A highlight of this project was a parent expo where the students physically walked their family members through the timelines, while providing a narrative. It was such an effective way to embed their understanding of chronology and the passing of time.

Recommendations

If you are thinking of moving towards greater tech integration in your setting, there are several core elements that have been instrumental to the project's success: support from the school leadership; a clear vision; robust subject leadership; good resourcing

(1:1 availability of iPads has been key); an IT advocate in each year; protected time for regular training (all staff are currently working towards their Apple Teacher badges); a supportive environment in which to try out new techniques; and staff buy-in (which comes once all the previous elements are in place). It's also a good idea to appoint student leaders, who can support the community with tech integration, and can themselves become catalysts for change.

As I reflect upon my career so far, it's almost as if my job has come full circle. I'm back to advising on tech integration, while also training staff on the more subject-specific elements of the curriculum. It would have been easy to continue with the discrete model at Harrow, rolling out the same tried-and-tested curriculum week after week. But, in the words of the educator John Dewey, over a century ago, "If we teach today, as we taught yesterday, we rob our children of tomorrow." **(HW)**



EMILY JONES

Emily has worked as a curriculum lead since 1999 in both the UK and Thailand. She is currently head of lower-school digital technologies at Harrow International School Bangkok, where she is implementing a whole-school technology integration project (@EJ7416).

THE EUROPEAN ASTRO PI CHALLENGE: WHAT YOU NEED TO KNOW

The Astro Pi team catches up with primary teacher **Sophie Hudson** about her experience of taking part in the European Astro Pi Challenge

The European Astro Pi Challenge is back for 2022/23! Last year, 28,000 young people participated and had their messages and projects run aboard the International Space Station (ISS). We caught up with UK-based primary teacher Sophie Hudson about her experience of running the challenge in her school last year and the impact it had on her pupils.

What is the European Astro Pi Challenge?

The European Astro Pi Challenge gives young people the opportunity to conduct scientific investigations in space. Despite being a primary teacher for ten years, I found the idea of supporting pupils to take part in a coding challenge quite daunting. But I thought, "What's the harm in trying? If it goes wrong, it goes wrong." It just seemed like such a good opportunity, and the children got so excited when I told them about it.

There are two challenges to choose from: Mission Space Lab and Mission Zero. I carried out the Mission Zero challenge — the simpler of the two — for the first time last year. Last year's challenge guided children through the process of coding a personal message and creating an animation to display to astronauts on the Raspberry Pi computers on the ISS.

It aims to teach the basics of coding, so the children taking part (and their teacher) do not need any prior experience. Importantly for the classroom, you don't

need specialist equipment either, other than internet-connected computers.

How did you introduce the challenge to your learners?

The first step was talking about the actual challenge and what we were going to be doing. I showed pupils the Mission Zero website and we looked at the message I had coded, so they could see what the project would look like at the end. We then discussed the ISS and talked about what happens there.

When we started the task, we didn't just jump onto computers. We planned it all offline first, and then they copied over their code once they were at computers. The Astro Pi website really helped with planning, and the instructions were clear and easy to follow. I also printed off my finished project so that if pupils were getting frustrated, they could go through it line by line, comparing it to a finished one to find any errors. I was so impressed with the whole class — they were so careful with their coding.



Pupils get to learn completely new skills



An Astro Pi computer on the International Space Station

I took a mixed-ability group approach, with children working in teams of three at one computer. That worked well, as they could share out the roles depending on their strengths. For each group, we had one person acting as a coach (thinking about what they were going to type), one as a coder (doing the typing), and one as a debugger (checking the work).

It gave them responsibility, but also a safety net. It's very common and normal for errors to appear when coding, so it was nice that they had a team to lean on and they could problem-solve together.

The coding itself took an afternoon for my Year 3 and Year 4 classes (aged seven to nine), but we broadened the opportunity and started learning about pixels and the ISS — so to teach it thoroughly, it took us about a day. We loved it so much that we'll be taking part again this year, alongside another class in my school.

What were the benefits for your pupils?

It was super for them to learn about how coding looks in the real world — the fact that it wasn't just a computer game to play, and that their code actually went into space. Each team also received a personalised certificate listing their names, along with the date, time, and location that their experiment ran on the ISS, which they were all very excited about.

My class is made up of seven-, eight-, and nine-year-olds, so for a lot of these children, it was their first look at coding in Python, and it was great to see them learning totally new skills. It also piqued lots of interest in joining our school's Code

Club, which is brilliant. In particular, there was one pupil who was struggling with a lot of problems in their life. This project gave them something they were good at and something they were really into. That pupil is now one of our key members of Code Club, and they come every week. So there were some very clear, real benefits to taking part that went far beyond the challenge itself.

