THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS

INTERACTIVE FICTION

Combining computer science & literacy

COOLEST PROJECTS

The world's leading technology exhibition for young people

SILICON VALLEY

An itinerary for the ultimate computing school trip

TEACH COMPUTING

With the National Centre for Computing Education

NO HEADACHES
HOMEWORK

A strategy that helps students and eases workload

PLUS SCRATCHMATHS • COGNITIVE LOAD THEORY IN ACTION • CARE WORKERS TEACH CODING • PARSON'S PUZZLES
RASPBERRY PI ARCADE GAME • GETTING STARTED WITH THE CRUMBLE • DIGITAL MAKING AT LIBRARIES & MUSEUMS

{ code club }

# START A CODE CLUB IN YOUR SCHOOL!

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

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

- ✅ **Code Club is free**

- ✅ **Code Club offers engaging, hands-on activities that have been designed to spark curiosity and inspire creative thinking**

- ✅ **Our free online training is perfect for teachers who are new to coding or are starting a Code Club for the first time**

Join the network of over 7000 clubs across the UK teaching more than 100,000 young people to code!

Get involved at codeclub.org

HELLO, WORLD!

What an incredible privilege it is to join the team at *Hello World*! From my short time here, it's already clear to me that the magazine is both a useful resource for educators and a core centre of the computing education community. *Hello World* helps readers feel connected to like-minded individuals who strive to do the best for their students.

I'm really happy to say that Miles Berry is still very much a key figure in *Hello World*, both in his role as columnist (turn to page 97 for his revealing insights into the evolution of computational thinking) as well as his role as Contributing Editor, where his experience and expertise in computing education are invaluable.

This issue, we're really excited to bring you more news of the National Centre for Computing Education (pages 14–27). This unprecedented and world-leading effort to support computing teachers is still in its infancy. However, there are already numerous brilliant resources and opportunities for development available through the initiative. Many of these are freely accessible online from anywhere in the world.

My primary aim as Editor of *Hello World* is to build the value of the magazine even further. It's a really exciting time to be joining the team as we increase our publication frequency from three issues a year to five. This means we can share even more incredible resources and spark a more active dialogue between educators. And to do this we need your help: get in touch with feedback, questions, and ideas for resources and features at **contact@helloworld.cc** or via Twitter **@HelloWorld_Edu**. We look forward to hearing from you!

Sian Williams Page
Editor

FEATURED THIS ISSUE

SIMON PEYTON JONES
PRINCIPAL RESEARCHER, MICROSOFT RESEARCH

Simon chairs Computing at School and the National Centre for Computing Education. Together with Sue Sentance, he explains what the NCCE means for computing education on page 14.

CLAIRE WICHER
NEWLY QUALIFIED TEACHER

Claire is the founder of CodeUp UK and a Raspberry Pi Certified Educator, Pi-Top Champion and National Coding Week Ambassador. Turn to page 78 for her Escape the Classroom resource.

SABER KHAN
MIDDLE AND HIGH SCHOOL TEACHER

Saber teaches in Brooklyn, New York. On page 44 he introduces p5.js, a visual arts library from the Processing Foundation, where he works as Education Community Director.

(HW)

(HW) CONTENTS

COVER FEATURE

TEACH COMPUTING
With the National Centre for Computing Education

14

PRIMARY SUPPORT
How the NCCE will help primary teachers

20

ISAAC COMPUTER SCIENCE
Resources and events to inspire at A-level

16

EXCELLENCE IN ALL SCHOOLS
The NCCE's focus on increasing diversity

26

NEWS, FEATURES, AND OPINION

ARTEFACTS TO INSPIRE
Girls get making at Leeds libraries and museums

42

SILICON VALLEY
The trip of a lifetime

62

RETROPIE ARCADE GAME
One student's incredible project

64

■ The Mad About Litter team explaining their award-winning app

COOLEST PROJECTS 2019

Young digital makers showcase awe-inspiring creativity

Faye Magellan

The first thing you noticed at the recent Coolest Projects UK event in Manchester, was the electric atmosphere. The excitement of the young people, their families, and the community was contagious. At this world-leading tech showcase for young people, organised by the Raspberry Pi Foundation, young innovators shared their digital projects with one another and with the public.

The projects were mind-blowing: take for example the project that Matt, 12, showcased. He created the Adrenaline Pen Smart Case for schools, which displays life-saving audio-visual instructions whenever the pen is removed. Or the Scratch game, I'm Doing My Homework Really, which a team of 8- to 11-year-olds from Eccles developed, and which changes the player's screen from a game to homework when they hear their parents coming.

Some young people had spent a year developing their projects, and others just a few weeks or days, but the calibre of all their creations was incredible, and the judges had a truly tough task choosing the best from among more than 80 projects.

Having travelled all the way from Fife in Scotland, team of five, Jack, Elsa, Bethan, Tara, and Lily won the Mobile Apps award, which was sponsored by Atom Bank, for their Mad About Litter app which helps tackle littering in their local area.

The winner in the Hardware category was 13-year-old Freddie from Monmouthshire, who made Door Pi Plus – a door security system for the elderly, based on facial recognition software. Freddie said: "I enjoyed meeting other people, and seeing their amazing projects. I started coding at my primary school Code Club, but now I mainly code at home. Coding is cool because you can invent cool things to help you and other

people around you. I do think more kids should code because lots of the jobs in the future are probably going to involved coding."

Young people's creativity knew no bounds at Coolest Projects. Whether their creation was a project benefiting someone in their life or in society, or a fun game, or piece of hardware, the participants were passionate about telling the judges, visitors, and fellow participants the story behind their inventions.

Parent Ann-Marie Caporn, a computer science teacher from Manchester, said participating in the event helped her daughter gain confidence and a greater love for coding.

"My daughter can be quite shy so, even over the course of the day, it's boosted her confidence. She's an expert in her project, and explaining it to judges, other kids, and the public, she's been delighted. I've seen her grow in confidence over the day!"

COOLEST PROJECTS INTERNATIONAL
TICKETS AVAILABLE NOW!

COOLEST PROJECTS 2019

By Rosa Langhammer, General Manager of Coolest Projects and CoderDojo

Each time I walk into a Coolest Projects event, I never fail to be bowled over by the creativity and confidence of the young people presenting their projects. But, to me, of all the Coolest Projects events, Coolest Projects International is something truly special.

Coolest Projects International takes place this year in Dublin, Ireland on 5 May 2019. Over 10,000 people are expected to participate in or visit the event on the day. Coolest Projects International is special because young people from around the globe come together, showcase ideas, but, most importantly, share a common passion for technology and make new friends who they share a hobby with.

Why attend?

Coolest Projects is a great family fun day out with lots of interactive, hands-on activities from the maker community and our sponsors including robotics, VR, drones, and much more. You can try to get out from our digital-themed escape room, or explore the history of video games throughout the ages. Most importantly, you can be inspired by the projects that young people have created in a wide range of all technologies.

Tickets for Coolest Projects International are free but limited. Get them today at **coolestprojects.org**.

■ Coolest Projects gives young people the opportunity to share their amazing creations

Coolest Projects USA

The celebration of this generation of young creators continued in the US on Saturday 23 March at the Discovery Cube Orange County in Santa Ana, California. Creators aged 5–18 came together from all over the US, traveling from as far as Georgia, Louisiana, Texas, North Carolina, and Pennsylvania to showcase their incredible ideas, brought to life by them through the power of coding.

The judges learned a lot from Lara, the winner in the Websites category, who created a website named Another Purpose for Eggshells. Lara is passionate about birds, and built her site to teach people about the problems that ducks face. She conducted research to find out whether or not eggshells can be used as a source of calcium in bird food, in order to prevent birds from being fed carbohydrates, which can cause tumours.

Other incredible projects included the Smart Community Sprinkler System by Adarsh, who won the Hardware category. He was inspired by the need he saw in communities in California to manage water during the drought. Using a Raspberry Pi, he built a moisture-sensor-based smart sprinkler system that integrates real-time weather forecast data and Twitter feeds to dispense only optimum amounts of water in compliance with city water regulations – fantastic!

A community of makers

Maddie Moate, presenter of the CBeebies series *Do You Know?*, and one of the Coolest Projects UK hosts, said: "Whilst we were here to announce winners, it very much felt like a day celebrating a community of makers, builders, and change-makers, and that's what today is really about. The careers these creators will be doing in the future probably don't exist yet, and giving them the opportunity to take part in showcases like this lets them take control of their own futures and build their own pathways." **(HW)**

■ Young creators from all over the US came together for Coolest Projects USA

CELEBRATING DIVERSITY
IN COMPUTING

A set of free posters is available for you to use in your classroom

Jane Waite & Lowri Jones

A new set of posters from Queen Mary University of London, the Institute of Coding, and cs4fn has been released to celebrate diversity in computer science. They show the faces of key figures in historical and contemporary computing: dozens of people of different genders, ethnicities, and backgrounds. The team are not finished with the project yet, so would love to hear who you would like to include.
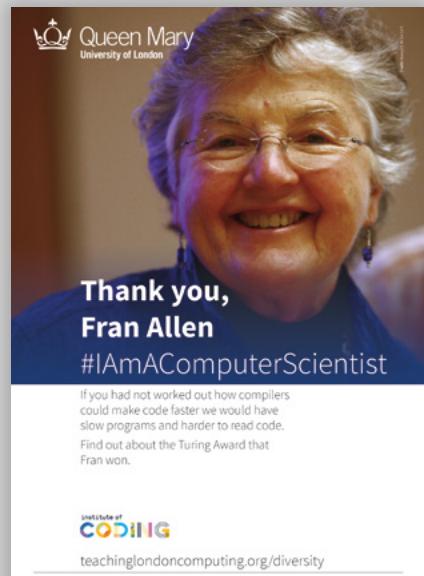
The posters form part of a wider set of resources from cs4fn – an organisation that promotes computing as a fun, fascinating subject that is accessible to everyone, regardless of their background, gender, ethnicity, age, or current attainment.

Indeed, anyone can be a great computer scientist. However, there is plenty of evidence to suggest that computing in the UK largely remains a niche subject for boys. Analysis of 2017 exam results published by Roehampton University, for example, revealed that 30,000 fewer girls took a computing KS4 qualification than in 2014; only 20% of entrants were girls at GCSE. This fell to 10% at A-level. And the Royal Society's *After the Reboot* report found that black pupils were significantly less likely to take computing at GCSE than white or Asian pupils.

This trend of low female and BAME (black, Asian and minority ethnic) participation in tech-related education is mirrored in industry, where the British Computing Society (BCS) reported that only 17% of IT specialists were female. IT specialists are also under-represented, with respect to disabled people and older people, compared with the wider UK workforce. And a report last year, from Inclusive Boards, found that only 8.5% of senior leaders in the tech sector are from BAME backgrounds.

So the picture is rather bleak: our digital world is not created, nor cared for, by a diverse workforce, and figures from education suggest the picture does not seem to be getting better. There is therefore work to do to reach out to people and show that technology is a meaningful and rewarding area to be in.

Thank you,
Juliana Rotich
#IAmAComputerScientist

You co-developed a platform that saves lives and supports democracy across the world.

Find out who provides the data for Juliana's organisation, *Ushahidi*.

teachinglondoncomputing.org/diversity

Thank you,
Safiya Umoja Noble
#IAmADigitalResearcher

Your work on ethics and the design of digital media platforms reminds us that search engines can reinforce racism and that algorithms can have gender bias.

Find out about data science and ethics.

teachinglondoncomputing.org/diversity

Thank you,
Fran Allen
#IAmAComputerScientist

If you had not worked out how compilers could make code faster we would have slow programs and harder to read code.

Find out about the Turing Award that Fran won.

teachinglondoncomputing.org/diversity

The Government supports this view, announcing in 2018 a £40 million nationwide initiative – the Institute of Coding (IOC) – to promote the next generation of digital specialists, and boost equality and diversity in technology-related education and careers. The posters have been produced and shared as part of this work, with the hope of encouraging students from all backgrounds to consider careers in computing and technology.

The posters were piloted at a number of schools. One piece of feedback was to add a question or activity to the posters to encourage students to find out more about each person. Now there are longer profiles for each person on the project's website.

Ideas for activities

There are numerous ways the posters could be used in cross-curricular activities, incorporating history, geography, and PSHE (Personal, Social, Health and Economic) education. Safiya Umoja Noble could be introduced to talk about how groups within society have had their voices silenced or amplified at different times

in history. Safiya worked in marketing, and realised that large corporations were using search engines to advertise brands in ways that people were not aware of. Now working in academia, she warns against the risks of not understanding how technology works and not keeping policy in step with advances in computer science. The work of Hedy Lamarr could be

available to download for this purpose. If you use the posters, get in touch via **teachinglondoncomputing.org** and **contact@helloworld.cc**. We would love to know if the posters have sparked discussions with your students, and we can feature your stories on the **teachinglondoncommputing.org** blog, and in *Hello World!* (HW)

> ## ONLY 8.5% OF SENIOR LEADERS IN THE TECH SECTOR ARE FROM BAME (BLACK, ASIAN, AND MINORITY ETHNIC) BACKGROUNDS

featured in a World War II history lesson. A film star and inventor, Hedy co-designed frequency hopping, a technique to protect secret radio signals from being jammed. This could tie in with work on Alan Turing, code-breaking, and how computer science, communication, and cryptography has changed the course of history.

How to get involved

The posters are free to download from Teaching London Computing (**teachinglondoncomputing.org**). They are available in PDF or adaptable PowerPoint format and would make a brilliant display in a classroom or computing lab. The PowerPoint can also be used as a rolling presentation on a screen, with a special 16:9 format

Thank you, Segun Fatumo #IAmAComputerScientist

You are saving lives by using your computer science expertise in genetic research. Find out about bioinformatics.

teachinglondoncomputing.org/diversity

Thank you, Shafi Goldwasser #IAmAComputerScientist

Without your work on algorithms and maths used in encryption we would not be as safe online. Congratulations on the Turing Award. Find out about the ball method of explaining zero knowledge proofs.

teachinglondoncomputing.org/diversity

Thank you, Jeannette Wing #IAmAComputerScientist

You put Computational Thinking on the map. How abstract is that? Find out about decomposition and algorithmic thinking.

teachinglondoncomputing.org/diversity

Sonya Teich is a visual effects artist featured in the campaign. She has a degree in Engineering and Computer Science

THIS IS ENGINEERING

How a digital campaign challenges misconceptions of the industry

Sian Williams Page

A campaign to change teenagers' outdated views of engineering has seen enormous success in its first year. This Is Engineering aims to challenge the misconceptions of what engineering is, with the goal of encouraging more young people to pursue careers in the industry.

The digital campaign features a series of short videos, each profiling a young engineer. Since its launch in January 2018, the campaign's videos have been viewed over 28 million times via social media channels. Analysis from the Royal Academy of Engineering found that young people who had watched the campaign's videos were 85% more likely to consider a career in the industry.

The campaign was devised in response to research which revealed an annual shortage of 59,000 engineers in the UK, as well a lack of diversity in the field. Black, Asian, and minority ethnic people make up less than 8% of the engineering workforce, and female engineers make up just 12%.
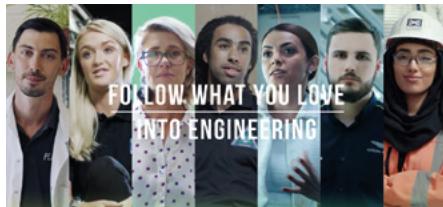
Engineering is critical to address the global challenges that impact sustainability, health, security, and quality of life. Jo Trigg, Associate Director at the Royal Academy of Engineering, explained, "This persistent

problem shortage of people and shortage of people from diverse backgrounds, encodes biases in engineering's approaches and affects its ability to address the challenges that it needs to address."

Preliminary research from the Royal Academy of Engineering found that teenagers have a narrow perception of engineering: they associate the word engineer with work to fix railway lines, boilers, cars, and photocopiers. The lack of relatable role models is compounded by the fact that online searches for the word 'engineer' predominantly return images of white men wearing hard hats.

One of the young engineers featured in the campaign is Sonya Teich who, after completing her degree in Engineering and Computer Science, started working as an Animation and Visual Effects artist at Disney. She is now Lead Effects Technical Director at Framestore in London, managing a team of visual effects artists, and has worked on films such as *Beauty and the Beast, Guardians of the Galaxy,* and *Avatar.*

Trigg explained that, "We found that kids didn't think there was a space in which they could be both creative and technical. Quite early on in the education system,

The campaign features young engineers from a wide range of backgrounds

children are forced to make choices about their subjects and they tend to see that they have either a STEM route or an arts route, and not a middle ground.

"We wanted to make sure that we could create that space, inhabit it, and communicate the true variety and breadth of engineering. We talk to teenagers about tech, sport, fashion, music. We show them that engineering is behind those things, bringing them in via a conversation they're already having."

Now that the campaign is in its second year, the team at the Royal Academy of Engineering are hoping to encourage parents and teachers to talk about engineering in positive ways with young people. More details of the campaign, including the videos, are available at **thisisengineering.org.uk**. (HW)

■ A STEM artefact on display at the launch event of the new MA at King's

EARN A MASTER'S
IN STEM EDUCATION

King's College London launches a new MA in STEM Education, with scholarships for teachers in English state schools

Sian Williams Page

King's College London has launched a new MA in STEM Education, which will run for the first time from September of this year. The course is aimed at education policy makers and educators – both in formal and informal settings – from science, mathematics, engineering, computing, and geography backgrounds. The programme is expected to broaden participants' understanding across the entirety of STEM, and enable students to engage with research and policy in a range of global contexts.

The technology company Wipro is sponsoring 15 scholarships. These are available to teachers working in English state-funded schools and cover 70% of the fees. A number of scholarships will be ring-fenced for teachers working in social mobility 'cold spots', such as Cornwall and Liverpool.

The course has been designed with a flexible structure, with online sessions to encourage applications from educators around the country. Teachers who have recently completed a PGCE will be able to transfer credits to count towards a third of the qualification, meaning that teachers undertaking the MA on a part-time basis will not need to reduce their hours at school.

The course will bring together educators and researchers in STEM education, and will have a focus on widening participation in these subjects. Lulu Healy, Professor of Mathematics Education at King's and an academic involved in running the course, explained that this was a big motivation for her own career path: "One of the reasons I was attracted to mathematics education was the invention of the microcomputer,

and the ways in which computer programming and digital resources might make it possible to create a new way of doing and experiencing mathematics.

"I got involved in computer programming after working with people with disabilities and thinking about how, by using digital technology, you can open up opportunities to do things in very different ways, like doing maths through sound or touch."

Students will be able to explore creative approaches to STEM education in a hands-on manner thanks to a new makerspace at King's. There's also an optional internship unit, where students will have the opportunity to work in STEM organisations.

More information on the course, and details of how to apply, can be found at **helloworld.cc/kcl**. (HW)

■ Beverly Clarke seeks to inspire heads and aspiring heads of computer science departments on the course

A COURSE FOR NEW AND ASPIRING
HEADS OF DEPARTMENT

The Creative Computing for Key Stage 3 course showcases
new ideas for inspiring pupils in computing

Simon Humphreys & Claire Penketh

Computing at School (CAS) has been working with one of its lead partners, Microsoft UK, to provide a two-day course for new and aspiring heads of computing in UK secondary schools.

Day one of the course is called 'Lighting the Spark'. Participants examine the use of use of creative and innovative teaching techniques and creative resources to bring the subject of computing alive.

Participants then have the chance to apply what they have learned before returning for the second day. On day two, called 'Fanning the Flame', participants have the chance to look more closely at pedagogical techniques, experience a lesson with the micro:bit, and hear ideas on raising the profile of their departments.

Beverly Clarke, one of the trainers on the course, explained how the content is designed to meet the needs of experienced teachers, as well as those who are relatively new to the subject. "What I try to do is find the gaps in their knowledge and sometimes it's a case of adjusting what material I have on the spot, but it's always a joy to see the teachers grow in confidence."

As can happen in any classroom, Beverly can sometimes find herself mulling over a tricky question posed by course participants. "I was asked how to make cybersecurity more interesting, less dry and I suggested a couple of unplugged activities to teach encryption, and I came up with the idea of using Morse code and semaphores."

She said, "I thought of interesting ways that they could teach what a code is, and how it can be used in a practical way and this seemed to be the answer. I even suggested how to make the semaphore flags using coloured paper and straws!"

Trainer Tig Williams devised the course and also teaches on it. Tig said, "I've had participants come up to me afterwards and say 'that was awesome!' They have been beyond effusive in their comments."

One participant who attended the course in Nottingham said it was a "great opportunity to think about what you're doing and why you're doing it and how you're going to do it differently when you get back into school." Another added: "I'm already teaching Key Stage 3 and plan on making big changes to scheme of work, based on what I've learned on the course today".

Microsoft UK has provided funding for 100 teachers to attend this course, with priority given to teachers in areas of low social mobility. More dates will be held soon. Contact **compatsch@bcs.uk** to register your interest. **(HW)**

NEW ONLINE COURSES FREELY AVAILABLE
TO EDUCATORS WORLDWIDE

Develop your understanding of computer science and guide others to learn

Dan Fisher

The Raspberry Pi Foundation has recently launched a series of free courses that will give educators a grounding in the concepts and practical applications of computing in all its forms. The courses are aimed at teachers of GCSE Computer Science (ages 14-16), but are available to anyone with an interest in the subject.

Here are details of three courses you can sign up for now. They each start from 1 July of this year.

Each course takes three weeks to complete, with approximately two hours of study per week. The courses can be taken wherever and whenever is convenient. They include peer-led discussions, trainer interactions, and feedback on knowledge as the courses progress. Teachers from England taking these courses can earn credits towards certification with the National Centre for Computing Education.

An Introduction to Computing Networking for Teachers explores how computers connect and communicate with each other. Participants will learn how to create connections between computers, by exploring the fundamentals of computer networking. These include: networks and architecture; transmission of data and network protocols; secure transmission of data; the internet, including routing, DNS and the 'World Wide Web'. Educators will complete the course with enough confidence to teach the subject to their students.

Understanding Maths and Logic in Computer Science explores some of the more mathematical aspects of computer science – topics that can be challenging to understand and to teach. The course is inspired by PRIMM methodology. It builds on an understanding of a range of topics including programming, equations and logic. It also covers the concepts of logical operators, logic gates, truth tables, and control structures, as well as guiding educators on how to develop an engaging escape room activity.

Understanding Computer Systems covers how the components of a computer work together – from the software to the hardware, from the RAM to the CPU. Participants will learn about what a computer operating system does and why it is needed, and will also address the factors that affect computer performance.

More information on these, and other courses, is available at the following link: **helloworld.cc/courses**. For more information on certification, via the National Centre for Computing Education, visit **helloworld.cc/certification**. **(HW)**

TEACH COMPUTING WITH
THE NATIONAL CENTRE
FOR COMPUTING EDUCATION

What does the National Centre for Computing Education mean for the future of computing teaching? **Sue Sentance** and **Simon Peyton Jones** provide an overview of the new opportunities from this unprecedented investment in the subject

In the last issue of *Hello World*, we introduced the new National Centre for Computing Education (NCCE). In the next few pages you will find much more detail about the many different aspects of the NCCE – the Hubs, certification, Computing at School (CAS) Communities of Practice, the Computer Science Accelerator programme, and more. It's a very busy time for those involved and, although it's still early days, there will be much for teachers in England to benefit from. Here we wanted to step back and look at why this initiative is so important and what it can mean for computing teaching.

Computing as a school subject

The English National Curriculum for computing frames a big vision for computing as a school subject. On the one hand, it is associated with the technology we use day to day. This is constantly changing and affects our lives and society in a myriad of ways. We need to know how to interact with technology and to be able to negotiate the ethical and societal implications of the ever-increasing power of computers. On the other hand, it's a discipline, with fundamental principles that haven't changed since we first started building computers. Like maths and

natural science, all children should have an elementary understanding of computer science, so that they can understand and make informed choices in the complex world that surrounds them.

You can see computing as aligned to mathematics, science, engineering, or all three – computing is the quintessential STEM subject. (Matti Tedre has written a great book on this topic, called *The Science of Computing: Shaping a Discipline*.)

Studying computing enables us to gain insight into computational systems of all kinds, whether or not they include

computers. Computational thinking and computing skills can be used in many other domains, including biology, chemistry, linguistics, psychology, economics, and statistics. Learning computing allows us to solve problems, design systems, and understand the power and limits of human and machine intelligence, and being able to think computationally gives us the ability to conceptualise and understand computer-based technology. It's a hugely exciting subject and, in our opinion, essential for everybody!