How did it tie in to the wider curriculum?

Last year, our Year 1 and 2 class (aged five to seven) was already doing a project on space, so it tied perfectly into that. For the older class, it was a great way to revisit some previously learnt knowledge about the subject.

In terms of the computing curriculum, they were all very much beginner coders.

- Do the challenge yourself in advance, so you know what you're letting yourself in for. You'll also need to register in advance, so you can do that at the same time.
- Print the code so the children can follow it line by line to iron out any mistakes.
- Assign individual roles to give each child responsibility and support.
- Try to give each student as much choice as possible over what they create.
- Emphasise the real-world impact the activity has — their work will actually be going into space!

All in all, incorporating the European Astro Pi Challenge into my lessons was a brilliant experience and I'm so glad I took that first step. I can't wait to see what my class comes up with this year!

The European Astro Pi Challenge is a project by the European Space Agency

WE LOVED IT SO MUCH THAT WE'LL BE TAKING PART AGAIN THIS YEAR, ALONGSIDE ANOTHER CLASS IN MY SCHOOL

They'd done a bit of work on Scratch previously, and since the project, we've further broadened our curriculum so that in Years 5 and 6 (aged nine to eleven) they do some work with micro:bits and HTML.

Taking part in the European Astro Pi Challenge boosted my confidence to think about incorporating other platforms into Code Club. We're now using the HTML and CSS Code Club projects, rather than just the Scratch projects. It made me see that they are very accessible, and that the children pick them up a lot more quickly than you would realise.

What advice would you give to teachers who want to try the challenge?

Although it might look daunting, running the challenge is actually very manageable, and there are clear and helpful instructions. For any teacher wanting to take part, my top tips would be:

in collaboration with the Raspberry Pi Foundation. It empowers young people, no matter their experience with computers, to write a simple computer program and share a personalised image with the astronauts orbiting above Earth. The challenge is open until 17 March 2023 (astro-pi.org). (HW)



SOPHIE HUDSON

Sophie is a teacher at Linton-on-Ouse Primary School and Nursery in York, UK.

WHY DID THE PROGRAMMER GET STUCK IN THE SHOWER?

Laura Sach shares some programming jokes
to add a touch of humour to your teaching

Bad jokes to make my students laugh (or groan) were always popular in class. Here are some jokes for you to inflict upon your students at timely moments in your teaching. I've broken down the jokes to relate to teaching topics, because jokes are only truly funny when they are explained, right?

"There are 10 types of people in the world: those who understand binary and those who don't."

I have to start with this one, because it's the ultimate computing teacher joke. It only works when written down, though, because the 10 in question is not actually 'ten' but 'one zero', which in binary represents the number two.

Q: Why did the parrot say "pieces of nine"?

A: He had a parrotty error.

The proverbial pirate's parrot usually says "pieces of eight", but in this case the number is incorrect. When data is transmitted across a network, a parity check might be performed on the bytes sent and received, to try to detect errors during transmission. For example, if a connection requires even parity, you would expect all bytes received to have an even number of 1s. This can be achieved by designating a 'parity bit', usually at the end of the data, to ensure that the number of 1s transmitted is even. For example, if we sent the number 8 — 00010001 — we would need to set the final parity bit to 1 to ensure an even number of 1s. If this byte was then received as 00010011, the receiver would know there was a problem because there was a parity error, in this case an odd number of 1s.

Q: Why do programmers get confused between Halloween and Christmas?

A: Because Oct 31 = Dec 25.

This is a number base joke. Dec refers to decimal (or denary), our most frequently used base-10 numbering system. Oct in this case refers to octal, which is base-8, where each place value is a power of 8:

Place value	8	1
Octal digit	3	1

We can convert the value 31 in octal to decimal by multiplying each digit by the place value and adding the results together: $8 \times 3 + 1 \times 1 = 25$. Thus, Oct 31 is Dec 25.

Q: Why did the programmer get stuck in the shower?

A: Because the shampoo bottle said 'lather, rinse, repeat'.

This one is a reference to an infinite loop: the instructions on the shampoo bottle don't explicitly tell you to stop washing your hair, but most humans don't follow instructions literally like computers!

!false

(It's funny because it's true.)

In many programming languages, the ! operator means 'not'. So '!false' means 'true'.