What the NCCE will do?

Our challenge as teachers is to make that big vision come to life in our classrooms, and to inspire our young people with a real sense of excitement and possibility. A deep subject discipline, that you can creatively bring to life through writing

SUE SENTANCE
Sue is Chief Learning Officer at the Raspberry Pi Foundation. An ex-teacher, teacher-trainer, and academic, she is passionate about research in computing education.

SIMON PEYTON JONES
Simon is a Principal Researcher at Microsoft Research and an Honorary Professor of the Computing Science Department at Glasgow University. He chairs Computing at School and the National Centre for Computing Education.

To recap from the last issue, the NCCE is run by a consortium consisting of BCS, STEM Learning, and the Raspberry Pi Foundation, together with many others working

> ## " LEARNING COMPUTING ALLOWS US TO SOLVE PROBLEMS, DESIGN SYSTEMS, AND UNDERSTAND THE POWER AND LIMITS OF HUMAN AND MACHINE INTELLIGENCE "

programs, and that leads directly to a huge variety of meaningful and well-paid jobs – what's not to like?

But computing, in this broad sense, is relatively young as a school subject, meaning that most of us did not study it at school, or at least not the wide span of principles and practice that are included in the curriculum in England. Therefore, having access to a range of professional development opportunities is absolutely key, as highlighted by The Royal Society in the recent 'After the Reboot' report (**helloworld.cc/reboot**). Many teachers have benefited by their involvement with the CAS Network of Excellence in previous years; the new funding means that we can scale up that work and offer many more professional development opportunities through the NCCE and associated programmes.

alongside us. It has been tasked to deliver three pieces of work:

- A National Centre for Computing Education, which will establish a network of 40 school-based Computing Hubs to provide continuing professional development (CPD) and comprehensive resources for computing teachers in primary and secondary schools and colleges. The NCCE will also facilitate strong links with industry.
- A teacher training programme to upskill existing teachers to teach GCSE Computer Science. This is called the Computer Science Accelerator programme.
- A programme to support A-level Computer Science students and teachers with high-quality resources and CPD. This is called Isaac Computer Science, a sister project to Isaac Physics.

The NCCE will succeed only if it does things *with* teachers, not to teachers. We know from research that professional development is most effective when it's collaborative and sustained. So the 'Centre' is virtual; it is distributed around the country in the school-based Hubs and CAS Communities of Practice. By being decentralised and school led, we hope to ensure a grounding in classroom practice.

Bursaries are available to support teachers, and our mission is to particularly support schools in priority areas, so that all children in England have an opportunity to receive a world-class computing education.

The level of investment for computing education that has been provided for England is unprecedented anywhere in the world, and offers great opportunities for schools and teachers. Moreover, the resources (including online courses, curricula, lesson plans, schemes of work) will be available for free, forever, to anyone in the world.

The articles on the next few pages will enable you to dig into the detail of the work planned by the NCCE over the next few years, and we welcome your feedback and engagement with what we are doing. Our goal is for the NCCE to be at the heart of the computing teaching community and enable us all to work together to move our subject forward and give young people amazing learning opportunities and a great future! (HW)

INSPIRING THE NEXT GENERATION OF COMPUTER SCIENTISTS

LAURA SACH
Laura (@codeboom) is a former Head of Department and has many years experience teaching Computer Science. She now leads the Isaac Computer Science A level content team.

Laura Sach explains how Isaac Computer Science can help your A-level class

As a former teacher, I know all too well that planning A-level Computer Science lessons can be hard and time-consuming. In recent years, changes to exam specifications have pushed content which might have previously been covered in the first year of undergraduate study into A-level, stretching the capabilities and time resources of even the most confident teachers. While textbooks can provide a good foundation for learning the content, extra exam-style questions are difficult to come by, and worked examples are like gold dust. I have spent many hours inventing new problems, checking my working, and making sure I am confident at explaining the possible misconceptions students may encounter on a particular topic. All of this ate into my own time in the evenings.

As part of the National Centre for Computing Education provision, the Raspberry Pi Foundation and the University of Cambridge are collaborating to produce a new platform called Isaac Computer Science. I am proud to lead the team creating the content, offering much-needed support for A-level Computer Science students and teachers. The project follows in the footsteps of Isaac Physics, a highly successful web-based platform for A-level Physics students which provides a wide range of questions with specific feedback. Schools actively engaging with the platform have increased 40% of their cohort's A-level grades from C to B. I hope that Isaac Computer Science will help busy teachers to boost their classes' attainment, while allowing them to spend less time writing new questions and devising worked examples.

Here in the Isaac Computer Science team, we are particularly excited by the variety of ways students will be able to answer questions to test their learning. Thanks to the expertise of the University of Cambridge technical team, answering a question might require the student to create an equation with Boolean logic using the equation editor, manipulate statements as part of a Parson's problem, or even type short answers in natural language. Every question will offer feedback to address common misconceptions, and most will be accompanied by a video hint showing a partially worked solution. Some topics will be available in May, with full coverage of the UK's AQA and OCR specifications available from November of this year. Educators from any country will be able to sign up to Isaac Computer Science and use the resources for free. You can register your interest now at **rpf.io/alevel**. (HW)

Teachers attending a training session on object-oriented programming at the Raspberry Pi Foundation offices in Cambridge. By Dan Fisher

EVENTS TO **INSPIRE** AND **ENHANCE** STUDENT PROGRESS AT A-LEVEL

FERGUS KIRKPATRICK

Fergus leads the A-level event programme for Isaac Computer Science. Before joining the Raspberry Pi Foundation, he worked at the University of Cambridge on part-time master's courses and e-learning.

Fergus Kirkpatrick introduces the collaborations between Isaac Computer Science and English universities

A key facet of Isaac Computer Science is a programme of events we are organising to inspire students to take Computer Science at degree level. We will be offering a range of events including teacher CPD, student classes focusing on particular aspects of the A-level curriculum, and 'Discovery' events to enthuse and inspire. We are delighted to be involved in facilitating this programme at the Raspberry Pi Foundation, given that the original purpose of creating the Raspberry Pi itself was to address the pipeline issues facing universities recruiting computer science students. Eben Upton – co-creator of the Raspberry Pi computer and CEO of Raspberry Pi Trading – has said he wants students to see computing as a "platform for creating their own destiny", and that is the future we seek to inspire with these events.

We hope that the events programme will enhance the abilities and progress of A-level students. They will also act as an opportunity to, inform them about the opportunities for taking Computer Science degrees. We will be partnering with universities across England to deliver events regionally, working with course leaders and outreach teams to engage communities and schools. Our hope is to close the gap between A-level and undergraduate study,

■ The first Isaac Computer Science 'Discovery' event will take place on Monday 1 July at the University of Cambridge

ISAAC COMPUTER SCIENCE EVENTS

The programme of events will focus on:

- Discovery: computer science courses, research, and possible future career paths
- Masterclasses: stretching students' knowledge of A-level topics
- Boosters: increase students' confidence and ability
- Teacher professional development: recap or learn new skills

helping students to make a smooth transition into higher education. It's really important for us to emphasise the career opportunities for Computer Science graduates in an increasingly digital age, and I'm looking forward to speakers from industry and academia joining our 'Discovery' events to share their experiences and insights. There will be some fun tech to play with, too.

The events programme will be starting with our first 'Discovery' event on Monday 1 July at the University of Cambridge – look out for the event opening on Raspberry Pi social media channels soon. To receive more details about the Isaac Computer Science platform launch and the events programme as it becomes available, please register your interest at **rpf.io/alevel**. **(HW)**

THE COMPUTER SCIENCE
ACCELERATOR PROGRAMME

Claire Arbery introduces an exciting new opportunity from the National Centre for Computing Education for those teaching, or wanting to start to teach, GCSE Computer Science

Computing is vitally important for young people today: it contributes to their personal and intellectual development and their wider understanding of the world. The global economy is increasingly reliant on a digitally skilled and knowledgeable workforce and that many young people currently at school will end up in jobs that do not yet exist.

However, in the UK most GCSE Computer Science is taught by non-specialists. GCSE Computer Science is only available in 52% of secondary schools in England, and only 11% of students choose to take it, most of whom are male. Girls are outnumbered in GCSE Computer Science classes by 4:1.

The Computer Science Accelerator Programme has been designed to change these statistics. We want to empower teachers to confidently teach GCSE Computer Science and we want to persuade more schools of the importance that their students have the chance to study it.

What support are we providing?

The Computer Science Accelerator Programme is a supportive programme for secondary teachers who currently teach Computer Science at GCSE, or plan to do so. It is designed to be flexible and highly personalised to fit into teachers' schedules, enabling them to study in a way that works for them.

It is delivered through a combination of face-to-face and online courses. After completing a simple diagnostic test which will help you to identify any misconceptions or gaps in your knowledge, you can choose the courses that best meet your needs. You can currently select from four face-to-face courses and ten online courses, so long as you pick at least two of each. But there is no limit to the number of courses you take and as we introduce more courses to our calendar, you are welcome to complete as many as you like!

This programme aims to boost GCSE Computer Science teachers' subject knowledge. You will be able to extend your understanding of algorithms and data

PROFESSIONAL DEVELOPMENT

"The Computer Science Accelerator programme has been some of the best CPD I have ever done. It has been of great help to me in my professional development and already is very useful to my school in developing our Computer Science Curriculum at all levels. I thoroughly recommend it to any CS teacher."
Adam Van Asch,
Second in Department
ICT/Computer Science

systems, and explore computer systems. You can take your learning online and unpick network and cybersecurity issues, create simple systems that respond to and control the physical world using a Raspberry Pi, or learn the essentials of Python programming.

The face-to-face courses are delivered across England, through our network of Regional Delivery Partners. Each course is two days long, with a gap task for you to complete back at school in between, giving you the opportunity to test out and embed what you have learned. The online courses can be completed at a time and location that suits you.

Helping you with the costs of participating

We recognise that it can be difficult to get out of the classroom to attend face-to-face professional development courses, so we have worked hard to minimise the cost of the programme. The Computer Science Accelerator Programme and its face-to-face courses are free to secondary teachers of computing or computer science in English state-maintained schools who do not hold a post-A-level qualification in Computer Science or a related subject . Online courses are free to all teachers and form part of the Computer Science Accelerator Programme.

We offer bursaries to state-maintained schools and colleges, which are designed to contribute towards the costs associated with attending professional development,

■ The Computer Science Accelerator programme supports secondary school teachers of computer science

including course fees, travel, and supply cover. They exist to support teachers through the required 40 hours to complete the programme.

Recognising your achievement

Once you have successfully completed this programme, you will receive a nationally recognised qualification, 'The National Centre of Computing Education certificate in GCSE subject knowledge'. There is also a pathway to chartered status should you wish to pursue this further.

Completing the programme demonstrates that you have reached a nationally recognised benchmark of competence,

which will help you to progress in your career. As a graduate of the Computer Science Accelerator Programme, you are also automatically entitled to free courses from the National Centre for Computing Education that will add excellent pedagogy to your subject knowledge. The Computer Science Accelerator programme also counts towards your National Centre for Computing Education secondary teacher accreditation. (HW)

■ Getting started is easy.
Visit **teachcomputing.org/accelerator** to register and start your journey today.

CLAIRE ARBERY
Claire is Programme Manager for the Computer Science Accelerator Programme.

A BOOST FOR
PRIMARY COMPUTING

Ben Hall explains what the National Centre for Computing Education means for primary teachers

Until just a few months ago I was a full-time primary school teacher in Rochdale, splitting my time between a Year 6 class in the morning and a Year 1 class in the afternoon, spending my lunchtimes marking Year 6 books while simultaneously preparing for provision in Year 1! At the same time, I was the computing coordinator, encouraging a great bunch of teachers to tackle the subject in their classes, in many cases for the first time. In ten years in the classroom, I have seen many initiatives come and go; however, two challenges remain constant: time and resources. These apply across the curriculum, but are particularly felt in a resource-hungry subject like computing.

An extraordinary challenge

As somebody who's always loved technology and had an interest in computing (as well as a background in the IT industry), I settled comfortably into the role of computing coordinator, eventually as a specialist teacher. I was lucky to have access to some great resources and lots of support through the CAS networks, who accepted me as a Master Teacher in 2016.

Not all teachers have these opportunities. For many, the switch from ICT to computing in 2014 created more problems than it solved – how would they find the time to learn a new subject and fit it into an already packed curriculum? Primary teachers are

renowned for their versatility; however, we all have our weak point. Many find computing a step too far and it can get pushed to the margins.

The 2017 Royal Society report 'After the Reboot' assessed the teaching of computing in England as "patchy and fragile" with a majority of teachers "teaching an unfamiliar subject without adequate support". The consequences of this are worrying. In 2018, there was a 17% drop in the number of children leaving school with a computer-related qualification, with an increasing gender imbalance in those who did take up the opportunity to study the subject.

To make matters worse, this is all happening in an environment where computing skills have never been more important. The digital skills gap, defined by Digital Skills Global as the lack of digital skills amongst the existing workforce and the lack of properly trained graduates to fill posts in growing technological industries, is growing. The consultancy firm Accenture estimates that these shortages could cost the UK economy over £140bn over the next ten years. The pace of change in the digital economy has caused the government to predict that within 20 years, 90% of all jobs will require some element of digital skills. This is without considering the proportion of jobs that do not yet exist.

That's the bad news; however, all is not lost. Working in a school where computing

computing lessons, no matter what their level of experience or expertise in the subject. These will be freely available to all.

To ensure we're keeping up-to-date, continually learning, and creating resources that are the best that they can be, the NCCE resource repository will be research-based. As well as practical strategies, such as reading code (would you ask a child in Reception to write before they have picked up a book?) and paired programming (great for engaging more reluctant learners), we will be drawing on overarching methodologies, such as Predict, Run, Investigate, Modify, Make (PRIMM) developed by the Raspberry Pi Foundation's Chief Learning Officer, Sue Sentance. Our plans will cover the whole breadth of the curriculum, from helping children use computers safely and

with my local NCCE hub to widen my professional network. It will be great to see many other primary teachers taking up this opportunity, too.

The NCCE is also providing comprehensive professional development opportunities in the form of online and face-to-face training. Face-to-face courses are already being delivered by NCCE consortium partners, STEM Learning. Some online training is already freely available. For example, those new to computing can learn the fundamentals of teaching programming in primary schools through Scratch. This course will help you access computing in a clear and concise way, giving you practical activities to take back to the classroom. More experienced teachers can look forward to thorough research-based investigations into wider aspects of computer science, helping you stretch your knowledge and develop computing throughout your school.

The pressures on teachers are not going to ease anytime soon; however, with the right support and encouragement, computing can be given the profile it deserves in all schools. The NCCE is here to ensure that all teachers have access to the very best content and resources which are, in most cases, freely available. We have a once-in-a-generation opportunity to close the digital skills gap and develop exceptional computer scientists. I hope you will all join us on this exciting journey. (HW)

> ## " OUR PLANS WILL COVER THE WHOLE BREADTH OF THE CURRICULUM, FROM USING COMPUTERS SAFELY, THROUGH TO DIGITAL MAKING AND ROBOTICS "

has been valued, and where children have the opportunity to develop digital skills from an early age, leads me to believe that we can inspire a generation of computer scientists. Children love technology and if we can relate it to their future careers and aspirations, we have an opportunity to do something special. How often, in other subjects do you get that 'wow' moment when the program works, or the robot moves?

Resources for all levels of expertise

I am delighted to have the opportunity to be part of a project which enables any teacher to feel confident about teaching computing. I am working alongside several other primary computing specialists at the Raspberry Pi Foundation to develop a comprehensive set of resources for the National Centre of Computing Education. There will be schemes of work, lesson plans, activities, and assessments - equipping teachers with the tools they need to deliver outstanding

responsibly, through to digital making and robotics. We will also be drawing on some of the excellent resources already available.

Subject experts will also benefit from the NCCE. As well as the opportunity to contribute ideas and resources, they will be able to adapt activities from the resource repository to meet their own requirements under the Open Government Licence. The NCCE network of hubs and CAS Communities of Practice will also give teachers the opportunity to showcase their expertise with other teachers and schools.

As a subject expert in computing, I was always looking for ways to develop my own skills and those of my colleagues. The NCCE will enable those with confidence in computing to take and adapt activities under the Open Government Licence. I am continuing to run CAS Communities of Practice in my local area. These provide a space where teachers can share projects and ideas. I am also looking forward to engaging

BEN HALL
Ben (@hengehall) is a Primary Learning Manager at the Raspberry Pi Foundation. He is a CAS Master Teacher and runs a Code Club at Stubbins Primary School near Manchester.

A NEW ERA FOR COMPUTING AT SCHOOL BEGINS AS THE NATIONAL CENTRE FOR COMPUTING EDUCATION IS LAUNCHED

Simon Humphreys outlines the role of CAS Communities of Practice

O ver the past decade, the Computing at School (CAS) network has provided mutual and beneficial support to many thousands of UK teachers dedicated to their subject and their pupils. Through coming together and sharing their skills, knowledge, and practice, they have found better ways to address the challenge of introducing a new subject into the school curriculum. Through our network of local communities, supported through online discussion and exchange of resources, teachers' confidence has steadily grown.

The newly launched National Centre for Computing Education will bring a much-needed catalyst to embed that support and training. Here, we discuss how CAS Communities of Practice will be working as part of the National Centre for Computing Education to address these ongoing challenges.

Challenges

I'm mindful of the many conversations I've had with teachers about how difficult it is to recruit staff with the right skill set, and the difficulties of being released to attend much-needed training. CAS has been working tirelessly in this space for over ten years and brings something unique and powerful to the National Centre for Computing Education.

The CAS community is part of BCS, The Chartered Institute for IT, one of the partners in the National Centre for Computing Education consortium with STEM Learning and the Raspberry Pi Foundation. Together, we will be leveraging, expanding, and improving the existing CAS Community network as part of its aim to upskill 8000 computing teachers by 2022.

For CAS, the existing community will continue to be its lifeblood and foundation. Facilitating and brokering connections of teachers in their local area remains at the heart of CAS activity. But the National Centre for Computing Education provides us all with the opportunity to make those community engagements more effective and meaningful.

HOW TO GET INVOLVED WITH CAS

- Join CAS at **computingatschool.org.uk**
- Join a CAS Community of Practice
- Apply to run a CAS Community of Practice in your area – **helloworld.cc/ CAScommunityleader**

Networking for teachers

Since CAS started in 2008, we have focused on building connections between teachers in their local areas, working with local volunteers to host termly meetings in both primary and secondary schools. The main priority for these meetings is that teachers return to their classroom with something they can use. Many participants point to the benefit of these meetings and the value of the local teachers developing their own confidence and understanding of the subject.

Since 2016, these local networks have been supported by the CAS Regional Centres set up as part of the DfE-funded CAS Network of Excellence. That programme came to an end in March 2018 (Hello World, Issue 6).

The new programme is centred on 40 school-based computing hubs, which represents a significant uplift in both the funding being made available to support the teachers and our subject, but also increased access to training centres across the country.

What's in a name?

So far I've avoided using the phrase 'CAS Hubs', which has been the term we used to name the CAS local meet-ups since the very first one was launched in Cambridge in 2009. Having two networks with the name 'hub' in the title will end up confusing us all,

1 At least one resource for teachers to use in their classroom

2 Theme or topic relevant to teachers of computing and computer science

3 Enable teachers to collaborate on a task or activity - to 'have a go'

4 Provide opportunity for teachers to chat and network with each other

■ The CAS Charter outlines the aims of CAS Community meetings

so the CAS Hubs are being renamed the CAS Communities of Practice. Each will be named according to their location, e.g. CAS Cambridge, CAS Coventry, etc. Those running the groups will be known as CAS Community Leaders, and that role will be highlighted in their membership profile on the CAS online forum.

The name may be changing, but the objective and role of these groups remains the same as it ever was. They will dock into the 40 school-based computing hubs and CAS will be able to expand our network and provide better training and support for our Community Leaders, with the support of the National Centre for Computing Education, as they build their community of practice with primary and secondary teachers of computing.

CAS members share a common concern of how to teach their subject with the maximum benefit to their pupils. We come together to fulfil those goals. Ideally, these are not one-off meetings, but something which is sustained over time and enriched through regular commitment and contribution to each other, building confidence, trust and relationships. These communities are the embodiment of a community of practice.

The CAS Community Leaders commit to running at least one meeting per term; each meeting will centre around a topic of relevance to the teachers led by the community leader or invited speaker or a member of that group itself, but the aim is uppermost: all will return to their schools with something to use in the classroom.

The format of the meetings varies. It may be directed from the front, or a collaborative activity where a lesson plan or resource is being reviewed and discussed – the choice is left to the Community Leader and the group to decide.

To get involved with these groups, enter your postcode on the front page of the CAS website (**computingatschool.org.uk**) and a list of the local communities will be shown. Or, if you are already a member of CAS, go to your profile page and your nearest CAS Communities of Practice will be listed there.

Alternatively, you can offer to lead a session. We all have something we can contribute, whatever our level of confidence or experience with the subject. Sharing what you are doing is always a great encouragement to others. (HW)

■ We are always looking for new CAS Community Leaders. If you are UK-based and there isn't an active CAS Community of Practice in your area, please complete the application form: **helloworld.cc/CAScommunityleader**

SIMON HUMPHREYS

Simon Humphreys is Head of Computing at School, part of BCS, The Chartered Institute for IT.

■ CAS Master Teacher Tig Williams at a CAS meeting, encouraging computer science teachers as they work through a tricky task

NEW ROUTES TO CERTIFICATION
IN TEACHING COMPUTING

Simon Humphreys explains why certification is a crucial component of the new professional development programmes offered by the National Centre for Computing Education

One of the key demands for the National Centre for Computing Education will be to provide a programme of continuing professional development for teachers in both primary and secondary schools. It's vital to acknowledge that, for the most part, teaching is a graduate profession and many who have chosen this career have already worked very hard to achieve their qualification to teach. The certificate offered through the National Centre for Computing Education is not a qualification to teach, but the award will reward and recognise the teacher's:

- level of subject knowledge
- their commitment to professional development
- their understanding of appropriate pedagogies for teaching computing.

Three types of certificate

Teachers that gain new knowledge and skills will be awarded a meaningful certificate with currency in the wider school system. It will provide a nationally recognised benchmark and will be an important aspect of career development for teachers. The certificate will also provide a pathway to chartered status.

> **Three certificates are available to teachers:**
> 1. Primary Computing
> 2. Secondary Computing and
> 3. GCSE Subject Knowledge – part of the Secondary Computing Certificate

All certificates will be awarded by BCS, The Chartered Institute for IT, and accredited by the Royal Academy of Engineering.

> **The three separate components to certification are:**
> 1. Commitment to professional development
> 2. Commitment to professional knowledge credits (plus passing the GCSE Subject Knowledge certificate)
> 3. Supporting the professional community

Teachers registered with the National Centre for Computing Education are automatically part of the certification programme. To earn credits, teachers will successfully complete professional development activities and record their reflections on what they have learned and its impact on their pupils. The teachers pick their own courses after completing an initial diagnostic assessment.

The GCSE Subject Knowledge certificate is an important part of the secondary certificate and can be awarded separately; it is primarily for those accessing the Computer Science Accelerator programme, discussed on page 18. It is based on the successful completion of 40 hours of professional development and a subject knowledge test.

To take full advantage of all that is on offer, you can register with the National Centre for Computing Education at **teachcomputing.org** and complete the initial diagnostic assessment to find out which courses would be most beneficial to your teaching. **(HW)**

COMPUTING EDUCATION RESEARCH: PROMOTING EVIDENCE-BASED PRACTICE

James Robinson explains how the National Centre for Computing Education works to help teachers apply the latest research evidence

As well as providing high-quality training for teachers and a complete repository of free and open teaching resources, the National Centre will be engaging teachers with best practices from an emerging body of research. Supporting teachers to equip themselves with the best tools and approaches, alongside their content knowledge, is crucial to providing students with the best experience.

What are the challenges?

In computing teaching there exists something called a 'Know-Do' gap – a period of time that it takes for new knowledge about a discipline

■ Teachers and researchers collaborating to discuss evidence and its application in practice

to be embedded in practice. This gap means that while there may be strong evidence for (or against) a particular approach or model, practitioners of that discipline are either unaware or under-prepared to change their practice for a significant amount of time. This results in practices with strong evidence not being used and ineffective or harmful practices persisting.