[“hip”, “hip”]

This is one of my favourites when tackling arrays with students. The joke is "hip hip array" — but if you're teaching with Python, you're bound to get one smart cookie who points out that in Python it's technically "hip hip list". (HW)

TECHY ONE-LINERS

Throw in these one-liners to guarantee a groan, or perhaps set as a starter activity to explain why they are funny:

- I love pressing the F5 key; it's refreshing.
- My friend's band is called 999MB. They haven't got a gig yet.
- I'd tell you an infinite loop joke, but you'd never hear the end of it.
- walks a packet UDP A bar into.



LAURA SACH

Laura is a senior learning manager at the Raspberry Pi Foundation.

THE BEBRAS PUZZLE PAGE

Each issue, **Chris Roffey** shares a computational thinking problem for your students based on the work produced by the International Bebras Community

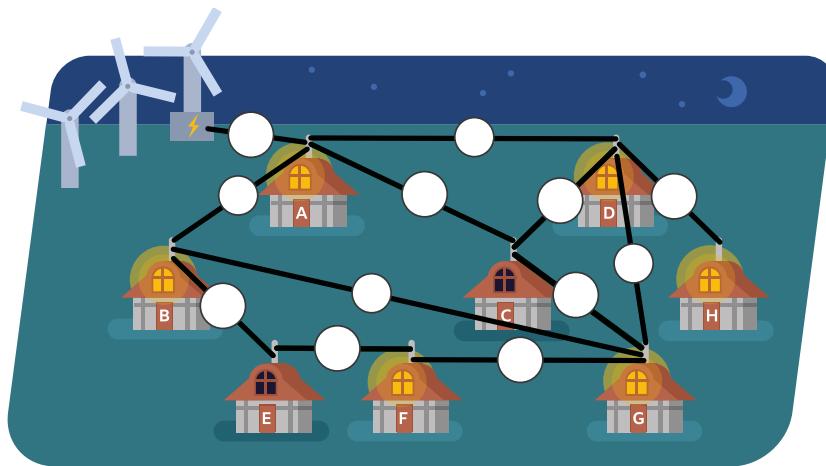
ABOUT BEBRAS

Bebras is organised in over 50 countries and aims to get students excited about computing and computational thinking. Last November, over 366,000 students took part in the annual Bebras Challenge. Our archived questions let you create your own auto-marking quizzes at any time during the year. To find out more and register your school, head to bebras.uk.



THE PROBLEM: ERROR DETECTION

Communication and networking



DOMAIN

Computer processes and hardware

SKILLS

Abstraction, decomposition, and evaluation

AGE

12–18 years

DIFFICULTY RATING

Ages 12–14 hard; ages 14–16 medium; ages 16–18 easy

Electricity is carried from wind turbines to houses through the network in the image above. Some links are faulty. The two houses with lights off don't have electricity any more, but the others do! Electricity can be carried from house to house in any direction. Label each connection with an X if it's faulty, a tick if it's working, or a question mark if it's impossible to tell. *This problem was originally written by the Hungarian Bebras Team. Solution on page 96.*

Further information

In computer networks, just as in electricity-distribution networks, links can be faulty — slow,

overloaded, or outright broken. Having redundancy in the structure of a network ensures its continuous availability in case of faults (if there are not too many at the same time).

Computer scientists often have to fix errors, not only in computer networks but also in software development. To fix an error, we have to identify its precise source, which is usually done gradually in several steps. Graph notation, which can represent large network structures, can help. Many algorithms have been developed by computer scientists and mathematicians to work with graphs to, for instance, determine a missing link in a given network structure.

COMPUTING KEYWORD SPOTLIGHT: ALGORITHM

Defining everyday words and phrases in computer science

Algorithms are at the heart of computer science. In the early years of education, children learn how to write simple algorithms, such as how to make a cup of tea. Later, they learn how to formally design algorithms using pseudocode and flow charts, and turn these into their own programs.

Although computer science is a very modern subject, algorithms are not. They are almost as old as mathematics itself. Beyond simple numeracy, nearly all mathematical methods are algorithmic in the sense that they follow precise rules to perform a calculation and produce an output. An early example from about 200 BC is the Sieve of Eratosthenes, which is still considered one of the most efficient ways of finding all the small prime numbers.

HELLO RUBY: EXPEDITION TO THE INTERNET

If you made the internet out of snow, what would it look like?

Jen McCulloch reviews the third book in the *Hello Ruby* series

INFO

BY Linda Liukas | PUBLISHER Feiwel & Friends | PRICE \$18.99 | ISBN 9781250195999 | URL helloworld.cc/expedition

Jen McCulloch

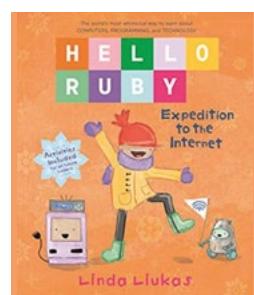
Ruby and Julia are best friends — they do everything together. When the snow falls, their imaginations run wild until they are interrupted by an incoming snowball from the hands of Julia's brother, Django. Reluctantly, the two friends allow Django to join them, and instead of making snow angels, the three children decide to make something much more exciting — a snow internet!

But what is the internet? Through the story's narrative, aimed at children aged four to eight, we explore key concepts surrounding computer networking, coupled with themes of friendship and teamwork. As the children build the component parts of the internet — servers, satellites,

and cables reaching as high as the sky and as deep as the ocean — the friends decide that, just like the internet, it is best when individual parts work together.

Expedition to the Internet is more than just a storybook, though. It also contains beautifully presented activities that build on the themes addressed in the story. The activities help make an otherwise dry topic fun and engaging. For example, one activity discusses the use of underwater cables as an important part of the internet. After using the World Wide Web to look at a map of underwater cables, children are given a fun coordinates challenge in which they send robots to fix broken cables on a map of the world.

The book is filled with other such games, printable and online activities, and internet scavenger hunts. These aim to consolidate the knowledge contained in each of the six chapters, which the author suggests can be worked through with a parent. The book also touches on those things that can be a little frightening online, opening up conversations between parent and child around their own use of the internet. Families could learn a lot and enjoy hours and hours of fun by working through the activity book together! (HW)



BEBRAS PUZZLE SOLUTION (PAGE 95)

The map in the image shows what we know about the links in the electricity-distribution network.

The first thing we know is that the two direct links to house E and the three direct links to house C are all faulty. As all the neighbouring houses have electricity, a working link would have brought electricity to houses C and E as well.

Next, links that are alone in providing electricity to houses where the lights are on cannot be faulty, otherwise no electricity could arrive there. This is the case for the link leading to house H and the link from house G to house F. The link from the turbines to house A must also be working, otherwise none of the houses in the network would have electricity.

The remaining houses, B, G, and D, are multiply connected to house A. For instance, B can get its electricity directly from A, but it could also get it from G if the link to

A was faulty. We can say the same about D. Finally, G can get its electricity either from B or from D. One of the links in the A – B – G – D – A cycle could thus be faulty, and these four houses would still get electricity.



UNCOMMON SENSE TEACHING

Is teaching a science or an art? This book attempts to swing the pendulum to explore the neuroscience of learning

INFO

BY Barbara Oakley, Beth Rogowsky, Terrence Sejnowski | PUBLISHER Perigee Books

PRICE \$20.00 | ISBN 9780593329733 | URL helloworld.cc/uncommonsense

Trisha Rhodes

As teachers, we often hear the question, 'Is teaching a science or an art?' The authors of this practical book believe we should approach teaching from the most recent scientific research, drawing upon their respective fields of neurobiology, education, and engineering.

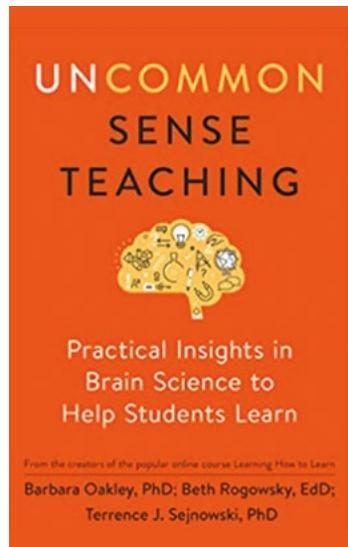
Uncommon Sense Teaching is split into ten chapters, ranging from collaborative learning to online teaching. Personally, I found the two active-learning chapters the most practical parts of the book. Here, the authors describe the difference between declarative and procedural memory. The declarative system is the learning system where students are conscious of what they're thinking about and learning. In contrast, the procedural learning

system is the basis of habitual actions. The book explores how good teaching draws upon both systems to enhance students' ability to remember their learning.

In my classroom, for example, I now use the wait-time strategy. This is where, if a student impulsively asks a question, I delay my immediate response until the end of the lesson or task. I coach students through their struggle to solve a problem by asking what strategy they used or what means they already attempted. Depending on the task, I might then use a Parson's Problem to introduce a new concept or skill that deliberately requires them to revisit their question and explain how they solved their problem. This usually ends with an exclamation: 'This is actually easy!' This section of the book has been a great help to me in guiding students through the process of learning how to answer their own questions through declarative strategies.