The Know-Do gap is by no means unique to computing – gaps exist in many areas, particularly where researchers and practitioners are different groups. Generally, researchers focus on communicating with other researchers to collaborate and build the body of knowledge. Practitioners often can't or don't engage with research, for a variety of reasons that include a lack of access, a lack of time, or even a perceived lack of relevance.

This problem is compounded in computing teaching, with those teaching the subject having many demands on their time with

little to spare for selecting, accessing, interpreting, and applying research findings. Many computing teachers have come to the subject from other areas of specialism; any time they can spare is necessarily spent developing their subject knowledge.

Finally, as recognised by the Royal Society in its report 'After the Reboot', more research into appropriate strategies and approaches for teaching computing is needed. The lack of research focused in schools, rather than universities, is an issue, as is the small scale of many studies carried out.

Engaging with research
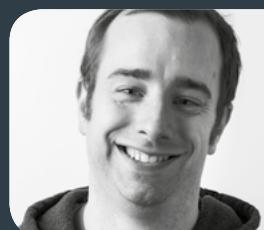
The National Centre is committed to helping teachers connect with and apply research evidence in their classroom. Through a variety of means, teachers will be able to access up-to-date research knowledge that has undergone reviews and been translated to advice for best practice. (HW)

RESEARCH BYTES

Research Bytes is the research-focused newsletter from the National Centre for Computing Education.

Each half-term, we'll present a selection of current research and evidence, hear from teachers embedding this evidence in their practice, and show you ways in which you can get more involved in computing education research.

You can sign up to receive the newsletter at **teachcomputing.org**

JAMES ROBINSON
James (@LegoJames) is a Senior Learning Manager at the Raspberry Pi Foundation, focusing on pedagogy within the National Centre for Computing Education.

PROMOTING A WORLD-LEADING COMPUTING EDUCATION FOR EVERY SCHOOL ACROSS ENGLAND

How will the National Centre for Computing Education increase the diversity of those pursuing studies in computer science? **Dave Gibbs** explains

Whether you take a short-term or longer view, technology is of vital importance to young people immersed in a digital world. It is of critical importance that they develop at least a basic knowledge of computer technology and digital literacy, and it is only fair that children are given the chance to express their creativity and satisfy their curiosity using computing devices.

The importance of computing to young people, right now, is clear.

- Nearly a quarter of four-year-olds have their own tablet device. A quarter of internet users first used the web when they were six or younger.
- Half of 9- to 16-year-olds use smartphones on a daily basis.
- 95% of UK 15-year-olds use social media before or after school. Around a third of them are classified as 'extreme internet users'.

The importance of computing to the future of the country has been demonstrated by the huge financial commitment made by the government to improve teaching of the subject. It is important that every young person has access to a world-leading computing education. To quote Sir Ken Olisa in the important BCS report 'Moving On Up', "in the world of digital, social mobility is practically frictionless." IT, a core pillar of computing, offers more social mobility than medicine and law, with more routes to entry

and a far lower cost for obtaining qualifications and skills than either of these more traditional vocations.

While computing presents an academically challenging and fascinating domain of knowledge in its own right, it is also clearly linked to skills and knowledge applicable in the workplace. The tech sector presents a world of opportunity, growing twice as fast as the economy as a whole. The productivity of its employees is well above the UK average, but a government report in 2016 stated that "a shortage in

> ## TEACHERS IN PRIORITY AREAS WILL BE GIVEN THE MOST FINANCIAL SUPPORT TO ACCESS PROFESSIONAL DEVELOPMENT

suitable digital skills for digital jobs persists in the UK labour market. This is a major risk to business growth, innovation, and broader societal development."

Yet, a worrying number of students, especially girls, just aren't interested in computing as a school subject, largely because its relevance isn't made clear, and the learning opportunities it presents aren't fully exploited. Of the girls who chose not to study GCSE Computer Science, 74% never even considered it as an option. For non-specialist teachers, especially, changing these attitudes presents a major challenge.

What's holding young people back?

There is a lack of awareness of career opportunities within the digital sector, and skill and gender stereotypes exist around the types of roles that are available. Barriers exist, especially for women, who are under-represented on higher education courses in computer-related subjects, and within the industry as a whole. Sometimes the issue starts at home. A recent UK digital skills report reveals that 23% of parents believe digital skills are irrelevant to their children's future career success, despite the fact that almost all jobs already require at least a basic level of digital skills.

Some of the regions in the UK with the worst levels of social mobility also have the poorest equality of opportunity for young people in computer science. In rural locations, young people are less likely to study GCSE Computer Science than their peers in cities. Almost all of the social mobility cold-spots recently identified by MPs are in rural counties, with the situation worsening with distance from London. Larger schools, usually found in bigger centres of population, are more likely to offer the subject than

■ Portsmouth is a growth hot-spot for employment in the technology industry

smaller schools, with the exception of some cities which also have poor participation in GCSE Computer Science.

Young people eligible for the pupil premium grant – an indicator of disadvantage – perform significantly less well than their peers in computer science and their take-up of the subject is falling. In 2017, only seven pupil-premium students gained an A* at A-level Computer Science in the UK. This situation must improve.

The problems are compounded for young black people who are, according to the Equality and Human Rights Commission, more likely to live in poverty. They remain under-represented among computer science students, potentially closing off a clear route to improving their own life chances.

It is common knowledge that big inequalities between young women and young men exist in terms of access to a high-quality computer science education. While London, again, does relatively well, other areas have room for improvement. Only one in ten GCSE Computer Science students in Middlesbrough are female. There were 25 local authorities with A-level Computer Science provision but no female participation. The Borough of Knowsley in

Merseyside had no A-level provision at all. The situation seems to be worsening: 30,000 fewer girls were entered for qualifications in computing subjects in 2017 than the year before. Meanwhile, nearly 87% of all A-level cohorts are part-filled at unsustainable levels. While these issues aren't always directly connected, they are chapters of a larger story. Computing should be for all, but it currently isn't.

The National Centre for Computing Education is pouring its support into the places that need it most. Teachers in priority areas will be given the most financial support to access professional development close to their schools.

There is much room for hope. Many of the areas where we can concentrate our support are those with growing tech ecosystems, providing future employability opportunities, as well as enrichment through industry-education partnerships. According to a report from techUK, employment in the sector is rising in many UK towns, with growth hotspots in Reading, Basingstoke, and Portsmouth.

The National Centre for Computing Education's mission is simple – to increase the number of teachers of computer

science and hence the number of students taking the subject from all backgrounds. We also aim to imprive the quality of teaching to every child. We hope to make access to a great computing education fairer for all.

Figures on computing education in the UK in this article are taken largely from The Roehampton Annual Computing Education report (helloworld.cc/roehampton). (HW)

DAVE GIBBS
Dave (@adgibbs) is the educational lead for computing at STEM Learning, part of the National Centre for Computing Education consortium. After working as a teacher and head of ICT, he has supported computing teachers through CPD, resources, and projects since before the new National Curriculum.

#INSIGHTS

SCRATCHMATHS: INTEGRATING COMPUTING AND MATHEMATICS

STORY BY Jonathan Dickins

Evidence has shown that learning programming can help students to develop higher levels of understanding of certain mathematical concepts. A project from UCL Knowledge Lab has been testing a new approach, involving learning computing alongside mathematical concepts. They have designed an integrated curriculum, using the programming environment Scratch, to teach mathematical ideas. A recent evaluation has found that using this approach improves computational thinking. There was no evidence, however, for an improvement in maths attainment, as measured by Key Stage test scores.

ScratchMaths is a two-year computing and maths-based curriculum for pupils in Key Stage 2 in England (ages 9–11). Scratch is used to integrate coding activities into mathematical learning to address a key problem that students have in learning mathematics – expressing mathematical concepts in formal language. The programme focuses on Scratch programming skills and computational thinking, with explicit links made to areas of mathematics. ScratchMaths provides teacher professional development to deliver the curriculum, along with a range of teaching resources. John Morris, Headteacher at Ardleigh Green Junior School in Essex, says, "For us, ScratchMaths is a breath of fresh air, because it allows us to approach the teaching of mathematics in a new, exciting, and engaging way."

The programme was funded by the Education Endowment Foundation (EEF), which has recently released a report produced by Sheffield Hallam University evaluating the impact of ScratchMaths on a sample of schools that worked with the curriculum over a two-year period.

110 schools, with over 6000 pupils, were involved at the beginning of the trial, with around half in the intervention group where ScratchMaths was taught for at least one hour every fortnight. They were compared to the other half of schools, who did not use ScratchMaths, in a randomised control trial. Teachers in participating schools received professional development for teaching ScratchMaths, with teacher mediation seen as a crucial factor in helping pupils to build links between computing and mathematics.

Participating schools were located across England, with schools in deprived areas included. Piers Saunders, lecturer in Mathematics Education at UCL, explains, "The curriculum was designed to meet the needs of all children, not just for high-attaining children, or those that often attend clubs. It really was for all children, as evidenced by the large range of national schools that we had involved."

Evaluation of impact

The independent team at Sheffield Hallam University evaluated the computational thinking scores of pupils after one year of intervention, and the maths scores of pupils after two years. They also gathered data from teachers via surveys and telephone interviews, and assessed how schools had implemented ScratchMaths.

They found no evidence that ScratchMaths impacted pupils' KS2 maths

---

THE 5 Es OF SCRATCHMATHS

- **Explore:** Moving from direct control to planning/building behaviours
- **Explain:** Experience concepts before defining them through discussions
- **Envisage:** Include unplugged activities and encourage identification with the object being programmed with
- **Exchange:** Provide the need to share or build on others' ideas
- **bridgE:** Explicit links to mathematics national curriculum

attainment as measured by Key Stage scores, but pupils in schools using ScratchMaths were found to have made additional progress in computational thinking relative to schools that did not implement ScratchMaths. Teachers reported that ScratchMaths was useful for addressing certain aspects of the primary computing curriculum, and good for improving Scratch skills.

This positive effect on computational thinking skills did not differ between boys and girls who were taking part in the evaluation, which is a noteworthy finding, as previous interventions in programming education often highlight gender differences in how children benefit. The evaluation also found that progress was highest for children who were, or had been, eligible for free school meals, suggesting that ScratchMaths is an accessible curriculum for children of different socioeconomic backgrounds.

It was also reported that many schools did not fully implement ScratchMaths during the trial, with attendance of training sessions, use of materials, and time spent teaching ScratchMaths decreasing between years one and two

" SCRATCHMATHS IS A BREATH OF FRESH AIR

of the trial. Pressures around SAT testing of 10- and 11-year-old students in the second year were reported as a barrier to implementation for many teachers. Some students involved in the trial also experienced a change of teacher between years one and two, with incoming teachers not receiving professional development from ScratchMaths, and perhaps being less proficient in Scratch.
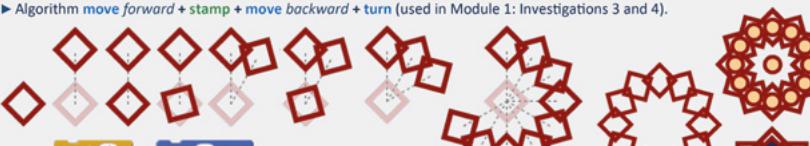
Given that the second year of the trial was the year in which computing concepts were to be more explicitly linked with mathematics ideas, it could be that these pressures resulted in the lack of improvement in mathematics attainment, and that application of the curriculum as intended could see the predicted
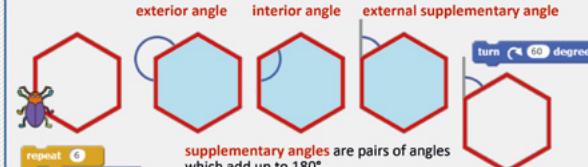
■ A resource from the Tiling Patterns module in ScratchMaths
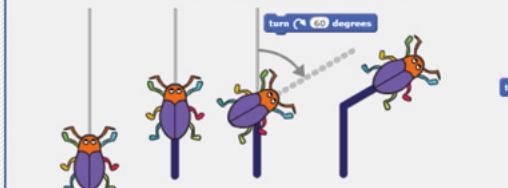
■ All ScratchMaths resources are freely available at **helloworld.cc/scratchmaths**

effect on mathematics attainment. It is also possible that the timescale of the study wasn't long enough to observe this effect.

With the ScratchMaths resources free and online, schools from a variety of regions are exploring implementing the curriculum in their settings. There is a trial underway in Spain measuring impact on students' computational thinking skills, and some educators have localised the resources to make them relevant for their existing Computing and Mathematics curriculum. In the UK, local coordinators are in place to give advice about implementing ScratchMaths.

While ScratchMaths has not been shown to raise mathematical attainment in this trial, the report suggests that where there is a need to develop teachers' computing skills, ScratchMaths could act as a low-cost form of professional development, helping teachers to develop computational thinking skills in pupils.

You can find out more about ScratchMaths and access resources at **helloworld.cc/scratchmaths**.

Are you using ScratchMaths? We would love to feature your thoughts on the resources in *Hello World*. Email us at **contact@helloworld.cc**. **(HW)**

OFCOM FINDS CHILDREN SHIFTING TO ONLINE VIDEO

STORY BY Oliver Quinlan

Every year the UK's communications regulator, Ofcom, runs a study to understand how children and young people aged 5–15 use media and communications. In previous years, the time children spent on online had been increasing, but this seems to have levelled out since 2018, at an average of two hours and eleven minutes per day.

What is changing is the time spent watching TV; average TV time has fallen to be 19 minutes a day less than time spent online. It seems that many children are using online services instead of traditional TV, with the most used online destination for children being YouTube. More than half of children who watch both traditional TV and YouTube say they prefer YouTube.
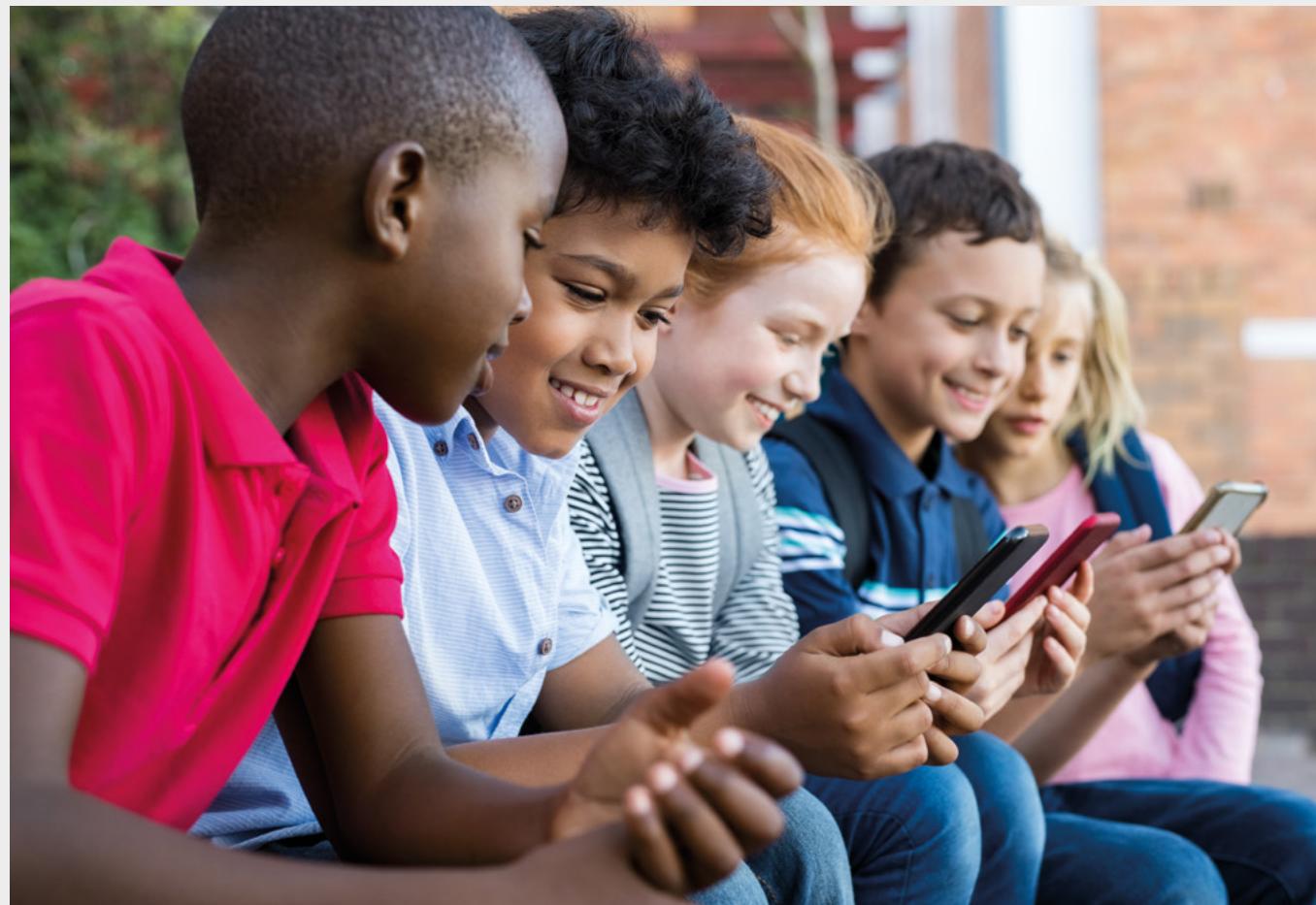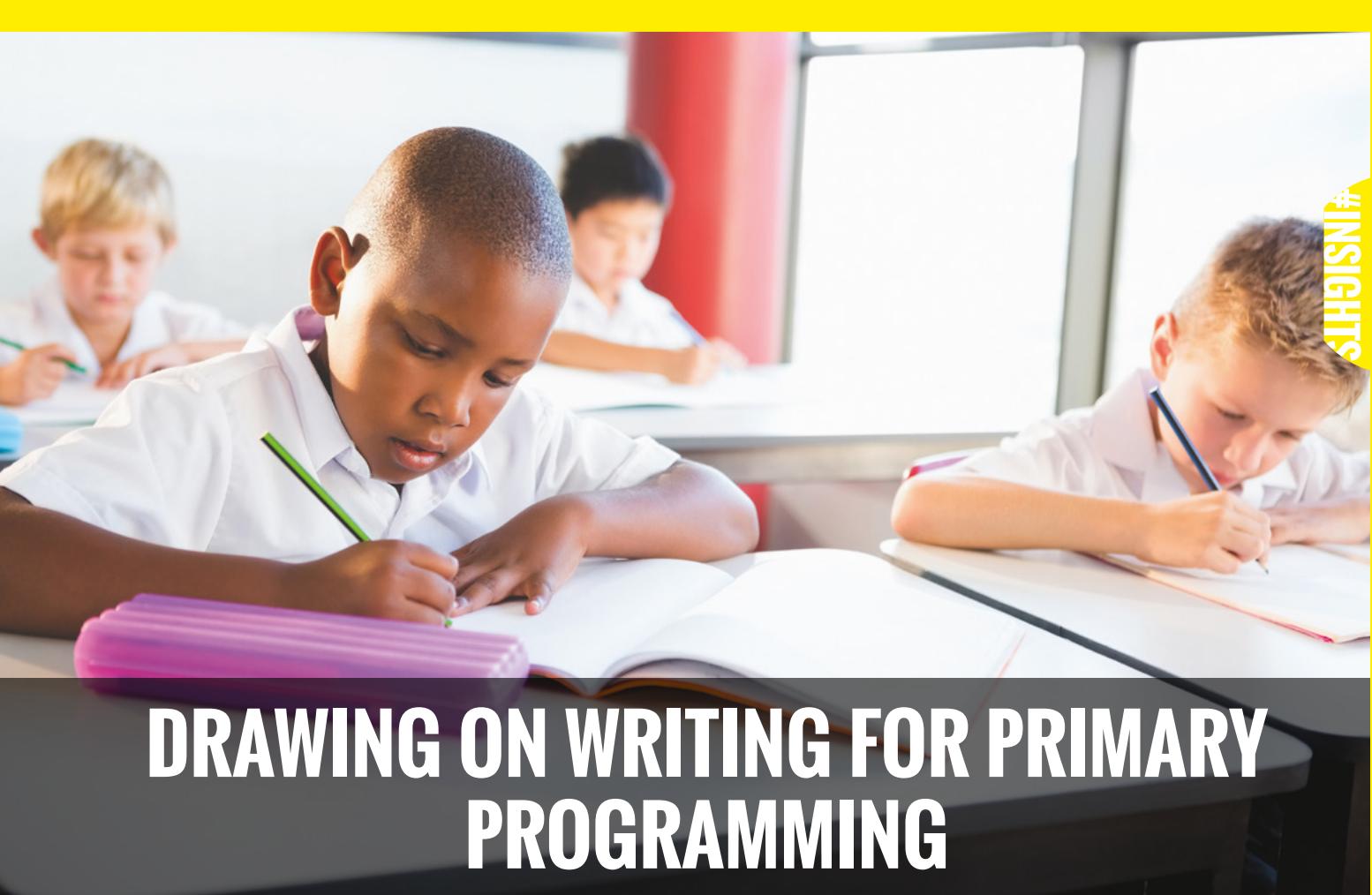
The reason for this seems to be that YouTube offers children choice. They can personalise their viewing to their interests due to the varied nature of YouTube content, and they can take control of what they watch. They like to watch videos related to their hobbies and passions, including offline activities like music or football. They follow well known YouTube vloggers and there is also a lot of interest in 'sensory videos', such as people playing with slime.

This is a notable shift in both the nature and the content of children's viewing habits.

In the past few years we have seen a move from children mostly watching content that is scheduled for them and watched live, often following the conventions of television, to a hugely diverse and self-directed viewing experience that can follow any direction they choose. This raises concerns, as with this freedom can come access to content that is produced with older audiences in mind. There are also many positive opportunities, as children use this diverse resource to explore and develop their interests, and take advantage of learning content.

Read the details of the Ofcom study at **helloworld.cc/timeonline**. (HW)

DRAWING ON WRITING FOR PRIMARY PROGRAMMING

STORY BY Oliver Quinlan

Teaching programming is new to many primary teachers. Although there is a lot to learn for this new subject, there are opportunities to transfer expertise from the teaching of other subjects, and teachers don't have to start from scratch with their teaching techniques. New research, by a team of researchers from Queen Mary University of London and King's College London, suggests that there is a big opportunity for teachers to draw on their skills teaching writing and build them into their teaching of programming.

It is well established that planning is an important part of the process when teaching writing. Teachers use resources, such as storyboards and planning grids, to help children organise their ideas, set out criteria they need to meet with their writing, consider audience, and focus on the purpose of their writing. There is research that shows planning has an impact on the development of children's writing.

The research team explored whether these kinds of approaches were being used by teachers for teaching programming, and how this related to their confidence in teaching. They gathered their data using a survey, with questions based on earlier work interviewing teachers in depth.

They found that teachers mostly reported planning as being a useful approach to teaching writing, with 78% reporting it was 'Essential' or 'Very Useful'. 88% reported that they 'Always' or 'Usually' used planning when teaching

of them are using design. The survey found that the longer someone had been teaching programming, the more likely they were to be using design. This suggests there is a huge opportunity here for more teachers to make use of design in their teaching, an approach that more experienced teachers take that others could learn from. It also suggests there may be potential in exploring

> ## " THERE IS A HUGE OPPORTUNITY HERE FOR MORE TEACHERS TO MAKE USE OF DESIGN IN THEIR TEACHING

writing, so teachers think planning is useful and take action to use it. The use of design in programming had a different picture. 82% of teachers said design was 'Essential' or 'Very Useful', but only 44% reported they used design 'Always' or 'Usually'.

Teachers think design is important for teaching programming, just as they think planning is important for writing, but fewer

other parallels between well-accepted teaching approaches in other subjects and how they could transfer to programming.

The full paper also covers more specific details about the types of design approaches teachers use, and differences between teachers of different genders and different levels of experience. You can read the paper at **helloworld.cc/primarywriting**. **(HW)**

RESEARCH AS A GUIDE FOR PROFESSIONAL DEVELOPMENT

Jane Waite gives an insight into her use of a conference paper in creating a professional development course on teaching networks

Reading papers can be an interesting and useful thing for teachers to do, and I wish I had read more research when I was a teacher. But it is very time-consuming, and it can be hard to know where to start. As I am going through the process of learning how to read and use papers now, I thought it might be useful to share my experience of doing this.

What is a conference paper?

Very simply put, a conference paper is a write-up of research that has been accepted to a conference. Conference papers can be short, at just two pages in length, but often they are around eight pages. Conferences are where academics come together to share their work. To get your work published in a conference, you submit your paper to the respective committee and they organise for your paper to be peer-reviewed by two or three academics (usually experts in the area of research). The reviewers decide if the paper will be accepted or rejected.

Why read this paper?

The paper I am reading is about teaching networks. I am reading it to help me devise a teacher professional development course which we are running at Queen Mary University of London. I plan to share different approaches for networing with the teachers on the course, so I am looking for research evidence on what resources and approaches are available, and what evidence there might be on the effectiveness of their use in classrooms.
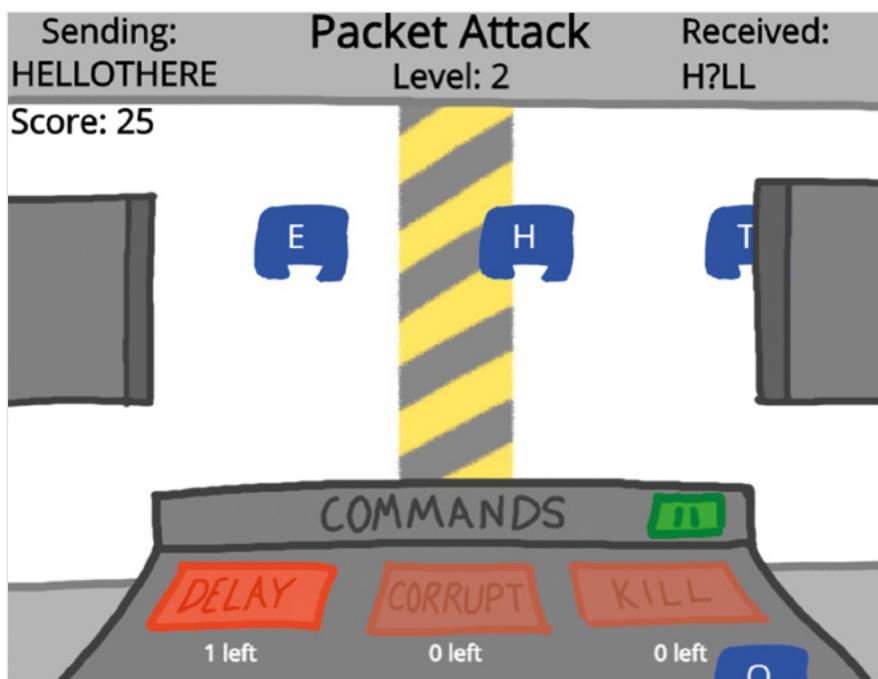
What is the title of the paper?

The paper's title is 'A game to teach network communication reliability problems and solutions'.

In a bibliography, it would appear as: *Jarman, S. and Bell, T., 2014. A game to teach network communication reliability problems and solutions. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education. ACM, pp. 43–49.*

Where was the paper published, and who wrote the paper?

I normally start reading papers by checking where the paper was published. This paper is from WiPSCE (Workshop in Primary and Secondary Computing Education), which is a well-respected computing education conference focused on teaching in schools, rather than at university level. The eight-page paper was written by two academics from New Zealand, one of whom I have heard of: Tim Bell, who was one of the original CS Unplugged authors (**csunplugged.org**) and is a very well-respected academic in the area of computer science education.

■ A screenshot of Packet Attack. Image courtesy of Jarman and Bell 2014 helloworld.cc/2GhOexb (CC BY-NC-SA 4.0)

What is the paper about?

From reading the abstract, I can see the paper describes a new online game, called Packet Attack, which was developed to help teach networking in New Zealand schools.

The game is part of the Computer Science Field Guide (**csfieldguide.org.nz**), an online textbook for teachers and students, developed in New Zealand. The focus of the game is to teach students key concepts on network communication. Students learn about protocols, as they actively construct their own understanding by playing the game.

There has only been a small-scale evaluation of the resource, so there is no clear evidence that the approach is more effective than any other. However, this looks like a very useful paper to help me develop the course on networks. Therefore, I read the rest in detail.

How will I use the paper?

Having read through the paper, I can now work out how I will use it.

15. Network Communication Protocols

15.1. What's the big picture?

Think about the last time someone sent you mail via the post. They probably wrote some content on some paper, put it in an envelope, wrote an address and put it in a postbox. From there, the letter probably went into a sorting center, got sorted, and was put in a bag. The bag then went into a vehicle like a truck, plane or boat. The vehicle either travelled through water, the air, or on the road. The postal system is a complicated one, designed to let individuals communicate easily, yet being efficient enough to group many letters into one postal delivery. The same ideas apply to how messages move around the internet.

■ One of the many resources in the Computer Science Field Guide. Image courtesy of Jarman 2014 helloworld.cc/csfg_ncp (CC BY-NC-SA 4.0)

As I want to know what resources are available to teach networks, this is a great paper – it has provided me with a game for teachers to look at and possibly to trial in class, but has also introduced me to the Computer Science Field Guide. The guide has an entire chapter, with videos and other resources, that the teachers can look at.

The paper describes why the authors adopted the approach of creating a game where the learner develops their own understanding by having to stop packets from being successfully delivered. I will refer to the author's ideas in the course and see if teachers agree.

Included in the paper is a critical analysis of several unplugged games, and other approaches for teaching about protocols, including using simulators. I can find out more about these approaches and include them in the course, as well as referring to the author's views on these approaches.

What next?

Despite being a small-scale study, the paper does provide me with a resource for teachers to compare against other techniques for teaching networks.

I now need to embed the resources and approaches from the paper in my course on teaching networks. I will also ask teachers what they think of these approaches. Finally, I will write to say thank you to the paper's authors for the contribution they made to the course from the other side of the world. (HW)

JANE WAITE

Jane is a former industry developer and primary school teacher. Jane now works in computer science education research, teacher professional development, and undergraduate teaching at Queen Mary University Of London.

COGNITIVE LOAD THEORY
IN THE COMPUTING CLASSROOM

Phil Bagge explains how implementing cognitive load theory has improved his teaching, and recommends some strategies to try

Cognitive load theory has gained in popularity since research into the idea began in the 1980s, and is backed by recent findings in evolutionary psychology. In 2017, the educationalist Dylan Wiliam tweeted that Sweller's research into cognitive load theory is "the single most important thing for teachers to know". And in February of this year, Ofsted defended the significant role cognitive load theory plays in its new inspection framework for English schools, which will come into force from September.

I believe that applying cognitive load theory to teaching has the potential to alleviate some of the problems at the heart of programming, especially at primary school level, and can help pupils develop long-term agency.

PHIL BAGGE

Phil (@baggiepr) is a Computing Inspector/Advisor working for Hampshire Inspection & Advisory Service, primary teacher, and CAS Master Teacher. He is the author of the **code-it.co.uk** free computing resources.

What is cognitive load?

Totally new information is first processed by the brain's limited short-term or working memory, which is typically able to hold two to seven elements at a time, depending on the complexity of the data and surrounding distractions. This information is then transferred into the brain's unlimited long-term memory. If the brain's working memory is overloaded with new information, it hampers, reduces, or stops the transfer of information into long-term memory, which is necessary to retain knowledge.

Cognitive load theory is based on the idea that, to maximise learning, we need to ensure that we do not exceed the cognitive load that our pupils can deal with.

Managing intrinsic load

Intrinsic load refers to the complexity of the new information that needs to be processed. This is strongly dependent on a student's prior knowledge, and while what we teach is often out of a classroom teacher's control, there are approaches we can use to keep intrinsic load manageable.

One problem comes when we introduce too many new concepts at the same time. To create even the most elementary programs, you often need many concepts and it can be tempting to use ones that pupils haven't learnt about yet. We like what it does, and we want to share it with our pupils, but are we creating more cognitive load that will lead to less long-term assimilation of ideas?

We can also be tempted to introduce a concept using complex examples. The
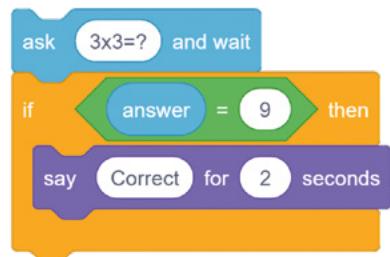
golden rule is: if the idea is totally new, strip away all complexity and present it in its most simple form, with the least information possible, and increase complexity gradually.

As an example, imagine we want to introduce the idea of conditional selection.

> If you are hungry
>   clap once
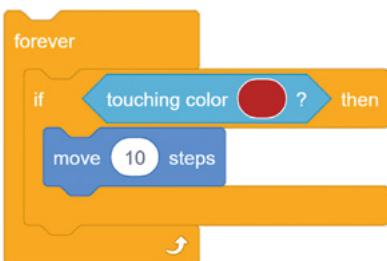
The above example presents a simple condition that triggers an action that will only be checked once. It is written in everyday readable language, with the only nod to computing being the layout and the indentation. It uses pupils' understanding of language, which they study on a daily basis.

Now consider:

This also uses conditional selection, but uses the popular quiz format. There is extra complexity in understanding how the 'answer' variable works with the input 'ask' block and in making sense of the more complex condition.

Finally, we could also use a typical example of conditional selection, where the condition is checked repeatedly inside an infinite loop:

This repeated checking adds more complexity and thus more cognitive load.

All of these examples are useful and will build understanding, but there is less cognitive load in the first, so I'd recommend it as the simplest starting point. It also has the added bonus of developing an algorithmic concept away from a specific programming language, so increases the chance of pupils transferring the knowledge to their long-term memories.

Worked examples and completion problems

Extraneous cognitive load is affected by how the information is presented, and what the learner has to do as part of the learning process.

A key strategy for reducing extraneous cognitive load is to provide pupils with worked examples that help them to see how a specific type of problem can be solved. In computing, this might involve showing pupils all the stages of programming, including a clear idea of what we want to create, an algorithm that includes initialisation and stages of coding, and the debugging process.

Designing effective worked examples is more challenging than we might think. Worked examples do not always elicit careful consideration in learners, and they can be less effective for pupils who are more advanced in the topic we are teaching.

One potential solution to these shortcomings is to make use of completion problems. These can provide a bridge between worked examples and conventional problems: pupils are given a partial solution that needs to be completed. This reduces scaffolding and ensures learners are engaged with the information being presented.

One strategy I have used is to provide pupils with an idea, code, and a list of bugs solved, and ask pupils to create the

■ An example of a completion problem: Year 6 students were tasked with modifying the algorithm to either collect a player's score or to include a player's name

algorithm. Alternatively, I have provided the algorithm, code, and bugs, and asked pupils to explain the idea.

Split attention effect

Another strategy for reducing extraneous cognitive load is to prevent the split attention effect. This is where two sources of information are presented alongside each other, neither of which is effective on its own, but in which both are needed for understanding. The learner has an increased cognitive load from trying to integrate both sources of information into a whole.

As an example, imagine we want to explain networks to pupils. We might provide a basic diagram illustrating the pathway that data packets take from our home computing device to a waiting web server to retrieve a webpage. Alongside that, we give them a description of each item in the chain and what it does. Neither example gives the full explanation on its own. To reduce cognitive load, we can integrate the text and the diagram into one resource with well-placed labels.

As with any aspect of learning science, there are criticisms and potential shortcomings of cognitive load theory. But the strategies I've shared here are well-grounded in empirical research: they are an effective way of helping our students to learn. I'd really recommend trying some of the approaches I've outlined and considering cognitive load in your teaching. (HW)

FURTHER READING

✔ John Sweller's recent review of cognitive load theory (open access): helloworld.cc/sweller2019

✔ Ton de Jong's discussion of the limitations of the theory (open access): helloworld.cc/dejong2010

✔ Ofsted's rationale on using cognitive load theory in its new inspection framework, from Daniel Muijs: helloworld.cc/ofstedclt

✔ Greg Ashman's blog post on how cognitive load theory has influenced his teaching of maths: helloworld.cc/ashmanclt

SUBSCRIBE

SIGN UP TODAY FOR **FREE**

THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS

(Hello World)

INTERACTIVE FICTION

Combining computer science & literacy

Issue 8 | May 2019 | helloworld.cc

COOLEST PROJECTS
The world's leading technology exhibition for young people

SILICON VALLEY
An itinerary for the ultimate computing school trip

TEACH COMPUTING
With the National Centre for Computing Education

NO HEADACHES HOMEWORK
A strategy that helps students and eases workload

PLUS SCRATCHMATHS • COGNITIVE LOAD THEORY IN ACTION • CARE WORKERS TEACH CODING • PARSON'S PUZZLES
RASPBERRY PI ARCADE GAME • GETTING STARTED WITH THE CRUMBLE • DIGITAL MAKING AT LIBRARIES & MUSEUMS

FREE IN PRINT
For UK-based educators!

SUBSCRIBE TODAY

- **Get five term-time issues**

- **Have them delivered directly to your door**

- **Hello World is not available in stores!**

KAREN MOUWS DIGITAL CREATIVITY AND INCLUSION

WHAT ARE WE CODING FOR?

Programming as an educational activity is continuously gaining popularity and is here to stay. But **Karen Mouws** asks the big question: why are we doing it?

I'd like to invite you to look at the organisations around you that provide coding activities. There are probably plenty to choose from. Now, consider the rationale of their existence: what exactly are they teaching coding for?

There are already over 8 million ICT specialists in Europe alone and, by 2020, we will need a lot more. But should obtaining a job in ICT be the only reason to learn how to code? In short: should computational thinking be reserved for those that are most likely to become a professional programmer in the near future or later in life?

Vocational vs pedagogical

We can position programming activities on a scale based on their perspective or rationale: from the extreme vocational to the solely pedagogical one. The former will want to focus on delivering skilled workers to a growing market; the latter only cares about the educational

foundations. Probably most of the organisations you thought of will be somewhere near the middle.

When looking at the scale, both perspectives are perfectly valid. However, I'd like to pose that a mere vocational focus can never be enough. Programming languages and syntaxes change quickly, rendering previous courses obsolete if learners have no foundation in the basic concepts of coding. Successful coding education requires a solid pedagogical vision, founded in constructionist principles: learning by making, solving authentic problems, powered by conversation between learners.

Of course, the vocational perspective is a very urgent one. We need these activities to deliver our software professionals of the future. Some organisations are pushed by the perceived threat of losing jobs to AI and machine learning, but this cannot be our only motivation for teaching computational thinking to younger generations. Focusing solely on jobs will lead to the exclusion of a large proportion of learners, so we'll also miss out on a lot of hidden talent.

Experience has already shown that teacher expectations can make a push towards, or deter someone from, studying in a STEM field. Just look at the percentages of girls in science classes for GCSE or A-level exams. The same is true for programming. If we focus on jobs and jobs only, we will train the students we see as most likely to succeed. But what about those that vary in gender, ancestry, socio-economic status, cognitive abilities, and so on?

In a pedagogical vision to coding education – contrary to the vocational one – everyone should be given the

■ When focusing on jobs and jobs alone in coding education, we lose a lot of valuable opportunities for inclusion and empathy

opportunity to learn both computational thinking and soft skills. Skills such as problem-solving, debugging, and collaboration shouldn't be limited to the future workforce. Everyone benefits from them. The same thing can be said of the 'maker attitude' and a feeling of technological ownership. We own technology, not the other way around.

Empathy through inclusion

One of the main arguments for a more inclusive programming education program is that of empathy. When confronted with the onset of AI, the main careers that will remain will require empathy, emotional awareness, and human intuition. No computer program can ever replace the emotional range of a human being. But, it is a skill that we will have to train in ourselves and coming generations.

Allowing everyone to participate in coding activities will not only increase self-confidence across the board, but also empathy: students learn to understand different people in different situations. Even more: they learn that life is not 'Instagram perfect' and being able to empathise with others will become even more important.

The problems of the future will require a more inclusive approach, as our societies become increasingly global. The ability to understand different perspectives and design solutions in a way that takes multiple stakeholders into account will become even more of a key skill. No one wants facial recognition software that can only recognise Caucasian males under the age of 50. Autonomous vehicles will have to be taught to navigate the world with compassion and ethics.

Inclusion means that everyone gets the opportunity to learn, at their own level: meet them where they are at, regardless of differences. Foundations such as 'CS For All' already call for this. But, it also means that everyone gets a voice, a chance to create, and bring their perspective into the world, regardless of gender, ancestry, or ability.

Coding creates a whole new platform for minorities to express their opinions and needs. If we deny them the basic right and opportunity to learn how to code, it would be comparable to blocking their access to written or visual media. A more diverse base of programmers will increase the diversity of voices heard in society.

INCLUSIVE CODING IN BELGIUM

Capital Digital BXL
- Youngsters (16+) from Brussels are trained as coding monitors to lead vacation camps for children aged 8 to 12
- Children and youngsters experience new roles through code
- Participants gain a new-found sense of accomplishment and pride
- helloworld.cc/capitaldigital

Onbeperkt Mediawijs ('Media Literacy Without Limits')
- Adults and youngsters with cognitive disabilities are trained as media coaches and learn the basics of code
- Increased self-confidence and feeling of ownership over technology
- Promotes the ethos that computational thinking is for everyone
- helloworld.cc/mediawijs

What are you coding for?

Making your own activities more inclusive does not have to be a painstaking endeavour. You can start small. Proven tactics are creating mixed groups, an open atmosphere and trying out paired programming or mentoring roles. Maybe reserve some spots for students with different socio-economic status and deliberately invite them to participate.

Fostering collaboration across diversity can be challenging, but very rewarding. Allow your students to switch from the role of trainee, to mentor, to expert, back to trainee. Make pairs for a collaborative project that would not intuitively work together. Have students – or outsiders – pose design challenges for real-world problems.

So, I would like to invite you to consider what you are coding for. Which perspective are you starting from? There is no right or wrong answer: both perspectives are necessary. But I would argue that we have to look beyond the purely vocational goal of filling jobs of the future and seek to understand the value in teaching every child to code. (HW)

KAREN MOUWS
Karen (@LimsKaren) is an educator in digital creativity and inclusion in Belgium. She focuses on teaching coding to children, youngsters, and adults in vulnerable social positions or with special educational needs, as a way to empower them. She is a proud Raspberry Pi Certified Educator.

A WHOLE SCHOOL APPROACH TO DIGITAL SKILLS

Claire Buckler shares the benefits of embedding digital skills across the curriculum

L ike most schools, at Devonport High School for Boys we want to ensure that all of our students are equipped with the digital skills they need to thrive in the world. We have been working hard to ensure that even those students who don't choose to take computer science are technologically literate. Across the school, technology has a strong presence. We operate a bring-your-own-device policy, which will shortly be replaced with a 1:1 Chromebook scheme. Most departments have Chromebooks or iPads, and students are encouraged to use mobiles phones as learning devices. As a Google Reference School, we use Google tools in creative and innovative ways across the whole of our curriculum. This healthy relationship with technology is essential for our students to ensure that they are data aware, and can use digital tools to communicate and collaborate – essential skills for being digitally literate when leaving school. We offer a coding club here, twice a week, which is full of students who don't take computer science as an option but still see the benefits in learning to code.

The benefit of SLT support

All of this comes from having a school that sees digital literacy as an essential skill and is happy to provide the time and resources to get all staff and students on board. My role is a great example of this: I am the Director of the Learning Commons, a library and digital breakout space. We have 60 Chromebooks, which are available at break times, and an innovative learning space. The Learning Commons is often booked out by teachers who enjoy the freedom that the space offers, compared to a traditional classroom. Our Assistant Head, Nick Berryman, will bring his business students in when they need to access technology and says, "Digital literacy is an important skill and at DHSB we have embraced this. The Learning Commons inspires the future generation to be creative and learn the skills needed to adapt." Google Expeditions is currently a favourite app among teachers and students, who can use our class set of Google Cardboards – the affordable VR headsets.

Another way we involve students in technology is in our Digital Leader scheme. Students take the roles of the IT helpdesk staff, and technical support tickets are assigned to them. They are an amazing resource for the school, saving time and allowing technical staff to concentrate on infrastructure issues. Students that are successful in their bid to be part of the Digital Leader scheme take the new Google Applied Digital Skills course. They are available to run the theatre lighting desk, set up for assemblies, lead Code Club sessions, and produce graphics and informatics for departments across the school.

We currently have over 30 Digital Leaders, with a healthy waiting list. Some of the team are also e-safety ambassadors, and run relevant workshops. The students involved show fantastic problem-solving and creative thinking skills. But the most amazing thing about these students, in my opinion, is that very few of them are taking our popular GCSE Computer Science option. Clearly they have a great relationship with technology. One of our Digital Leaders,

BECOMING A GOOGLE REFERENCE SCHOOL
– DAN ROBERTS, HEADTEACHER AT DEVONPORT HIGH SCHOOL FOR BOYS

Devonport High School for Boys was approached to be a Google Reference School and we opted to use Google for Education to reduce costs, improve productivity, and efficiency. More importantly, we valued how the apps empowered young people to collaborate, rather than more traditional virtual learning environments which limited collaboration and learning.

■ DHSB Learning Commons: a space where students are encouraged to use digital technology as a tool for learning and collaboration

George, told me that he feels he has improved his digital skills by being part of the team, and this had enabled him to take different subjects at GCSE.

No time for complacency

With constant access to technology, we need to be mindful that our students don't fall into the trap of becoming passive users, but stay curious and enthusiastic about how technology works. We need to ensure that the students who do not opt to study computer science are still encouraged to find technology interesting and relevant. The next step for our school is to implement a makerspace. Our makerspace will be part of the Learning Commons, where students will be encouraged to experiment at break times and after school. Rather than being shoehorned into one subject, any teacher will be able to use the space to create and innovate across the curriculum. Part of our culture here is to inspire an innovative and entrepreneurial spirit in all of our students. Makerspaces achieve this by encouraging students to design, experiment, and build in a hands-on environment, without being required to

> **"** ALL OF THIS COMES FROM HAVING A SCHOOL THAT SEES DIGITAL LITERACY AS AN ESSENTIAL SKILL

meet any specified outcome, and the best thing about it is it will be led, of course, by our digital leaders.

Getting started

While having a hugely supportive Head and SLT has been key to our success, there are many small wins available. Setting up a Digital Leader scheme may seem labour-intensive, but we have quickly seen the benefits. Offering a Code Club is simple – the **codeclub.org** resources require no planning. A makerspace may seem a lot of work or expensive, but could be as simple as a few Raspberry Pis or micro:bits, with some electrical components, most of which are reasonably priced. All of these could encourage more students, outside of those who choose to study computing, to be inspired by technology. (HW)

CLAIRE BUCKLER

Claire (@clairegowland) is Director of Learning Commons and Teacher of Computer Science at Devonport High School for Boys in Plymouth, UK. She is a Level 2 Google Educator and Community Leader/ Trainer for Computing At School.

ARTEFACTS TO INSPIRE

How can museums and libraries support young people to get involved in digital making? **Izzy Bartley** and **Claire Duffield** explain how they have used public collections of historic objects as a springboard for inspiration

In January, the Libraries and Museums and Galleries teams from Leeds City Council came together for our first major collaboration entitled 'Queens of Industry': an all-girls STEAM hack, inspired by the rich textile history of the city. On the day, girls aged between 7 and 14 had the chance to solve historic and real-world challenges using technology, as well as creating their own punch card and turning it into a work of art using Minecraft.

Artist and maker Gemma May Potter ran a workshop entitled PatternCraft (**patterncraft.co**), where the girls used punch cards and an Arduino electronics platform to create 3D designs in Minecraft.

This provided the perfect historical tie-in with the Industrial Museum's collection of Jacquard looms, which used punch cards to create their intricately detailed fabric designs. Hack participants were introduced to the looms and handled original punch cards and Jacquard textiles during their initial tour of the museum, giving them a historical platform on which to build their experiences of this new technology.

Another challenge, set by the Museums team, focused on designing a safety device for the Victorian children who would have worked at the mill. At the beginning of the activity, participants were taken on a tour of part of the museum, witnessing the spinning mule in action and experiencing the noise, smell, and the very real danger this heavy machinery represented to the young workers. It was not uncommon for children to suffer severed fingers, broken arms, or head injuries working with these machines. Having witnessed the mule in action and the small crawl space these children had to work in, the hack participants had a real insight into the risks and hardships involved for the mill children, and could use this understanding when tackling the challenge.