The biggest difference between this book and similar titles is the number of examples of best practice for teaching dyslexic learners. The book regularly explains how and why these students respond differently to learning new content, and gives plenty of practical examples of how teaching should be adapted to facilitate these different learning pathways.

This manual has encouraged me to reflect and analyse my teaching habits and style, along with how to maximise my students' learning. (HW)



ESSENTIAL READING

Three books that focus on the internet

THE FUTURE OF THE INTERNET AND HOW TO STOP IT

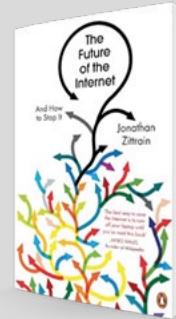
BY Jonathan Zittrain

PUBLISHER Penguin

PRICE Free/\$21.86

ISBN 9780141031590

URL helloworld.cc/thefuture



A freely available title that explores the dangers the internet faces if it fails to balance tightly controlled technologies with the flow of innovation that has generated so much progress in the technology field.

TRANSMISSION

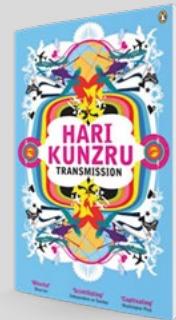
BY Hari Kunzru

PUBLISHER Penguin

PRICE \$14.90

ISBN 9780141008295

URL helloworld.cc/transmission



An entertaining novel following a programmer who sends out emails containing a malignant computer virus in a bid to keep his job, unintentionally causing worldwide chaos.

YOU ARE NOT A GADGET

BY Jaron Lanier

PUBLISHER Penguin

PRICE \$17.90

ISBN 9780141049113

URL helloworld.cc/gadget



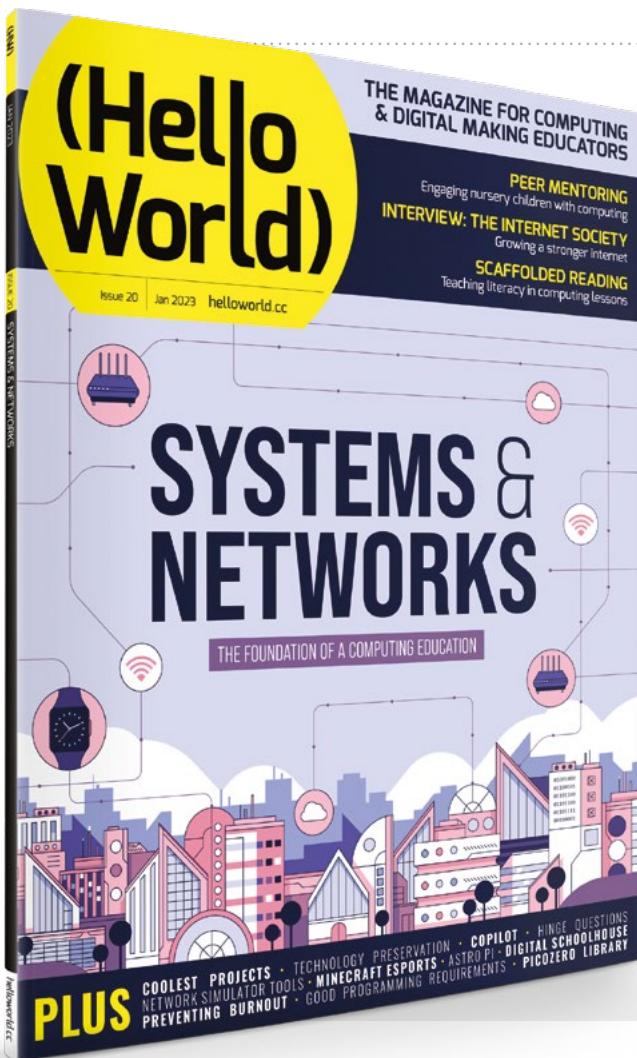
A powerful manifesto against 'mass mediocrity', critiquing the modern internet and how the author believes it is eroding creativity.

“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 and accompanying podcast 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.



Q WHO MAKES HELLO WORLD?

A The magazine and the accompanying podcast are produced by the Raspberry Pi Foundation.

Q WHY DO 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 podcast at helloworld.cc/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 will 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

FIND US ONLINE



@HelloWorld_Edu



[Want to talk? You can reach us at:
\[contact@helloworld.cc\]\(mailto:contact@helloworld.cc\)](https://www.facebook.com>HelloWorldEduMag</p></div><div data-bbox=)

www.helloworld.cc



(Hello
World)

helloworld.cc