The final challenge was entitled 'E-textiles: from the Mill to the Catwalk'. Inspired by collections from Leeds Libraries, the girls were asked to look at the artefacts from the past and imagine the future. They were challenged to think about textiles and fashion and the role technology plays in creating our clothes, and how it may become an integral part of our clothing in future. Would clothes talk or change colour, for example, or would the machine become the designer of the future?

An inspiring location
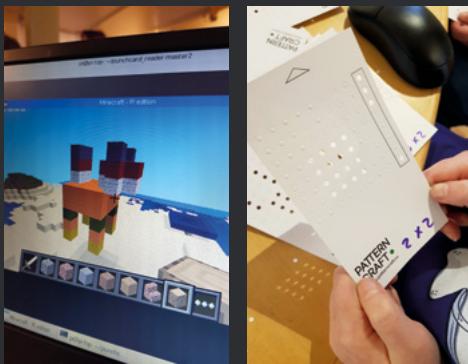
The Queens of Industry event took place in Leeds Industrial Museum. Once the largest woollen mills in the country, the museum's Textile Gallery now houses historic machinery such as carding machines, a working spinning mule, and Jacquard looms. Leeds Library collection includes essays and 'A Treatise on the Art of Weaving', providing sources of inspirational imagery, as well as a fashion collection curated by Kenneth Sanderson (1896–1977). Described as a 'sartorial feast', his collection

■ A participant shows her micro:bit warning device she's coded to use the radio function to protect Victorian child workers from being injured by mill equipment

HOW TO APPROACH OBJECTS AND DIGITAL MAKING

- Use mystery objects as an unplugged activity to arouse curiosity and spark inspiration.
- Challenge students to use new tech to improve or augment historic designs.
- Critically analyse the design of an object to identify its strengths and weaknesses. How can students apply this learning to their own designs?
- Use archive documents to highlight a historical problem and challenge students to come up with solutions.

of fashion ephemera spans textiles, small dolls (historically used as an early form of mannequin/catalogue), photographs, books, prints, and early magazines.

Cultural collections and computing education

Both the Libraries and Museums teams had been working independently to connect our collections with digital making. Leeds Libraries created a micro:bit-powered 'sorting hat' for their Harry Potter exhibition and an escape game for Ada Lovelace Day. In addition, Leeds Libraries runs regular Code Clubs across the city, providing access

Leeds Museums and Galleries comprises nine different sites, and Libraries have 34 branches. Between us, our collections are vast, and the range of exhibitions happening at any one time means that there are opportunities for (curriculum linked) digital making at every turn. The Queens of Industry event is the first in a series of collaborative efforts to inspire young people in STEAM.

Hack outcomes and beyond

Participants at the Queens of Industry event created varied and innovative projects within the hour they were given.

We had prioritised finding strong female mentors and role-models for the girls attending the hack, and we were brilliantly supported by technology companies and digital groups from across the city. Volunteers included software developers, designers, and regular Code Club mentors. This enabled us to respond positively to those hackers who needed a bit of extra support. A young girl with autism was initially reluctant to join in, but working one-to-one with a librarian, she overcame her anxiety and ended up presenting her design, a 'Praise Bit' which encouraged the clothes designer by using empowering messages, to a room full of 40 people.

Parent and participant feedback alike was incredibly positive: "[The hack] exceeded our expectations. It was an incredibly friendly event and it was really fantastic to have so many support / facilitators encouraging the girls and showing positive role models."

The end of the hack did not signal the end of making for the girls. Leeds Libraries runs a micro:bit loans system, and participants can choose to borrow a micro:bit, just as they would a library book. This initiative is a fantastic way to help widen access to digital making across communities.

As one parent put it, "It would be excellent if there were more events linking to tech and the women who helped shape IT through history to the present day." (HW)

> ❝ VOLUNTEERS INCLUDED SOFTWARE DEVELOPERS, DESIGNERS, AND REGULAR CODE CLUB MENTORS

to coding and creative tech to young people and families. In Museums, we'd been building circuits and coding to create a 'call the servant' board in one of our historic houses, and a touch piano for the music room. Having seen each other's work, we realised we were both passionate about using our collections and grounding new tech in the achievements of past generations, and using this to inspire today's young thinkers, explorers, and makers. From there, it was a natural step to start collaborating.

The girls programmed Sphero robotics to form textile patterns, light shows for the catwalk, and light-up danger warnings for Victorian mill children. BBC micro:bits were used to light up LED patterns inspired by weaving patterns, along with motion sensors and radio communication to warn of equipment proximity and injured children. In the PatternCraft workshop, participants produced repeating patterns, and combined different punch cards to create 3D sculptures.

IZZY BARTLEY & CLAIRE DUFFIELD

Izzy Bartley (@FireflyHeritage) is the Digital Learning Officer at Leeds Museums and Galleries. She is an RPi Certified Educator.

Claire Duffield (@flowersclaire) is Digital Engagement Librarian for Leeds Libraries (@leedslibraries).
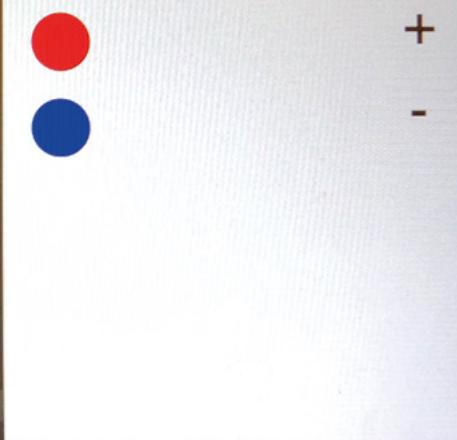
Saved: 12 days ago   Preview

```
1  // Painting program where user picks color
2  // by hovering over a color in a palette
3  // also change brush size when user hovers
4  // over + and -
5
6  // Paint brush only draws when mouse is pressed
7
8  // changing brushSize with a variable
9
10 var brushSize;
11 var red;
12 var blue;
13 var green;
14
15 function setup() {
16   brushSize = 25;
17   createCanvas(400, 400);
18   background(220);
19   // palette colors
20   fill('red');
21   ellipse(50, 50, 50, 50);
22   fill('blue');
23   ellipse(50, 125, 50, 50);
24
25   fill(0);
```

CREATIVE CODING WITH P5.JS

Saber Khan introduces p5.js, a JavaScript library for visual arts from the Processing Foundation, as a welcoming way to start text-based programming

The transition from block-based programming, on platforms like Scratch, to text-based coding can be challenging for students and teachers. Often, learners are asked to dive deep into arcane terminology and engage in projects that are uninspiring. Instead, prioritising accessibility and creativity, as Scratch does with its 'low floor, high ceiling' philosophy, can create an easy pathway to the powerful world of text-based coding.

p5.js and its older cousin Processing, a Java library and editor, share that spirit with Scratch and other block-based platforms. They seek to level the playing field of coding by bringing artists, students, educators, novices, and amateurs into the fold. Processing was created by Ben Fry and Casey Reas in 2001 at MIT Media Lab,

and p5.js was created by Lauren McCarthy in 2013. Both are open-source projects and under continual development from a community of contributors. p5.js offers an accessible avenue for students to learn the basics of coding via creativity, self-expression and art. It comes with an online editor, a growing body of curricula, and an inclusive community.

As a JavaScript library, p5.js adds extra functions, making it easier to create shapes, colours and animation. It creates a canvas on a webpage where the coder can quickly see the result of what they have coded. This quick and easy response helps students become engaged with the mechanics of how the code works. Each p5.js sketch or project starts with two pre-existing functions: `setup()` and

`draw()`. The `setup()` function, which runs once at the start of the sketch, is a good place to put in commands like `createCanvas(400, 400)`. The numbers in between the parentheses, called arguments, are often used with functions to specify how a command should be run. Here, '400, 400' is creating a canvas that is 400 by 400 pixels wide.

In contrast, in the `draw()` function, all the commands are looped endlessly from top to bottom. In there, you will see `background(200)`, which puts a colour on top of the canvas. With this knowledge in mind, it would be a good moment to head over to the online editor, which works on all major browsers, and try some coding for yourself. You don't need to have an account to use the editor. Please keep in mind that

p5.js will work in projects outside of the editor; you just to link to the p5.js library online in your HTML file.

Try this for your first program in the p5.js editor:

```
function setup(){
  createCanvas(400. 400);
}

function draw(){
  background(200);
  ellipse(200, 200, 50);
}
```

This will create an ellipse at 200 on the x-axis and 200 on the y-axis. The third number, 50, determines the diameter.

Let's make a painting program
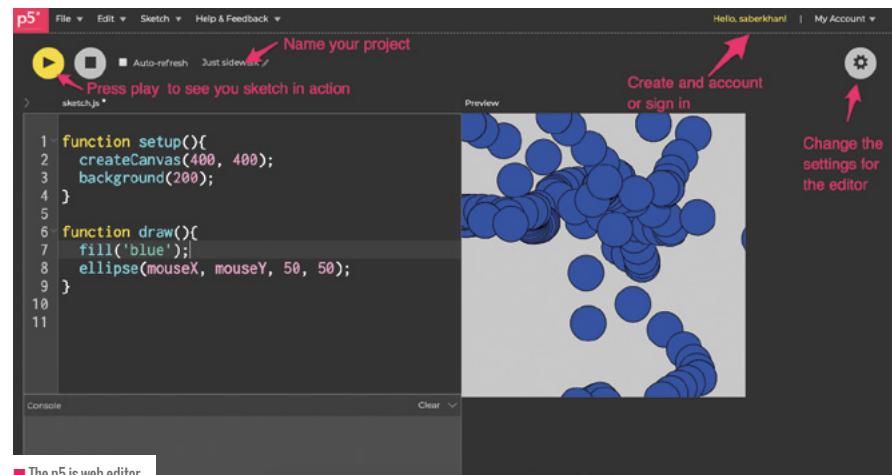
You can read more about ellipses, other shapes, and more on the p5.js reference page: **p5js.org/reference**. There is information on other functions and variables built into p5.js, including `fill()`, `mouseX`, and `mouseY`. The `fill()` command lets the program put in a colour for the shapes using colour names, RGB numbers, or HSB number. And `mouseX` and `mouseY` variables let the program know the x and y position of the mouse, respectively. Let's use them with the ellipse command.

```
fill('blue');
ellipse(mouseX, mouseY, 50);
```

The ellipse is drawn wherever the mouse goes. We can take this idea to build our first program, a painting application. The first question you have to solve is how to

■ The p5.js web editor

preserve the old circles. My hint: `draw()` is a loop from top to the bottom. Once you have figured that out, try to make a palette that lets users pick the colour for the paint. This will probably require using conditional statements, if-then statements, or creating functions.

A simple painting program:

```
function setup(){
  createCanvas(400, 400);
  background(200);
}

function draw(){
  fill('blue');
  if(mousePressed){
  ellipse(mouseX, mouseY, 50,
  50);
  }

}
```

There are many different directions to go in, and online guides available as support. There are various paths available for you and your students, depending on what modality you prefer.

The New York City Department of Education has been collaborating with the Processing Foundation to bring creative coding to their students via p5.js. They have made the curriculum publicly available: **helloworld.cc/p5js**. I can't fit everything into my classes, so I often use it as a guide or borrow ideas from it. Another option is to follow Dan Shiffman's Coding Train videos on YouTube. Dan's energy and joy are infectious. He provides

an accessible playlist of beginner videos (**helloworld.cc/shiffman**). For intermediate or advanced lessons, his videos include topics on machine learning, game development, and more. Beyond these two essential resources, he shares tutorials on Medium, Twitter, and GitHub regularly.

There are a number of artists and coders in the United Kingdom, or close by, who can provide further inspiration, such as Phoenix Perry (**@phoenixperry**), Dave Wythe (**@beesandbombs**), and Saskia Freke (**@sasj_nl**). As the Processing Foundation Education Community Director, I'm committed to helping students and educators get started with p5.js or Processing. Please get in touch if I can help – you can send me an email at **saber@processing.edu**. (HW)

SABER KHAN
Saber (@ed_saber) is a middle and high school Computer Science teacher in Brooklyn, NY, and the Processing Foundation's Education Community Director.

RESOURCES TO GET STARTED WITH p5.jS

- Documentation - **helloworld.cc/p5js_docs**
- Editor - **helloworld.cc/p5jseditor**
- Curriculum - **helloworld.cc/p5js**
- Dan Shiffman (Coding Train) on YouTube for projects and extensions - **helloworld.cc/shiffman**
- Saber Khan's email - **saber@processing.org**

THE BIG IDEA

The Duke of York Inspiring Digital Enterprise Award (iDEA) is a free programme to help learners of all ages develop digital and enterprise skills. **Gemma Moine** explains how she has embedded it in her teaching of computing

I f you have not had chance to investigate iDEA, a new digital award, now is the time to dive in. In 2017, the Duke of York launched the Bronze Award and learners around the globe were quick to rise to the challenge. Bridging the gap between ICT, digital enterprise and computer science, the iDEA Award allows learners of all ages to gain an industry recognised award, and an official certificate, direct from Buckingham Palace!

Initially, we ran the iDEA award as an extra-curricular club at my school. Students who joined the club included sixth-form students wanting to strengthen their UCAS personal statements, Year 6 academic scholars with a keen interest in computer science, as well as a wide range of students from across the school.

The interactive online platform that iDEA uses appealed to our students. After observing how closely the iDEA badge tasks are mapped to the UK Computing National Curriculum, we decided to review our Key Stage 3 (ages 11–14) scheme of work and identify links between units of work and badges. Badges were assigned to our existing units of work and students completed them, either in class or as homework, to consolidate and extend learning. The range of topics covered are vast and would complement a wide range of schemes of work. For example, in Year 7 we were able to link 'e-safety and online etiquette', 'safety online', and 'social media ethics' badges to our digital awareness unit; in Year 8, the 'coding solutions', 'random coding', and 'Python quiz' badges were linked to our Introduction to Python unit.

■ Pupils at The British School Al Khubairat work on iDEA badges

HOW TO GET STARTED WITH iDEA

- ■ iDEA website: **idea.org.uk**
- ■ Twitter: **@idea_award**
- ■ The Bebras Challenge (**bebras.co.uk**) helps learners earn iDEA points

The iDEA badge tasks can align to the UK National Curriculum

WHY RUN THE iDEA AWARD?

There are many benefits of running the iDEA award within a school or educational institute:

- Students achieve an industry-recognised award
- It is completely free and accessible across a wide range of devices, operating systems, and browsers
- It raises the profile of ICT and Computer Science departments, and offers a digital alternative the Duke of Edinburgh Award
- It encourages lifelong learning: it allows for students who have not opted for ICT/Computer Science at GCSE to continue to further their knowledge

- It supports students evidencing their wider learning on UCAS personal statements
- All badges have been mapped against the UK National Curriculum and the Skills Framework for the Information Age. They can be integrated into schemes of work as class activities, homework, or enrichment activities
- It can be used for professional development purposes: many of the badges contain valuable learning opportunities across a wide range of topics potentially covered, such as big data, Internet of Things, cloud computing, Python, and HTML.

How iDEA works

The Duke of York Inspiring Digital Enterprise Award (iDEA) is a global programme that aims to develop digital and enterprise skills. Learners can access a series of free online challenges and earn badges in the process. The award is open to all age groups, and it hopes to encourage lifelong learning. The iDEA Award is currently being run at educational institutes, but it is also accessible to community outreach organisations, and other educators. Learners can select from five main categories: citizen, worker, maker, entrepreneur, and gamer. Each category has its own series of badges:

- Citizen badges support learners through online safety and digital awareness

- Worker badges teach the fundamentals needed to work in the digital workplace

- Maker badges bring out the creative flair in learners as they design, build, and make digital products

- Entrepreneur badges encourage investigation and innovation

- Gamer badges entice learners to explore the world of gamification, design, and build

Each badge is associated with a number of points, and learners need to achieve

> " THE AWARD IS OPEN TO ALL AGE GROUPS, FROM SCHOOL AGE AND BEYOND, AND IT HOPES TO ENCOURAGE LIFELONG LEARNING
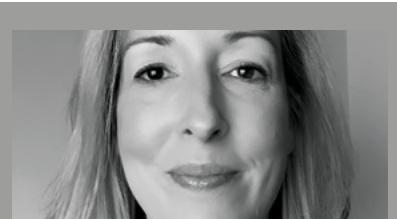
a minimum of 250 points to be awarded the Bronze Award. They can continue on to reach 'Badge Champion Special Achievement' status, if they complete over 50 badges. An additional extra feature is the 'Record of Achievement' document which allows students to download, print, and share their iDEA achievements to date. Once students achieve Bronze level, they can progress on to Silver and the Gold Awards, which will be available soon.

I found it really helpful to upgrade my account to Organiser Status. I could then assign codes to individuals, classes, year groups, or key stages, giving me detailed learner analytics about students' progress. Another added benefit of upgrading to Organiser Status is it allows teachers to access badges as often as needed – meaning I could prepare for my lessons in advance, and demonstrate sections of badges to my classes.

Beyond the regular badges

In addition to the badges available on the website, iDEA collaborates with organisations to accredit activities, workshops and events so that learners can earn iDEA badge points in the process.

Bebras – the annual Oxford University computational thinking and problem-solving event in November – is an excellent example. When learners receive their Bebras certificates, they can use the unique code on the certificate to earn points towards iDEA's Bronze Award. At my school, we will be running a series of robotics workshops for Year 9 that will allow students to earn their very own school-specific iDEA bespoke badge. If you want to propose a activity, workshop, or event that is taking place at your school, contact iDEA and pitch your idea! (HW)

GEMMA MOINE
Gemma (@BSAKComputing) is a Secondary Computer Science teacher at The British School Al Khubairat in Abu Dhabi, United Arab Emirates.

■ Students at The Priory School in Hitchin were set the independent learning challenge of 'creating their own environment in Virtual Reality', with the aim to test their final creation on the HTC Vive

VIRTUAL REALITY: LEARNING FOR THE FUTURE

Investing in high-end professional virtual reality hardware can be expensive, but **Sam Hankin** presents the benefits of immersing students in this new technology

We aim to prepare our students to live in a changing digital world: we teach a range of skills and tools which we hope they will use every day and into their future. Virtual reality (VR) allows you to experience near real-life 'reality' by immersing your visual and audio senses in a 'virtual' 3D-generated environment. While you're within this virtual world, the environment will provide appropriate responses to your actions, to enable you to feel like you're really there! In the future, VR could be a tool that's used, not just in gaming, but embedded in a range of industries to support workforces. There's already real examples of VR being used in medicine, construction, retail, and the military.

In computing lessons, we design algorithms and write code to complete specific tasks, such as a tax calculator in Python, a script to run a Sphero around the room, or a game interface to control a character's actions. But we also need to let students experiment with the hardware they may come across and have to program in the future, which is why at my school we decided to invest in an HTC Vive virtual reality headset.

The hardware

The most crucial part of creating this immersive environment is the hardware. Lots of companies currently offer different levels of 'VR' experiences. Google Cardboard and Samsung Gear VR both work with your smartphone device and are a great way to experience an introduction to VR for under £40.

The Oculus Go (£200) makes it easy to experience another level of virtual reality. It uses an all-in-one system that doesn't require a smartphone or computer and has access to its own store of over 1000 games and social apps. For education, ClassVR bundles together similar standalone VR headsets into class sets to allow students to experience VR environments in their lessons, and gives you access to a bank of curriculum content to use.

But the current market leaders for the top-end hardware, and consequently the most immersive VR experiences, are the HTC Vive (£500) and the Facebook-owned Oculus Rift (£349). Both of these headsets have high-resolution OLED displays in each eye, a 110-degree wide field of view, and a refresh rate of 90 Hz. They both also use

■ We created a 'Virtual Reality Corner' putting some cheap foam padding on the walls and a piece of artificial grass on the floor

■ Our A-level and GCSE Computer Science students were very keen to help in the process of setting up the HTC Vive

room-scale tracking technology to allow you to move in the 3D space, and motion-tracked handheld controllers to interact with your virtual environment. After extensive research on the two systems, we opted for the HTC Vive. It has based more in-depth tracking, we preferred the style of its handheld controllers, and it already came bundled with room sensors. The HTC Vive is also being used by industry developers when designing and producing VR content.

The headsets both require you to be tethered to a very powerful PC rig – unfortunately, standard classroom computers won't cut it! The actual specifications required differ for headsets, so for our HTC Vive we custom-built a PC for £735.

All in, this project to equip our classroom with a virtual reality headset came to approximately £1200. Through a successful online crowdfunding campaign on Rocket Fund (**rocket.fund**) and writing to some local businesses, we managed to fully fund our VR project independently from our school budget.

Independent learning

Recently, we have been working on a number of projects that require students to lead their own learning. We encourage students to make their own choices, and

we try to teach them to learn how to use the computer through experimenting and finding their own instructions, instead of leading them step-by-step. We need to prepare students for the fast pace of technological change. They won't have someone standing over them telling them how to use the newest type of computer in ten years' time, so it is important students learn these vital digital literacy skills to enable them to succeed in the future. I believe that because the VR technology is

school, which is great for building positive student-teacher relationships!

We're only at our first stages of using the virtual reality equipment for development, but our plan for the future is to enable more students to experiment with their own projects in the same way these Year 10s did. Our A-level Computer Science students also now have the opportunity to develop their Non-Examined Assessment projects in virtual reality, and the Year 12s have already started to come up with some exciting projects. (HW)

> **" ALL IN, THIS PROJECT TO EQUIP OUR CLASSROOM WITH A VIRTUAL REALITY HEADSET CAME TO APPROXIMATELY £1200**

still relativity new, students are more willing to experiment. This increases their technical knowledge and builds their resilience of how to survive in a changing digital world.

In our KS3 after-school coding club, we're giving students a proper VR experience for the first time, which will hopefully inspire them to make their own content and learn more about the specifications of the hardware required to run the games they play. And even if some of them aren't thinking 'technically' and just want to play Beat Saber in school, they're at least enthusiastically coming back to the department during lunchtimes and after

SAM HANKIN
Sam (@hankin) is a Head of Department for Computing at The Priory School in Hitchin. He is a CAS Master Teacher, Specialist Leader of Education, and Google Educator.

COMPUTERCRAFT: PROGRAMMABLE DEVICES WITHIN MINECRAFT

Graham Bridge argues that the open-source modification for Minecraft is a perfect way to get students started with text-based programming

ComputerCraft is a popular modification (or 'mod') for Minecraft that was written in 2011 by the British games developer Daniel Ratcliffe. It is downloadable, free, and open-source.

ComputerCraft adds computer hardware devices to the game. Handheld tablets, desktop computers, turtle robots, disk drives and disks, modems, networking cables, and scalable monitors can all interact with the Minecraft world and can be programmed by players.

ComputerCraft turtles

The turtle is one of the most exciting devices in ComputerCraft. These are one-block robots that can be equipped with tools such as crafting tables, pickaxes, swords, and modems. They can move in any direction, are immune to water and lava, and can destroy and place blocks. This means they can build any structure or mine underground in the search for diamonds. If fitted with a crafting table, they can make any item available in the game.

Players can program ComputerCraft's turtles with Lua – a scripting language which is lightweight and fast, but surprisingly powerful.

Lua is perfect for students just getting started with text-based languages. It doesn't include Python's confusing white space requirements and every function, if statement or loop has a matching 'end' keyword.

You can print an initial message on screen using:

```
print("Hello World")
```

Turtles have a set of built-in methods and properties which are easily mastered. For example, to move the turtle forward, use:

```
turtle.forward()
```

This will not work immediately, as turtles require fuel, which can be obtained from logs, planks, wooden items, coal, or lava. Every movement away from its current position uses one fuel unit. Rotation, tool use, and crafting do not use fuel.

A suggested first task for new Lua coders could be:

- Start with a Turtle and four planks in slot 1 of the inventory.

- Refuel using just one of the planks.

- Get the turtle to move forward ten blocks.

```
turtle.select(1)
-- select slot 1
turtle.refuel(1)
-- refuel using 1 item
```

Given the use of parameters in the functions above, the logical line to use next is:

```
turtle.forward(10)
```


■ Monitors and desktop computers are key additions to Minecraft from ComputerCraft. The ComputerCraftEdu interface is overlaid on the left

Unfortunately, this last command would not work. The parameter is ignored, and it moves just once.

Lua functions

Rather than just using a loop within the script to move five times, you can introduce functions straight away:

```
function forward(blocks)
    blocks = blocks or 1
-- in case number parameter
-- supplied
    for n = 1, blocks do
-- repeat number of times
-- supplied by blocks
        turtle.forward()
    end
end
```

As with Python, functions must be written at the top of the script and can then be called.

So, our complete script following the above function is:

```
turtle.select(1)
-- select slot 1
turtle.refuel(1)
-- refuel using 1 item
forward(10)
-- call the forward
-- function to move 10 blocks
```

On flat ground, the above script works perfectly, but what if there are blocks in the way? Sand and gravel blocks are another potential pitfall: they drop if their supporting block is removed, so a stack of sand or gravel will fall if the bottom block is excavated with `turtle.dig()`.

The function needs to be modified to take account of blocks in the way, and the possibility of sand or gravel:

```
function forward(blocks)
  blocks = blocks or 1
  for n = 1, blocks do
   if not turtle.forward() then
     while turtle.dig() do
       sleep(0.5)
     end
    end
   end
end
```

GET STARTED WITH COMPUTERCRAFT

ComputerCraft requires the full Java edition of Minecraft - either version 1.7.10 or 1.8.9 - and a personal Mojang Account. It will run on Windows, macOS, or Linux.

■ ComputerCraft: computercraft.info
■ ComputerCraftEdu: computercraftedu.com
■ The Java edition of Minecraft: helloworld.cc/minecraft

■ The ComputerCraftEdu block-based `forward()` function, and its underlying Lua code

> " PAPER MESSAGES CAN BE PRINTED FROM THE PRINTER AND PLACED IN CHESTS FOR PLAYERS TO READ

Monitors and other peripherals

Another useful device found in ComputerCraft is the monitor block. When several of these are placed on the same plane, they form a single monitor. They are useful for displaying information, showing a program, or even the status of an IndustrialCraft2 nuclear reactor.

GRAHAM BRIDGE
Graham is a STEM Ambassador and Code Club volunteer. He earned a teaching certificate from BCS last year and has put that to use teaching C# to Year 12.

Players can also carry floppy disks which can be used to transfer programs or data between computers using the disk drive block. Paper messages can be printed from the printer and placed in chests for players to read. These are useful for instructions!

ComputerCraftEdu

ComputerCraftEdu is a related, but separate, modification for Minecraft. It uses block-based programming (not to be confused with Minecraft blocks!), and includes turtles that do not need fuel.

Both ComputerCraft and ComputerCraftEdu are free to download. They provide a captivating way to learn computational thinking inside Minecraft. (HW)

COLLECTIVE WISDOM: HOW TO START TEACHING COMPUTING

Taking your first steps into the world of teaching computing? **Neil Rickus** draws on the advice of the online education community for guidance on how to get started

Many teachers are new to delivering the computing curriculum in their schools and are often unsure where to begin. Using the views of the computing community on social media (Twitter handles are listed in brackets to identify contributors), we'll examine a few topics to consider when beginning your journey into teaching this subject for the first time.

You are not alone

Firstly, there is an extremely supportive community of computing educators online, who are able to offer impartial advice on everything from introductory activities for KS1, to how to reinforce A-level students' understanding of more advanced topics (**@Lenandlar**). On Facebook, two groups are particularly popular. The 'Primary Computing Coordinators UK' group has over 2500 members, and the 'ICT & Computing Teachers' group has over 4000 members. A number of other groups exist, including those for specific programming environments / languages (Scratch, Python, etc.), technologies, and examination boards.

In the UK, Tuesday at 8pm is the time for #CASChat on Twitter, which sees members of the education community come together to discuss various topics relevant to computing in the classroom. The discussion, which is organised by Simon Johnson (**@clcsimon**), can be used to ask questions, share knowledge, and learn from experts in the field, with regular contributors including Miles Berry (**@mberry**), Alan O'Donohoe (**@teknoteacher**), and Phil Bagge (**@Baggiepr**). Many other countries have similar Twitter-based discussions, such as the US-based #CSK8.

The Computing At School (CAS) online forums (**helloworld.cc/cas**) continue to be a useful platform for more detailed discussions, while CAS Communities of Practice (previously CAS Hubs) enable teachers to meet face to face on a regular, informal basis.

How to develop and choose appropriate resources

The resources you develop or choose need to include appropriate pedagogy to ensure learners communicate effectively and to develop the knowledge of students at different levels (**@Vanderpere**). Sessions should be enjoyable (**@MrBakerUK**), relevant to pupils' interests (**@Moorlag**), and promote creativity (**@SwayGrantham**). Effective links should also be made to technology familiar to children (**@chrisnash01**), such as step counters and technology-related news events (**@connoracton**).

For those new to the subject, teachers' understanding will develop alongside that of their pupils (**@amorimtweets**) and additional support could be provided in the form of Digital Leaders (**@ukstemdotuk**) and online tutorials.

Finally, teachers need to determine appropriate progression in the subject by ensuring pupils are examining more challenging concepts as they move through the school (**@computingmissa**).

Do you have any other advice for this starting to teach computing? Have you successfully implemented any of these strategies? Let Neil (**@computingchamps**) and the *Hello World* team (**@HelloWorld_Edu**) know via Twitter. **(HW)**

GETTING STARTED ADVICE
CAS MASTER TEACHER, CAT LAMIN – @CatLamin

Take the time to experiment with resources before deciding they are amazing or rubbish. Work through resources as though you were a student, so that you can understand how the students in your class will appreciate them. Don't just buy the most flashy or loudest or brightest-coloured piece of hardware - explore and play with it yourself.

NEIL RICKUS

Neil (@computingchamps) is a Senior Learning Manager at the Raspberry Pi Foundation and a Senior Lecturer in Computing Education at the University of Hertfordshire. He is a CAS Community Leader and a Raspberry Pi, Google, and Microsoft Certified Educator.

MAKE NETWORKS INTERESTING WITH FILIUS
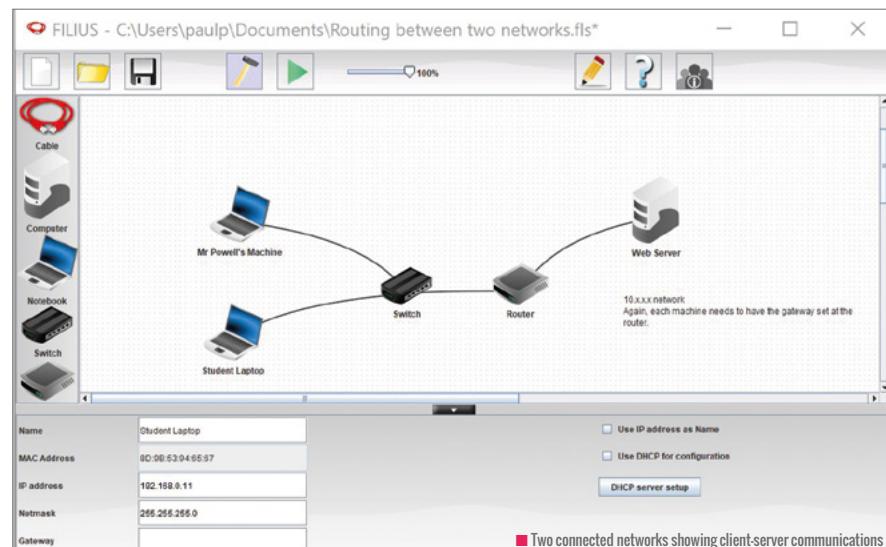
Paul Powell shares how you can liven up your lessons on networking

N etworking can be dull. Lots of terminology, acronyms, technical detail, and bits of binary. Apart from the underlying sense that it makes YouTube work, what exactly is the point of knowing all this stuff? Some schools are lucky enough to be able to make a network of Pis and get them talking to each other, but not every department will have the funds, the space, or the wide range of skills needed.

Filius to the Rescue

Trying to get around this issue, I started looking for a network simulator. Everything I found was either too complex (GNS3 or Packet Tracer) or too restrictive (the Teach ICT simulator – OK for KS3.) Eventually I stumbled on Filius (**helloworld.cc/filius**), a good midpoint between the two. Filius is an open-source Java app that was written first in German, but then translated into English. There is a decent English guide on the website.

Filius lets you set up a network with computers, switches, routers, and cables. At the most basic level, this can be used to connect the basic components. Each machine can be set up with an IP address, and then the simulation can be run. When in simulation mode, you can install software onto the machines. You can use this to ping between machines and you can see the packets going back and forth as green pulses along the wires, or by right-clicking

Two connected networks showing client-server communications

any computer or router and inspecting the packets as they go back and forth.

Once the basics of a LAN are out of the way, Filius then lets you set up multiple networks, routing tables, web servers and web browsers, email, DHCP, and more. This might all sound a little daunting, especially with classes that are a little less attentive. Fortunately, Filius lets us load and save our networks. For my classes, I prepared networks in advance in order to teach specific concepts. These are now available on the CAS website.

Making it accessible

Adapting the concept of the PRIMM approach, I tried to structure activities to read before writing. Typically, this meant giving students a network already set up with an element of the network working. Students need to predict what the network will do (mostly based on machine names and topology), simulate (run) the network and carry out a few tasks, investigate the settings, modify the settings of a non-

working portion to get it running, and then make a new section of the network.

I had success with this approach with my Year 11 class (ages 15–16). Everyone was engaged and it really helped explain the different forms of addressing. Next year, I'll be breaking the tasks down further so that those that are lower ability can work more independently. **(HW)**

PAUL'S NETWORKS

I have uploaded several demonstration networks to the CAS website for you to have a play with - helloworld.cc/casfilius

PAUL POWELL
Paul leads the Waltham Forest CAS Community of Practice and is Curriculum Lead for Computing at George Mitchell School.

EMMA BENNETT HEAD OF COMPUTING

DELIVERING COMPUTING WITH DIFFERENTIATION

Emma Bennett argues that many of the challenges of teaching computing to pupils with special needs are relevant to all learners, as are the strategies that deal with them

Since taking up my first SEND specialist position in 2014, I have developed strategies which I wish I'd thought to use back in my mainstream days. I believe that some of the biggest barriers to success in computing I've encountered at my special school are prevalent across mainstream schools too. After all, learners in specialist settings are not a different species, and not every mainstream pupil in need of extra support has an Education, Health and Care Plan, or a Statement of Special Educational Needs.

Low levels of achievement, engagement, and aspiration in computing are not necessarily caused by quality of teaching or learners' personal capabilities – one of the key issues is access to computing experience and appropriate resources.

Learners come to secondary school with wildly varying levels of computing experience. For a lucky few, their tech-savvy parents and teachers have given them a grounding in coding, computational thinking, digital literacy, and the rudiments of digital communication. Many others have had limited access to opportunities to develop the broad range of knowledge, skills, and understanding required by the curriculum:

- Some learners might have been removed from computing lessons for extra literacy or numeracy work.

- Some learners' primary schools might focus exclusively on coding; some are still teaching only ICT.
- Other learners will not have access to suitable technology at home.

These huge variations in prior knowledge, skills, support, and access to technology are often not reflected in students' baseline data. Whether you are in a special or mainstream setting, this presents an enormous challenge that is not captured by the data-tracking systems in place.

What's more, the UK computing curriculum is incredibly broad, covering a wide range of topics, from modelling data to the safe use of technology. Even in high-achieving mainstream settings, not all of these subjects will coincide with the skills and interests of every learner (or indeed teacher!). This can lead to disengagement or learners even resisting to take part in lessons.

Do you feel special(ist)?

Computing as a whole continues to fall in both 'academic' and 'vocational' camps, with GCSE Computer Science counting towards the English Baccalaureate, while equally challenging qualifications drawing on the other strands we work so hard to cover at KS3 are being unfairly relegated into 'Mickey Mouse' territory.

It's no surprise that, with the dearth of appropriate qualifications to offer at KS4, the other computing strands are being downgraded at earlier Key Stages in a rush to focus on GCSE content and knowledge acquisition for exam questions. Unfortunately, this can leave students with SEND and students in households with low levels of academic or technological engagement feeling excluded, and may have a negative impact on the well-being of teachers without a computer science background who have had to catch up with the 'real' specialists.

" SEEK OUT OPPORTUNITIES TO COLLABORATE

Luckily, there are plenty of opportunities to upskill, connect with other teachers, share resources, build, and share workarounds, thanks to organisations like Computing at School, the Raspberry Pi Foundation, and the National Centre for Computing Education.

Differentiating without dumbing down

There simply aren't many resources or opportunities specific to teachers of computing with pupils with special needs. Many computing learning sites, including e-safety providers, mention that their resources 'can be adapted', or recommend using primary-level resources with learners at secondary level. Imagine the reaction if teachers of other secondary subjects with steep learning curves were told to do this!

There are a few exceptions: Sheffield eLearning Service (**sheffieldclc.net**) is leading the way with research and resources and, if you are lucky enough to work in an area with more than one specialist provider, you might get invited to network meetings, but as far as I can tell that's it.

So how do we make sure that our learners can access material appropriate to their skill set and level of social maturity? Here are eight quick tips to help you get started.

EMMA BENNETT

Emma (@anglonemi) is Head of Computing at Oakwood Academy, Salford, a special school for pupils aged 9-18 years who have a range of moderate or complex learning difficulties. She has 15 years' experience of working in education and moved to SEND specialist provision from in 2014.

KEY STRATEGIES

Keep a checklist of your pupils' needs and relevant strategies to help you find or generate appropriate resources.

For example:
- poor working memory - short steps
- low literacy - simple language
- low numeracy - basic operations
- visual impairment - high contrast or large format
- physical disability - adapted equipment
- hearing impairment - visual outputs
- neurodiversity conditions - literal language, manageable sensory stimulus

Even something as simple as putting a painted box over the workstation of a pupil with ADHD to help them focus can have a huge impact - my student D calls it his 'office'.

1 Seek out opportunities to collaborate/share best practice
2 Take time to identify learners' strengths, interests, and aspirations
3 Keep expectations high – if work is too hard, learners will tell you. If it's too easy, you run the risk of losing even the most able and engaged
3 Remember that schemes of work are not NASA checklists – nothing is going to explode if you don't follow it exactly
4 Lobby school data managers for accurate, subject-specific baselining
5 Use a tool that allows the tracking of robust data and target setting for students with Special Educational Needs, such as CASPA (**caspaonline.co.uk**)
6 Use ICT-based work from other subjects to help you assess IT and Digital Literacy
7 Cross-curricular projects enhance engagement and help raise the profile of Computing as a subject
8 Use work-related or popular culture contexts – this helps students see the relevance of what they're working on and hopefully captures their attention. (HW)

■ Alan Turing conceived the idea of an early computing machine, as well as the Bombe machine used to decipher messages in WWII
Image: mendhak [CC BY-SA 2.0]

PROFESSIONAL DEVELOPMENT FROM CAS TENDERFOOT

John Woollard and **Roger Davies** introduce resource-rich professional development materials, designed to support teachers who are new to the computer science aspects of the computing curriculum

CAS Tenderfoot is a Google-funded project designed to equip Key Stage 3 (ages 11-14) teachers with professional development materials. The topics are selected on the basis that they offer understanding in key areas of computer science: programming, algorithms, data structures, logic, hardware (from chips to computers), and modelling systems.

A key feature of CAS Tenderfoot is the classroom-based teaching materials. The resources not only show how computer systems work, but show how to teach how computer systems work.

A good example is the Turing machine. If a pupil asks, "How does a Turing machine work?", it gives great confidence to the computer science teacher if they can:

- say what it is
- describe it in everyday terms
- give the pupil an interesting, chocolate-based, activity to show one in action

The final parts of the CAS Tenderfoot Unit 'Theoretical Computers: Fun with Finite State Machines' enables all of that. Previous parts of the unit deal with finite state machines and regular expressions – important concepts in making systems computable. The resource explains

"*Alan Turing conceived the idea of computing machine. He based it on the idea of a human 'computer' – someone who did calculations in a series of steps, using a scratch pad to jot down parts of the working out. It is remarkable for two reasons. First, he had the idea long before any electronic computers had ever been built. Second, his model has been shown to be as powerful as any computer built since! By powerful, we don't mean fast, we mean it can compute anything any modern computer can compute.*"

Teachers are introduced to the Chocaholic Turing Machine from CS4FN (**helloworld.cc/chocolateturing**). This consists of lots of milk chocolates, lots of dark chocolates, six coloured lollies (which act as

CAS Tenderfoot

Supported by Google for Education

'state' flags), and a computer program. After 20 minutes of fun, including a little eating, all computer science teachers can confidently respond to the question, "How does a Turing computer work?" CS4FN Chocaholic is a freely distributed resource, and our thanks go to Paul Curzon and his colleagues for putting high quality teaching resources into the hands of our computer science teachers.

For many, computer programming is at the heart of computer science. When considering complex systems and how to make them computable, the computer program is

the minimum number?' Many learners are motivated to find the actual number, but the computational challenge is to find a way of finding that number which works for all muddy cities, school networks or even home DIY projects!

Another mapping challenge in the CAS Tenderfoot materials is the Bacon number. What is the minimum number of steps to cross a network, for example, the internet?

Twenty years ago, inspired by the theory that nobody is more than six relationships away from any other person in the world, the movie fan Brian Turtle devised a game called Six Degrees of Kevin Bacon. The game involves trying to connect any movie star to the actor Kevin Bacon, linking them by other actors who have appeared in the same film.

For example, Alexandra Daddario was in *The Squid and the Whale* (2005) with Jesse Eisenberg, who was in *Beyond All Boundaries* (2009) with Kevin Bacon, therefore Alexandra

simple low-level languages to run in a child-friendly simulation of the Little Man Computer, a well-established model for visualising program execution. Many resources in the CAS Tenderfoot project were written by CAS members and originally shared on the CAS Community. In this particular unit, two experienced teachers share different logical challenges that build up in complexity, ending with half and full adders – the basis of all hardware computation.

CAS Tenderfoot has in total seven full days of CPD materials for training Key Stage 3 teachers. The materials are divided into sessions which can be delivered in approximately two hours. All the presentations, handouts, videos, and online resources are organised into the topics of:

Unit 1: Laying Firm Foundations: A Conceptual Approach to Programming
Unit 2: How Computers Do Stuff: Algorithms + Data Structures = Programs
Unit 3: Clever Stuff for Common Problems: Going Beyond Simple Algorithms
Unit 4: Bits and Bytes: The Digital Advantage
Unit 5: Bits and Chips: The Simple Ideas That Make Computers Tick
Unit 6: Theoretical Computers: Fun with Finite State Machines
Unit 7: Simulating Our World: Adventures in Agent Based Modelling

All teachers, worldwide, have access to the CAS Tenderfoot website at **helloworld.cc/tenderfoot**. Each unit has training notes and an introductory video. (HW)

" HOW DO YOU JOIN ALL THE POINTS ON A NETWORK WITH A MINIMUM NUMBER OF CONNECTIONS?

the ultimate abstraction. However, within computer programs there are important constructs. The CAS Tenderfoot materials introduce teachers to: sequence, repetition and selection; data structures including lists, stacks and queues, trees, maps. A challenge when teaching computer science is persuading the pupils of the value and relevance of what they are being taught. CAS Tenderfoot tackles this issue head-on by illustrating all the concepts through practical activities – it promotes the 'unplugged' computing curriculum.

One example is the complex issue of 'minimum spanning': how do you join all the points on a network with a minimum number of connections? CAS Tenderfoot uses the CS Unplugged challenge 'Muddy City'. CS Unplugged has a wide range of classroom activities that supports teaching and learning of the computer science concepts (**csunplugged.org**).

The problem can be given to five-year-olds and graduates alike to start the computational thinking: 'how do we go about finding

Daddario has a *Bacon number* of 2.

There are a number of online resources to extend this activity and bring a better understanding of the algorithms behind everyday applications, such as satnavs and route-finding software. The CAS Tenderfoot worksheet shows how, in a complex network, the visually shortest route need not be the actual shortest, and then enables the teachers to use an algorithm to definitively identify shortest routes across maps.

Form 0s and 1s to programs

One tricky area of computer science is the connection between binary logic and the hardware inside a computer. The 'Bits and Chips' unit from CAS Tenderfoot makes those connections. It first introduces the behaviour of the three logical constructs of AND, OR, and NOT, then truth tables, some combinational logic, and symbolic representation of circuits. Further gates are then introduced (NAND, NOR, and XOR).

Through a number of games and activities, the ideas build up to writing programs in

JOHN WOOLLARD & ROGER DAVIES

John Woollard is a teacher trainer, author, and university lecturer working in both computer science and special educational needs. He is a member of the BCS School Curriculum and Assessment Committee, and chair of CAS Assessment working group.

Roger Davies is a teaching resource author and the former (now-retired) director of IT at Queen Elizabeth School in Cumbria. He is a CAS Master Teacher and previously edited the CAS termly magazine SwitchedOn.

READ BEFORE YOU WRITE

Will Grey shares some evidence-based approaches for helping pupils to read code

Before students can write code, they need to be able to read code. Computer science pedagogy is often based around the ideas of Piaget's constructivism – where pupils develop their knowledge through exploration, and Papert's constructionism – where pupils learn through creating artefacts. However, learners need guidance to gain useful knowledge efficiently, and to organise that knowledge in a clear and logical way. They need to be able to break a problem down, remove the unnecessary detail, find patterns, and think algorithmically before they can start to write programs for solving problems.

Just as we wouldn't expect a young child to write prose before they can read, we need to provide guided approaches that use direct instruction and scaffolding to help our students read code before they can be expected to write code themselves. These guided approaches are needed just as much as, if not more than, creative discovery activities.

Explain the code

My first approach to improving code comprehension is to ask my pupils to 'explain in plain English' what a piece of code does. There are many variations on this activity, but I find that this works well when pupils explain to each other what the code does. That way, pupils have valuable contributions from others that they can then incorporate into their own written explanations.

We could take this further and ask students to carry out various explorations of the code. For instance, pupils could annotate or add comments to code, list and explain the purpose of the variables and functions, and create structure diagrams. I particularly enjoy doing this with the AQA A-level pre-release material – a reasonably substantial piece of code that enables a rich investigatory experience.

Worked examples

Another approach is to use worked examples. They can be delivered by modelling with live coding, or using tutorial guides. During a live coding demonstration, the teacher explains each step of the code as it is being written. This is beneficial because students can see how teachers tackle problems. Pupils also see that making coding errors is a normal part of the

FURTHER READING

✓ **Lopez et al. (2008)**
Relationships between reading, tracing, and writing skills in introductory programming, *Proceedings of the Fourth International Workshop on Computing Education Research*
`helloworld.cc/lopez2008`

In this issue
Cognitive load - page 34
Parson's puzzles - page 87

process. Another benefit of live coding is that the pace of delivery is generally slow and this may help some groups of pupils – I have found it particularly helpful with underachieving boys.

Nevertheless, demonstrations need to be kept short and only progress a few steps at a time, so as not to overload working memory. Lessons should have several shorter demonstrations spaced throughout, rather than fewer longer demonstrations. Perhaps live coding demonstrations could be pre-recorded as screencasts with audio and made available to pupils to watch, pause, and replay in their own time, thereby helping pupils who did not follow the first time round. I tend to couple the demonstrations with tutorial guides because it gives my pupils greater autonomy. I have used worked examples across the age ranges to deliver lessons on Scratch to Year 7 pupils, all the way through to advanced concepts in Python like server-side scripting with sixth-form students.

Sub-goals

The third approach is to group individual steps under meaningful sub-goal labels. In addition to helping students organise and structure the code in a coherent manner, this reduces the load on working memory because individual steps are chunked together. Pupils could be asked to do this themselves, or sub-goal labels could be provided by the teacher, according the specific needs of the student.

Trace the code

The fourth approach is for pupils to trace code by hand to simulate the outcome of each instruction in an algorithm. This is where the student replicates the task of the

An example of a Parson's puzzle: can you draw a square?

computer by keeping track and recording the values of the variables at each step, and performing the various arithmetic and logical operations. Pupils find tracing code tricky as it requires considerable patience and concentration to not lose their train-of-thought. It takes practice to master, especially when tracing recursive algorithms like merge-sort. I tend to practise these

regularly with my GCSE and A-level groups – both with the familiar algorithms for searching, sorting, and traversing, and with algorithms that they have not seen before. There are plenty of past paper questions across the exam boards that can be used support this activity.

Parson's puzzles

Parson's puzzles bridge the gap between reading and writing code. Here, code is presented to pupils jumbled up. The task is to put the instructions into the correct order so that the whole code performs a predefined task. Students do not need to be concerned with remembering syntax, and can focus on logic and sequencing. They make a lovely starter activity, and can come in a variety of forms, from those on

a computer, to paper-based and jigsaw-type puzzles with snippets of code that can be pieced together, but this does require quite a bit of preparation.

Putting it all together

Once pupils have gained comprehension of the various programming constructs and seen them applied in different contexts, they will then be able to write better code. There are many scaffolded techniques that can help students write code by taking away the challenges of code design and abstraction.

These ideas for reading code are generally very simple to implement, and can be used together and alongside methods for writing code and more general teaching pedagogies, such as flipped learning, direct instruction, questioning, and verbal feedback, to deliver effective lessons on coding in visual and textual languages to schoolchildren of all ages. **(HW)**

> " THERE ARE MANY SCAFFOLDED TECHNIQUES THAT CAN HELP STUDENTS WRITE CODE

WILL GREY
Will is Head of Computing at Comberton Village College, and coordinator of the CAS South Cambridgeshire Community of Practice. He has developed a comprehensive set of resources for A-level and GCSE computer science at
helloworld.cc/grey

DIFFERENTIATION TO STRETCH AND CHALLENGE

How can we ensure that high achievers reach their potential?
Kirstin Lindop shares her ideas

However you choose to define them, it can be easy to overlook high achievers in a busy classroom environment. Often surpassing their targets, these students are not always made a top priority. Yet high achievers need opportunities to develop their skills and to be challenged like their peers.

I have had the privilege of teaching computer science to some highly intelligent students in the sixth form (ages 16–18). I am not ashamed to say that their programming knowledge way surpasses that of my own. Rather than being threatened by this, I have always embraced and celebrated it. I would like to share a range of strategies for meeting the needs of my highest-achieving students from across the school. Most of these aren't onerous or too time-consuming, but have allowed me to make differentiation for my highest-achieving students a staple in my planning.

Identify the students who need a challenge

Identifying this cohort is the first challenge, as what constitutes as a high achiever in one subject may not be the same in another. At the start of secondary school in the UK, students arrive with little or no prior attainment data in computing. Talking to colleagues to understand your new cohort, as well as carefully observing students in lessons, is the way forward.

Higher up the school, exam target data is an obvious method of seeing who your 'brighter' students are, but this is not always a conclusive method of identifying high-achieving students. Students develop at different paces, and sometimes students can shine beyond their expected grades when given the opportunity and encouragement to do so.

Establishing a positive climate for learning

A well-thought-out seating plan can be a strong tool for differentiation. Often, high-achieving students are seated next to students who need more support. But I'd recommend trying to sit high achievers together. This introduces an element of competition. They can push one another to achieve more in lessons, and become a high-attaining network of support for each other.

At the start of the school year, I provide my exam classes with a folder containing a

CHALLENGING STUDENTS BEYOND THE CLASSROOM

- **FutureLearn has a wealth of free online courses on topics such as object-oriented programming (helloworld. cc/oop). Many of these are aimed at educators, but there is nothing stopping students benefiting from them too**
- **The Extended Project Qualification gives students the opportunity to carry out independent research on a topic of their choice. It is great preparation for university, and students earn a qualification equivalent to half an A level (helloworld.cc/epq)**
- **You could provide students with a reading list of books and articles**

copy of the specification, grade descriptors, a glossary, topic checklists to rate, and a small section of past papers and mark schemes. High attainers have often already developed their own strategies for making effective use of these resources and welcome them. I always include a snip of the relevant part of the specification at the top of any resources, so the students are fully aware of what the exam board requires them to be able to do.

Approaches for every class

Open-ended programming activities can be simple to set and often easy to assess, simply by running them. I have found these independently-led activities, where more challenging tasks are available, have encouraged my students' creativity and imaginative thinking.

I also like to introduce a competitive element in to some of my lessons. The Gamble (**helloworld.cc/gamble**) is a free resource, where students can wager varying numbers of 'chips' depending on how certain they are of their answers. This is a great approach for testing simple questions that students often get wrong, and I have had successes with including tougher questions too.

Giving students the freedom to research a topic prior to a lesson, either with or without supporting resources, is a straightforward way to stretch them. I often combine this with student-led lessons, where students teach the rest of the class. I try to make sure the more demanding elements are given to the students who need to be stretched the most.

Strategies for exam success

It's impossible to assume that, just because a student is intelligent, they are familiar with how to translate this into brilliant exam grades. Frequent timed exam practice is crucial. I don't often devote entire lessons to exam papers, but instead pick a challenging ten-mark question, give my students ten minutes to answer it in class, and then go through the answers together.

Modelling A* work with students can help them to know what they need to aspire to from the start. We also provide students with high, medium, and low attaining exemplar work and ask them to mark it using a mark scheme. They soon come to realise that quality, not quantity, is essential. It hones their exam technique.

two full days of revision of challenging topics. Giving up a weekend and preparing all the resources for this was definitely worth it. Students were fully immersed in the subject for 48 hours which, if their weekends are anything like mine were as a teenager, they normally won't be!

Finally, I'd really recommend becoming an examiner for the exam board that you teach. It equips you with the tools that you need to be able to push students towards the highest grades. I have also found that the networking I was able to do during my examiner training has enabled me to provide students with enrichment opportunities beyond the curriculum.

When writing this article, I asked one of my highest-achieving students his thoughts on school. He said that in some subjects he was simply "enduring the tedium" of easy tasks until he could get to the more challenging ones if he was "allowed"! To me, this really highlights the importance of avoiding an approach where high-achieving students do more of the same, less challenging activities. In supporting all of our students, we can't forget the ones who need the most stretching, too. **(HW)**

> ## GIVING STUDENTS THE FREEDOM TO RESEARCH A TOPIC PRIOR TO A LESSON IS A STRAIGHTFORWARD WAY TO STRETCH THEM

At the end of every topic, as a class we spend a short amount of time looking at the specification and writing out a revision card. The content is fresh, they know the important facts and, by the end of the year, students have a full complement of revision cards already written.

At my school, we deliver AQA at A level (ages 16–18) and, to support the teaching of this course, I subscribed to Exam Pro. From the start of the course, the questions we use in class are those that students could encounter in their exam, so they know exactly what they could be asked and in which ways. This exam focus tunes them into the important parts of each topic covered.
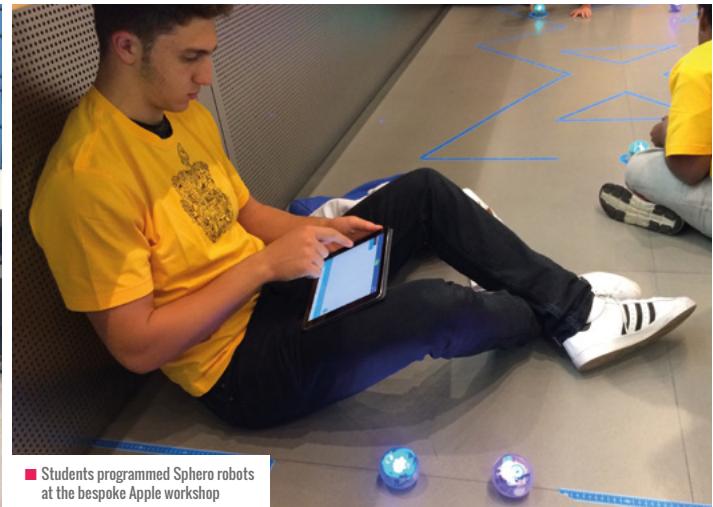
We have taken high-achieving students on a weekend residential trip comprising

KIRSTIN LINDOP
Kirstin (@HelsbyComputing) is Departmental Leader for Computing, Business and Economics at Helsby High School in Cheshire. She is an examiner, and has an interest in promoting STEM and collaborating with other schools.

Students enjoyed working collaboratively to solve algorithm challenges at Intel's education workshop

Students programmed Sphero robots at the bespoke Apple workshop

SILICON VALLEY: THE ULTIMATE COMPUTER SCIENCE TRIP!

Renowned as the birthplace of the digital revolution, Silicon Valley hosts a feast of computer science delights, making San Francisco an ultimate trip destination. **Gemma Moine** shares her experiences of leading a visit

Planning a computer science trip can be daunting, especially when you plan to visit a country thousands of miles away. At The British School Al Khubairat, we were able to take 32 students to California, and I would like to share the exciting itinerary with you, to make the process of pulling together a trip easier.

THE ITINERARY

Day 1: Take an evening open-top Big Bus tour

We arrived late afternoon and, after the students had unpacked, we all jumped on the hop-on, hop-off Big Bus tour that visited a variety of interesting locations.

Cost for the day: Big Bus ticket – $46.80 (10% discount online). Includes 1 day hop-on, hop-off bus ticket, free walking tour, free bike rental (can use to cycle across Golden Gate Bridge), and one night bus tour or Sausalito tour.

Day 2: Visit the Computer History Museum

Googleplex, Facebook, and SmugMug. The museum was like none we had visited before. It's home to the largest international collection of computing artifacts in the world. You could sit in self-driving cars, interact with exhibits old and new, and even personalise your own punch card. The museum hosts a range of exciting exhibits, such as the Babbage Engine and IBM 1041 Demo Lab. If booked in advance, the museum offers group tours and educational workshops.

We stopped off at the Googleplex and Facebook grounds. Unfortunately, unless you have an acquaintance who is an employee at one of these large organisations, you can't enter the offices. Students were, however, happy to walk around the grounds, watching employees cycling on the iconic multicoloured Google bikes, and dodge through the fun Android Lawn Art.

In the afternoon, we were welcomed by the owner of SmugMug, a startup business offering tailor-made cloud storage for digital photography and videos. We were given a behind-the-scenes tour of the offices, and students were able to observe employees at work and ask them questions. Students even got to meet the employees' dogs who, in true Silicon Valley style, come to work with their owners! The company was launched in 2002 from a family kitchen, and quickly became a top-ranking bootstrap startup company that recently acquired Flickr. The visit was amazing, and students came away with an understanding of true innovation at work, and the skill set employers are looking for in the future.

Cost for the day: $13.50 per student at the Computer History Museum, tours free if booked well in advance, workshops also available. Googleplex, Facebook, and SmugMug – $0.

The Computer History Museum is full of historical computing artifacts

The Technology Museum is brimming with interactive resources

Day 3: The Tech Museum and digital art gallery

We had underestimated how amazing The Tech Museum would be – we had to drag the students out! The museum has hosted exhibits ranging from Leonardo da Vinci, to *Star Trek*, to *MythBusters*. We began our visit at the museum's IMAX Dome Theater, which is home to the world's first 4K laser projector. Students then worked their way around the exhibits, including Cyber Detectives, Social Robots, Reboot Reality, and The Tech for Global Good.

In the late afternoon we visited a digital art pop-up gallery. San Francisco hosts some extraordinary temporary interactive art and technology exhibitions. Students were able to draw their designs on paper, 3D-print them, and then see their designs float across the 30 ft display wall, which allowed people to touch and move.

Cost for the day: IMAX child ticket – $10, Museum child ticket – $20, digital art gallery – $4. However, students would have happily spent all day at the museum.

Day 4: Stanford University, Tesla showroom, and Apple workshop

We started the morning with a walking tour around the prestigious Stanford University grounds, ensuring we passed the Gates Computer Science Building. The department does welcome visits from schools, but unfortunately examinations were taking place during our visit. Founded in 1885, the University prides itself on innovation, discovery, and expression, and offers a range of tours (walking, golf cart, group, and engineering quad tours).

A short distance walk from Stanford University is the Tesla Supercharger showroom. The impressive showroom hosted some of the world's fastest electric vehicles, and the students enjoyed listening to the staff talk them through the wonders of the technology.

In the afternoon, we went to an Apple store for a workshop on coding and programming robotics. Students were given a historic talk about the origins of Apple, and then taught how to program Sphero robots, incorporating a range of programming skills. They were then set the challenge of mastering a series of mazes the length of the store. The staff were amazing, and the students took over the whole store!

Cost for the day: Stanford tour options walking, engineering, quad – $0, golf cart – $5, groups dependent on size – $40 per group. Tesla – $0. Apple workshop – $0.

Day 5: Alcatraz, Golden Gate Bridge, and Big Bus night tour

We could not travel all the way to San Francisco without stopping by some iconic tourist locations. We set off in early morning to beat the crowds at Alcatraz. A short ferry trip across to the island and each student was given a Cellhouse Audio Tour, where they could listen to stories from prisoners and wardens. Alcatraz was once home to some of America's most notorious criminals, such as Al 'Scarface' Capone and the 'Birdman', Robert Stroud.

In the afternoon, we walked across part of the Golden Gate Bridge. The iconic orange 746 ft tall towers and sweeping cables were a marvel of Art Deco architecture. At the start of the bridge is a small tourist and information point that displays pictures taken at the time of construction.

In the early evening, we jumped back on the Big Bus night tour, and students had the extra surprise of witnessing the annual SantaCon event taking place, where everyone takes to the streets dressed as Santa!

Cost for the day: Alcatraz – $23 per child. Golden Gate Bridge – $0 to walk, or use the free bikes from Big Bus tour, purchased on day one.

Day 6: Intel Museum and workshop, cable car ride

The Intel Museum allowed students to explorer microprocessor history, silicon chip design, and chip fabrication. The museum is full of fun, interactive exhibits, and students also took part in a two-hour-long workshop in their Learning Lab. In the afternoon, we took the cable car ride down to Union Square.

Cost for the day: Intel Museum – $0, workshop – $0, cable car ride – $4. (HW)

GEMMA MOINE

Gemma (@BSAKComputing) is a Secondary Computer Science teacher at The British School Al Khubairat in Abu Dhabi, United Arab Emirates. She is an advocate for Computer Science.

GAMING FROM THE GROUND UP

Michael Bycraft shares the story of how one of his students
created a full-size arcade game

"Mr Bycraft, do you know what a Raspberry Pi is?" Two years ago Kevin Park, then an 8th grade (Year 9) student in my Collaborative Engineering class, asked me this question. We had just begun an 'Entertainment and Gaming' unit. Kevin had seen a Raspberry Pi used to build a handheld video game emulator, and was inspired to create his own system. Most students in class constructed board games, or toys for younger students. Kevin had bigger plans.

Over the following two years, Kevin worked with a team of teachers to build a full-size arcade machine. At two metres tall, it is a monument to creativity, technical skills and collaboration – all values espoused by the Design and Innovation program here at the Korea International School (KIS).

Humble beginnings

I loved working with Raspberry Pis, and wanted to find an opportunity to introduce the Pi into the Collaborative Engineering class, and Kevin's idea fitted well. Kevin worked diligently in class to learn the coding for the Raspberry Pi. There were setbacks, but over the weeks a wooden handheld gaming device came to life. It used a 7-inch monitor, lithium battery, and a USB controller. The Pi ran the RetroPie emulator and, to everyone's amazement, worked perfectly.

Both Kevin and I were ecstatic. The game console inspired students to explore coding and to learn to use Pis. I use microcontrollers in my class often, but this represented a significant step up in technology, coding, and design. The handheld game motivated us to do more.

Building the arcade

The following year, Kevin made the most of the opportunity to extend his project in his Programming and Computer Aided Design classes. Justin Marslender, an educational technology specialist at KIS, was one of the main teachers working with Kevin. Marslender's woodworking skills complemented Park's computing knowledge well. "It was great working with Kevin on this. Since I handled most of the physical construction and he the technical side, we met regularly to see what we could and could not do. He was always thinking about how to add another feature."

After months of planning, the design began to come together. Marslender encouraged the use of recycled materials. "I really enjoyed the challenge of putting

The 'brains' of the arcade game - a Raspberry Pi - behind acrylic for the user to see, alongside the 3D-printed controller storage

this together from scratch and repurposed material – a leftover monitor and speakers; lights from several old document cameras; all sorts of scrap lumber from the MS Makerspace." The sound is provided by an old set of PC speakers set and a subwoofer. The rig is cooled by a upcycled computer fan, and a strip of LEDs light the marquee.

In a particularly inspired design, the controllers can be stored in two 3D-printed holders. Between the controllers, the Raspberry Pi is visible under a clear acrylic sheet, showing off the 'brains' of the machine for all to see. Like before, a Raspberry Pi 3B runs the RetroPie emulation station.

"I think the arcade machine at KIS here serves as a method to show students where they can expand their learning," says Marslender. The cabinet continues to inspire students to make their interests come alive. They play games at lunch, but also know that this isn't just an arcade cabinet. It's two years of work and planning, made real by their peers and teachers.

The arcade game is not the only exciting project Kevin Park has worked on. He has programmed video games, soldered custom keyboards, and designed robots for the school's competitive VEX robotics team. Park noticed that the school's rotating block schedule was complicated, so he created an automated app that emailed students their schedule each day before classes.

Kevin's work is an exemplification of the robust design programme we implement here at KIS, which begins in kindergarten, and continues through our primary and secondary schools. The educational focus is on design thinking and project-based

learning. Classes examine real-world problems, and attempt to create solutions. Students are encouraged to follow their passions, and incorporate personal interests into their work.

We try to ensure authentic educational experiences are everywhere. Elementary art students learn printmaking, and create images of endangered animals to raise awareness. They sell those prints, donating the money to animal conservation. Middle school students plan and design themed mini-golf courses, which are open to all at an annual school golf tournament. In high school, CAD students search for needs in the school, then build solutions, working closely with their teachers to ensure real applications of skills. One project saw students design a secondary STEM suite, presenting their work to the head of school. Their proposal became reality six months later.

Kevin's arcade game is now the centre of attention at our Design and Innovation Center. Amid laser cutters and 3D printers, it dominates the scene. A group of students gather around during lunch, playing a racing game. The students cheer one another on. I see a student glance down at the Raspberry Pi driving the machine and overhear, "That little thing runs all this? Cool." (HW)

Kevin sketches some initial designs to hold the controllers in the cabinet

> " KEVIN'S WORK IS AN EXEMPLIFICATION OF THE ROBUST DESIGN PROGRAMME WE IMPLEMENT HERE AT KIS

MICHAEL BYCRAFT
Michael Bycraft (@mabycraft) is head of Design and Innovation at Korea International School in Pangyo, Seoul. He teaches CAD, Makerspace, Robotics, and Design. He has been a science and technology educator for 13 years, and is a RPi Certified Educator.

A DAY IN THE LIFE OF A PRIMARY SCHOOL COMPUTING TEACHER

Managing technology, leading computing, and teaching every pupil on roll – **Sway Grantham** tells **Mac Bowley** her strategies as a computing lead at a two-form primary school

Mac Bowley: **What are the most challenging parts of being a computing lead at a primary school and how do you manage them?**

Sway Grantham: Being a primary teacher who only teaches one subject has lots of challenges, but I think there are two big ones. Firstly, equipment. Computing is a very equipment-heavy subject, which means a lot of organising to make sure everything is put away correctly, charged, and working. There's a chance that you'd feel obliged to do all of this in your own time, but there simply just isn't the time. And the children need to understand the importance of getting out and putting away equipment correctly, so this is something I always factor in to my planning. I often have photos of how the equipment should look in the box so that the children know there should be five red

crocodile clips, for example, and to check on their tables if they're misplaced!

Secondly, establishing solid and meaningful relationships is really difficult with such a small amount of contact time with each class. To manage this, I make sure I have regular conversations with children about anything that's not related to computing. They see me talking to their teachers, which means they recognise we're all working together as a team. I also run lots of clubs so that children can see me in a different environment.

**MB: What are your 'must have' teaching tools for computing?**
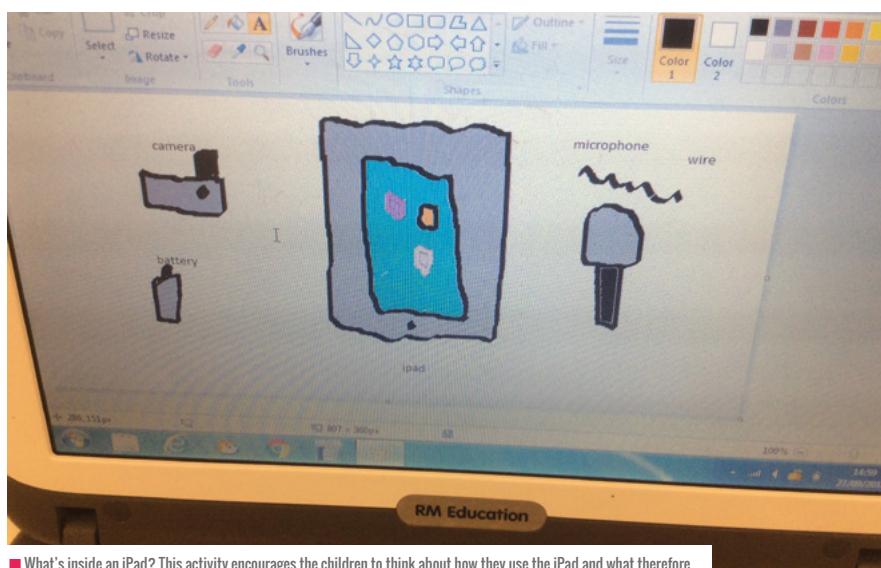
SG: In all honesty, I think it's a real danger to have 'must have' tools. While equipment is nice, you can do so much without even turning on a computer. Equipment doesn't define learning, the teacher does.

MB: What is your favourite activity to use in the classroom? Why is it your favourite?

SG: I really love anything creative. For example, in one of my recent lessons, my students used Sonic Pi to create a piece of music as a soundtrack for a section of a book. It was amazing how the children were able to communicate emotion and tone appropriately, while demonstrating their computing ability!

TEACHER TOP TIPS

1. Technology regularly doesn't work, no matter who you are or how much you like it, so always have a reserve idea in mind. I find discussion questions such as 'All computers need electricity - true or false?' work well for when systems are slow.

2. Make use of the children: they're often far more enthusiastic to be helpful, set things up, and do monotonous tasks than you are! Digital Leaders can charge, organise, and prepare equipment for your lessons - one less thing for you to do!

3. Don't worry about the time things take. Often we get really worked up that by the time the computers are turned on, Year 1 have had their input, navigated to where they need to go, etc. and there's only 20 minutes left. In the long run, it's far better to spend the lesson time learning to get things out/put them away; they will get quicker and your quality of work/life balance will thank you!

■ What's inside an iPad? This activity encourages the children to think about how they use the iPad and what therefore must be included, e.g. a camera. They then practise their mouse skills to create a drawing in a paint program!

MB: If you could go back and give yourself one piece of advice to yourself on your first day as a computing teacher, what would it be?

SG: To make mistakes and learn from them. Computing in primary schools is such a new thing, there's so little research into how to do it effectively, so how likely is it you're going to rock up and do it amazingly from the off? Pretty slim. An iterative attitude is what you need, with a willingness to review lessons and topics, and suggest changes for the next time you teach them. Each year it will get a little better.

MB: What is the best part of your job?

SG: The freedom to create, explore, and make the curriculum what I want it to be. The lack of formal assessment in Computing at primary level in the UK means we can truly explore its breadth, rather than focusing on specific end targets. I can work to what the children enjoy and like, and we can learn together. (HW)

MAC BOWLEY & SWAY GRANTHAM

Mac (@Mac_Bowley) loves digital making, video games, audiobooks, and baked goods. He's a Learning Manager at Raspberry Pi and is passionate about Computer Science education. He has taught in a range of different environments, from summer camps to GCSE classes.

Sway (@SwayGrantham) is a Computing teacher at Giffard Park Primary School and Primary Curriculum Manager for the National Centre for Computing Education. She is an RPi Certified Educator and a Specialist Leader in Education.

A TYPICAL DAY FOR SWAY

08:00 Visit class teachers to gather information for my classes today

08:30 Fix any technology problems in our classrooms - we have no IT department here

09:00 Teach first lesson to Year 5 (ages 9 and 10)
- I lead the pupils in how to set up the kit - it's an important skill
- When any technological issues are dealt with, I bring the class back together and begin explaining the tasks for the lesson
- At the end of the lesson, pupils leave the kit out ready for the next lesson

10:00 Teach second lesson to the other Year 5 class
- I might tweak this lesson, depending on how it went with the first class
- At the end of the lesson, I lead this group in how to pack away the kit
- I'll label broken kit, ready for repair

11:00 Morning break
- I'll spend the morning break managing behaviour, reporting back to class teachers, and prepping for next lesson

11:15 Teach Year 3 class (ages 7 and 8)
- When I turn on laptops, they've started an update, which I hope will be finished by the time we've done the input and starter discussions and activities
- I use materials from the earlier lesson to extend more advanced students
- I'll lead the group to pack away. If we're using equipment with lots of parts, I provide photos of what should be put where, and how many of everything there should be; e.g. three red crocodile clips

12:15 Lunchtime
- I'll use this time to feedback to class teachers I couldn't find at break time about the morning
- Children pop up to see if they can do some extra computing at lunchtime, or if there any jobs that I might need help with
- My school's Digital Leaders get out any equipment I might need for the younger children in the afternoon. They'll check apps are installed, get laptops laid out in classrooms, etc.
- I'll prepare for the afternoon ahead and eat!

13:00 Teach Year 1 lesson (ages 5 and 6)
- With the younger year groups, it's even more important to break down tasks: I'll guide them on how to get set up at the start of the lesson (turning on computer, opening a browser, and navigating to a website, for example). This all happens before they listen about the task so they don't have to remember as much at once

14:00 Teach second Year 1 lesson

15:20 Run computing club
- The children get out own equipment and help each other set up - this is an extra-curricular opportunity and I use it to make the children more independent (and I'm getting tired after a long day!)

16:45 End of day sorting
- Check what equipment needs to be charged/found for morning
- I'll plan what I need to do this evening; e.g. creating accounts on a website, changing the batteries in Pro-Bots

18:00 Home jobs
- Once home, I'll review work from the day and annotate plans accordingly
- Highlight children who might need feedback/support
- Do things on the 'to do' list for the evening in preparation for tomorrow!

AMANDA FORD GAMES LECTURER

OUT-OF-SCHOOL CARE STAFF BECOME STEM CHAMPIONS

West College Scotland and the Scottish Out of School Care Network have joined forces to help bring digital activities to children outside of school

O ut-of-school care settings provide a safe place for children to be cared for, while their parents work or are in training, before and after school and during school holidays. The Scottish Out of School Care Network provides support, training, and resources to these settings in Scotland. The aim of the organisation is for all children in the country to have access to high-quality play, care, and learning opportunities which meet their individual needs.

As part of this mission, I was tasked with the challenge of bringing more digital and STEM activities to children

access or desktop computers, and many of the care staff attending the course had no coding experience.

I decided to therefore centre the course on using Raspberry Pi computers, with 7-inch touchscreen displays running Scratch and Minecraft – the computers don't need internet access, and the software is relatively easy to learn and teach. By the end of the ten sessions, course participants were proud to have created games in Scratch and written code in Python to create rainbow walkways in Minecraft. On completion of the course, participants were

> " The aim of the organisation is for all children in the country to have access to high-quality play, care, and learning opportunities which meet their individual needs

and, in particular, girls in out-of-school care settings. So, I developed a course, entitled 'An Introduction to Creative Computing for After School Care Workers'. This was supported by a grant from the Scottish Government Children, Young People and Families Early Intervention Project Fund. Around 50 care staff have now attended the course, and over 400 children have had access to the digital activities shared in the training. Many of these children would not have taken part in extra-curricular activities, such as Code Clubs, otherwise.

Creative computing with limited experience and resources

The course was aimed to equip staff with the skills needed to introduce creative computing into their after-school care setting. Many settings don't have internet

named STEM champions, and have gone on to teach other staff at their care settings to deliver the activities to children.

Fun activities for all

The purpose of each activity we shared with the course participants was to have something that children were able to do in a short time – hopefully grabbing children's attention and then making them want to explore the resources further.

Perhaps one of the most enjoyable activities was centred around the Makey Makey – a kit that enables you to use any conductive objects in place of key presses or switches. I had created a lesson based on making a game controller and then testing objects like Play-Doh and fruit, to see which materials would be good for controlling characters. Some of the staff then worked with the children they cared for to use

■ Opening up a new world for out-of-school care staff

the Makey Makey to create dance mats or giant keyboards for stepping on.

Breaking the barriers

One of the main challenges in delivering the course was helping the care workers overcome their lack of confidence. I had to continually remind participants how much they had progressed in the space of a few weeks. One care worker told me, "The only sprite I ever knew was the drink!" Now, they are teaching the children how to create their own sprites in Scratch and make them move.

I have been inspired by the dedication and perseverance of the care workers: they learned these skills in order to provide a wider range of opportunities for the children they look after. And parents I have spoken to have been grateful to the care workers for going an extra mile to provide digital making and STEM activities to the children. **(HW)**

■ Out-of-school care services were given digital kits

KITS

Each service was provided through funding with two Pi starter kits, touchscreens, keyboard/mouse, and Makey Makey. While a lot of care settings are based in schools, they generally don't have access to computers or internet. Some services had said that they have games consoles for children to play on, but nothing for them to be creative with.

AMANDA FORD

Amanda (@aford78) is Games Lecturer at West College Scotland. She is a STEM Ambassador, Raspberry Pi Certified Educator, and Microsoft Innovative Educator Expert.

■ Dunoon Grammar pupils showcase their Apps for Good ideas and prototypes with industry experts at this year's Apps for Good Scotland event

MACHINE LEARNING
WITH APPS FOR GOOD

Paul Gallanagh shares the successes of his school's collaboration with Apps for Good

Ever wondered how the Netflix 'watch next' manages to hook you in for an unplanned weekend of binge viewing? Or how Spotify's 'recommended for you' song becomes the perfect soundtrack for your day? This is thanks largely to machine learning – a term I was completely unaware of when Apps for Good first asked if I would deliver their new course on the subject.

Stanford University defines machine learning as, "the science of getting computers to act without being explicitly programmed". In recent years, it has brought us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome.

Indeed, chances are that already today you have encountered machine learning going about your daily routine, and the impact of machine learning is going to grow exponentially, with many organisations seeking to maximise its potential.

The experience

Although we had reservations about our capacity and capability to successfully deliver a machine learning course to our students, we were keen to give it a go. We knew that that Apps for Good would support us as we progressed through the course, thanks to our experience of piloting their Internet of Things course a few years back.

The fear of this unknown was tempered with our desire to keep our computer science provision real and relevant to our students – and Apps for Good has really helped us keep our provision fresh, exciting, and current.

We were really impressed by the high-quality resources supplied by Apps for Good. Well researched, clear and concise, and accessible, these helped staff to enrich their own knowledge and skills in machine learning. Alongside these, we also used the first class, practical tasks offered by Dale Lane via his excellent Machine Learning for Kids site (**machinelearningforkids.co.uk**).

Having familiarised themselves with the resources, staff were equipped to successfully deliver this rich, experiential course to our S3 pupils (11- to 12-year-olds) who had chosen to study computer science.

It was no surprise that our pupils immersed themselves fully in this experience from the get-go. The convergence of currency of context, quality of resources, creativity, teamworking, and problem-solving proved to be a potent cocktail that brought the very best out of our young people.

As my colleague Darren Boyd, a computing teacher here at Dunoon Grammar School, commented, "I 'freaked' a little when Paul first asked us to deliver this new machine learning course. However, due to the quality of resources provided by Apps for Good, I was able to upskill myself and deliver high-quality, engaging lessons which the pupils thoroughly enjoyed. Pupils enjoyed it so much that they took it upon themselves to learn key programming

> ## "OUR PUPILS' PROJECTS SPANNED TOPICS FROM NATURAL LANGUAGE PROCESSING TO RECOMMENDATION SYSTEMS

concepts beyond the level we would expect at this stage of their learning. This was in order to create their machine learning prototypes, which is extremely exciting to witness for their development within the subject".

Healthy competition

The culmination of the Apps for Good year are the regional and national 'market place' events, where pupils have the opportunity to pitch their concepts and prototypes to industry experts.

To help prepare for these Apps for Good events, we hosted our own, inviting local industry experts to visit our school so our pupils could share their ideas with them. Over 50 teams shared their machine learning and Internet of Things concepts during a very exciting day. It was extremely pleasing to see the diverse range of ideas on show, with pupils successfully drawing on the skills and knowledge acquired on the course to produce a range of commercially viable products, with supporting prototypes.

Our pupils' projects spanned topics from natural language processing to recommendation systems and decision-making systems. These included 'Paws and Relax' – a system that recommends a stray pet currently in a rescue centre to suit an owners' needs and lifestyle, and

'Mood Master', a system that evaluates your mood based on the language used and tone of your voice when speaking on your mobile. This would message the user with recommended self-help support and guidance.

We were then delighted to hear that one of our machine learning teams had scooped the People's Choice award at the Apps for Good Scotland event, as voted for by the industry experts in attendance. 'Carbon Kicker' is designed to evaluate your current impact on the environment and recommend ways to reduce this negative impact.

All teams received invaluable feedback during the event and this advice was supplemented by further support given by staff from the software engineering firm EPAM during a recent Google Hangout expert session. All 50 of our teams are currently finalising their entries for the Apps for Good UK awards, which will be held later this year in London.

Making the impossible possible

It is no understatement when I say that Apps for Good have revolutionised our provision of computer science over the last few years. We have record numbers of pupils continuing their computer science journey with us in certificated courses in our senior school next year, and increasing numbers are continuing this journey into tertiary education and industry.

This impact is perhaps best summed up by one of our current pupils, Olivia Robertson:

"I had not heard of machine learning, but once it was explained to us, it was clear that machine learning was already part of my life and I had already been encountering it through social media and services such as Netflix.

"I would never have believed I would have been able to code a machine learning solution, and it was really satisfying to produce our own prototype and get our code working effectively". **(HW)**

PAUL GALLANAGH

Paul is Principal Teacher of Business & Computing at Dunoon Grammar School, Argyll and Bute, Scotland (@dunoongrammar1). He was named Apps for Good UK Educator of the Year 2018.

CHRIS ROFFEY AUTHOR

WHY WE NEED A NEW IDE FOR SECONDARY SCHOOLS

There's an argument for a new educational programming environment, developed specifically for teenagers

I think that our 11- to 16-year-old students need, and deserve, their own purpose-built, integrated development environment (IDE). It should be the best the industry can produce, and one that students can carry on using into their professional life, if they decide to pursue a career outside of computer science.

Currently, schools have access to many professional IDEs, with streamlined education versions as well as IDEs created specifically for students getting started with text-based programming (Thonny, IDLE, Mu, and Greenfoot – to name a few). While fantastic tools in themselves, these often support only one programming paradigm, or were developed with limited budgets or resources.

Existing educational IDEs are created under the assumption that students will eventually progress to using their professional equivalents. With most of our teenage students, this is not the case. University tutors will have plenty of time to introduce Computing undergraduates to the professional tools designed for those working in the field,

but teenagers do not need the full scope of what these can provide. They need purpose-built, intuitive tools that will aid learning, and perhaps be useful in their later lives.

What could a new IDE for secondary students look like?

It is my belief that secondary students would choose tools with power over stripped-down simplicity. The challenge, therefore, is to produce an environment that is intuitive but still has all the facilities required. Students could also customise their environment. This can be accommodated if the IDE can save the environment, as well as the source code. This will help teachers with developing resources, because they will be able to provide students with a common starting environment. The IDE should allow easy and safe sharing of projects across platforms and devices.

Currently, the process of programming usually entails: have an idea, write some code, compile, notice a problem, rewrite the code, and repeat. It is now possible to add live coding sessions to our tools so programmers can 'jam' with their code and experiment with variable values using slider-cursors directly in their code, to get instant, visual feedback of the effects and changes that occur. Students can then have a more creative experience, and teachers can set experimental tasks to help students understand how their programs work.

Most of the potential improvements to the experience of programming are available as individual 'special features'

rect(100, 100, 80, 100):

■ One potential feature of a new IDE would be instant feedback with variable values. Here, clicking and dragging the slider cursor adjusts the size of the rectangle in the program window in real-time

■ A snapshot feature would allow students to keep track of their progress

in certain IDEs. But, with care, they could all be made available in one place and presented in a form that is appropriate for a young audience.

For example, students could make use of progress tracking and keeping snapshots of the various stages in a program's development. However, as learners, they do not need the collaborative version control present in GitHub. The IDE could simply auto-save every ten minutes, and have a camera button for taking a snapshot at any particular moment. A simple text box could be called, inviting a comment to be added as a reminder of the situation when they review their project history.
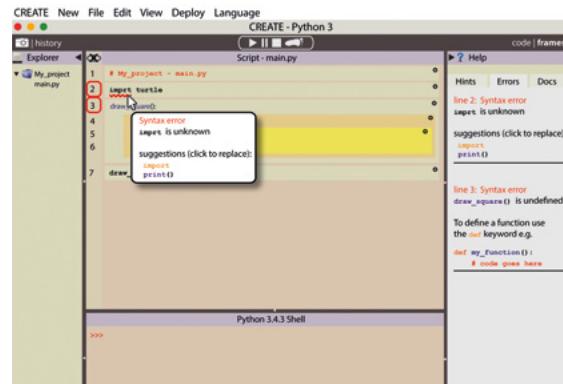
Which language?

An IDE should support all the languages commonly taught in schools in the UK. The investment necessary to create this kind of tool would be completely wasted if, in a few years' time, a new language becomes popular and it was impossible to port it to the students' IDE.

An IDE for computing education in the 21st century also needs to support all programming paradigms and all currently successful pedagogies. It should be easy to open a shell, to teach in a functional style, or use an objects-first approach. It should have facilities for easily making graphical programs for the many teachers who like to start with a wow factor, then burrow into the 'how' later.

The IDE could have its own graphical system that all languages have hooks to. No more worrying about Tkinter in Python, or whether Swing is the right way to go in Java. If it is built into the IDE, the user interface tools can be powerful and efficient. Native applications can more easily be produced at compilation time, whichever language a student is using.

By offloading the heavy lifting to the IDE, rather than the language, we are free to develop all sorts of powerful tools in the IDE. We can, for example, easily add the ability to toggle between a text-only and frame-based coding

> ## " SECONDARY STUDENTS WOULD CHOOSE TOOLS WITH POWER OVER STRIPPED-DOWN SIMPLICITY

environment or produce a powerful, interactive, and intuitive debugging system designed specifically for this audience. There will be no more need to help students understand debugging tools developed by and for an audience who work in a completely different way to learners.

A tool for everyone

There are far more students in the world than developers, so it makes sense to build something directly for them rather than amending a professional tool. There are also possible markets for such an IDE outside of education. A well-built, intuitive, and powerful IDE for the masses will have great appeal in many domains. These could include scientific and social modelling, computer-assisted musical composition, and making personalised programs to help run domestic tasks.

By accommodating a variety of languages in one tool, it may be that exam boards will find setting standard programming tasks easier. Finally, a tool that makes learning programming a much less frustrating, and more creative, process for students will also make teaching programming a much less frustrating, and more creative, endeavour. (HW)

CHRIS ROFFEY
Chris Roffey (@Coding_Club) organises the UK Bebras Challenge on behalf of Oxford University and developed and organises the TCS Oxford Computing Challenge. He is also the author of several programming books for children.

GREG MICHAELSON EMERITUS PROFESSOR OF COMPUTER SCIENCE

FROM RECURSION TO ITERATION WITH ACCUMULATION

Recursion looks like it should use lots of memory. But there's a technique for turning a simple form of recursion into iteration with minimal memory overheads

Recursion really isn't as complicated as is made out, once you get used to it. Still, there is something odd about recursion when we think about how it uses memory. Instead of repeatedly reusing the same variables, like iteration, recursion seems to generate new variables every time there's another recursive call. So what happens to all those variables, and why don't they gobble up memory? To try to understand this, we're going to look inside a common form of linear recursion, and a simple technique called tail recursion elimination, to make it more efficient.

Recursive functions

We're going to write some recursive functions in Python.

Let's start by making a recursive function to add up all the integers from some value n down to 0:

- if n is 0, we return 0. This is the *base case* when the recursion terminates;
- otherwise, we add n to the sum from n-1 to 0. This is the *recursion case* and execution continues.

In Python, that's:

```
def sumI(n):
    return 0 if n==0 else n+sumI(n-1)
```

GREG MICHAELSON

Greg is Emeritus Professor of Computer Science at Heriot Watt University. His research interests are in the design, implementation, and analysis of programming languages, and in programming education. He has taught programming since 1977.

For example:

```
sumI(3) ==>
3+sumI(2) ==>
3+2+sumI(1) ==>
3+2+1+sumI(0) ==>
3+2+1+0 ==>
6
```

During the recursion, the intermediate values from n down to 1, which here are 3, 2, and 1, must be retained. Thus, if n gets too big, it looks like we'll run out of memory.

Instead, suppose we introduce a new parameter v to hold the sum so far:

- if n is 0, we return the sum so far, v;
- otherwise, we add n to the sum so far, v, and find the rest of the sum from n-1 to 0.

In Python, that's:

```
def sumIA(v,n):
    return v if n==0 else sumIA(n+v,n-1)
```

To begin with, the sum so far will be the former base case return value of 0. For example:

```
sumIA(0,3) ==>
sumIA(3,2) ==>
sumIA(5,1) ==>
sumIA(6,0) ==>
6
```

Notice that now there aren't any intermediate values to store. Instead, we've *accumulated* the partial result in `v`, an *accumulation variable*.

This style of recursion is called *tail recursive* because, in the non-terminating recursion case, the call only involves the function itself rather than a more elaborate expression. Finally, let's turn the tail recursion into iteration. Note that the recursive call looks like a loop with modified variables. So we'll make the accumulation variable `v` a local variable, initialised with the base case return value. And we'll modify `v` and `n` through assignment:

```
def sumII(n):
    v = 0
    while n>0:
        v = n+v
        n = n-1
    return v
```

Note how the recursion termination condition `n==0` became the iteration non-termination condition `n>0`.

Joining string lists and commutativity

Now for a different example, let's join together all the elements in a list of strings, starting at element `i`:

```
def joinL(l,i):
    return "" if i==len(l) else
l[i]+joinL(l,i+1)
```

For example, to join all the elements in a list starting with the first:

```
joinL(["a","b","c"],0) ==>
"a"+joinL(["a","b","c"],1) ==>
"a"+"b"+joinL(["a","b","c"],2) ==>
"a"+"b"+"c"+joinL(["a","b","c"],3) ==>
"a"+"b"+"c"+"" ==>
"abc"
```

As before, let's introduce an accumulation variable `v` for the partial join:

```
def joinLA(v,l,i):
    return v if i==len(l)
            else joinLA(l[i]+v,l,i+1)
```

So:

```
joinLA("",["a","b","c"],0) ==>
joinLA("a",["a","b","c"],1) ==>
joinLA("ba",["a","b","c"],2) ==>
joinLA("cba",["a","b","c"],2) ==>
"cba"
```

Alas, we've reversed the list!

The difficulty is that, in making the tail recursive function, we've copied the recursion case expression:
`l[i]+joinL(l,i+1)`
…replacing the recursive call with the accumulation variable:
`l[i]+v`

The recursive call is the *rest of the computation*, but the accumulation variable is the *result so far*. So we've changed the order of evaluation. In the original function, we put the first element in front of the *rest of the join*; now we're putting it in front of the *join so far*.

In the sum and product functions this didn't matter, as integer + and * are *commutative*; that is, the order of evaluation doesn't normally affect the result:

$$integer1 + integer2 = integer2 + integer1$$
$$integer1 * integer2 = integer2 * integer1$$

But list concatenation + isn't commutative:

$$list1 + list2 \neq list2 + list1$$

Instead, we should have reversed the recursion expression so that the next element goes *behind* the result so far:

```
def joinLA (v,l,i):
    return v if i==len(l)
            else joinLA(v+l[i],l,i+1)
```

Thus:

```
joinLA("",["a","b","c"],0) ==>
joinLA("a",["a","b","c"],1) ==>
joinLA("ab",["a","b","c"],2) ==>
joinLA("abc",["a","b","c"],2) ==>
"abc"
```

Finally, the iterative version is:

```
def joinLI(l,i):
    v = ""
    while i<len(l):
        v = v+l[i]
        i = i+1
    return v
```

One of the myths about recursive programs is that they're inherently slow and inefficient, and this is sometimes used to critique functional programming languages. Actually, speed and efficiency are properties of implementations, not languages. Functional language implementations, like GHC for Haskell, now generate code that compares well with that for imperative languages, using optimisations just like tail recursion elimination. (HW)

MARC MCWHIRTER HEAD OF COMPUTING AND TECHNICAL EDUCATION

TEACH MACHINE LEARNING

We encounter machine learning systems every day. We should teach them too

Machine learning is increasingly pervasive in our lives: there are virtual personal assistants, like Siri and Alexa, navigation systems that update suggested routes to take account of traffic, targeted ads, and 'chatbot' online customer services, to name just a few of the applications of this emerging field of technology.

Why teach machine learning?

Natural language processing has become part of students' everyday lives, as they interact with digital voice assistants. These convert speech to text, and use machine learning to carry out the requested commands. As technology like this becomes more accessible, it makes sense to bring it into our classrooms so that students can develop solutions to modern problems using modern technology.

Machine learning also opens the door to ethical discussions that need to be happening in computer science classrooms across the country: should we allow machines to make decisions for us? An interesting starter for these debates is the MIT Moral Machine (**helloworld.cc/moral**). It is our responsibility, as educators, to ensure that we are developing our young people to consider the social consequences of their future technological creations.

At my school, I deliver machine learning to my S3 (Year 9, ages 13-14) classes, who have a good grounding in programming, with a strong foundation of computational constructs and data structures. The topic has captured their imagination; machine learning is fun, allowing them to solve more complex problems than they have previously been able to. Within our first 50-minute lesson, the students had created an image recognition program in Scratch, which could predict the genre of a book, based on its cover.

Marc McWhirter is Head of Computing and Technical Education at Speyside High School in Moray, Scotland.

MACHINE LEARNING WITH PYTHON

Some students will be able to progress onto using a textual-based language for machine learning, and Machine Learning for Kids provides instructions for some projects using the Python programming language. Alternatively, TensorFlow (**tensorflow.org**) is an open-source library that can be used to extend students' machine-learning projects, and can be used with the online IDE **repl.it**

Well resourced

All students are taking part in the Apps for Good project (**appsforgood.org**), where they create a solution to a real-world problem using machine learning. Apps for Good have developed a free, adaptable curriculum for introducing the topic of machine learning to students.

A great resource for machine learning is the website, Machine Learning for Kids (**helloworld.cc/mlforkids**), which provides step-by-step instructions for creating a variety of projects. These can make use of image recognition and natural language processing. The website also facilitates a connection between Scratch 3 and the IBM Watson platform – a business tool used to create chatbots and data platforms. These projects form the foundations of the skills that learners would require to begin to solve their own problems using machine learning.

I am a great believer in reducing the barrier to entry into computer science, using environments in the classroom which require as little setup as possible, so that students can engage with their learning at home if the spark of interest is there. In this case, the only requirement is a modern web browser which can support Scratch 3, and the Machine Learning for Kids website. **(HW)**

MARK THORNBER TEACHER
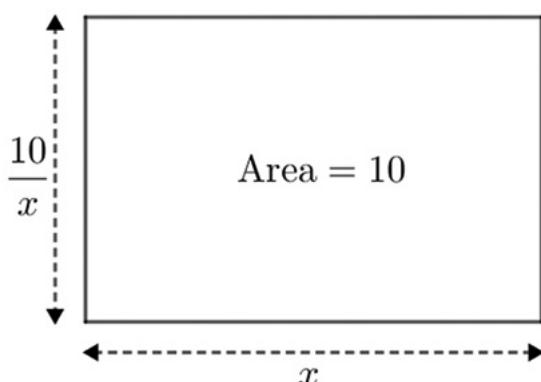
MATHEMATICAL MUSINGS

How do computers solve tricky mathematical problems?

We know computers are good at simple arithmetic, but have you ever wondered how they produce answers to more complicated problems like finding roots, or trig ratios? There are buttons on a calculator to do these things, but the chip inside just does basic arithmetic. The key is that the calculator or computer is usually doing many more calculations than you think. In this article, we'll look at how to find a square root.

In maths lessons we learn about trial and improvement. Suppose we want to find $x=\sqrt{10}$. We could proceed as follows: First try $x=3$. Since $x^2=9$, this is too small. Now try $x=3.1$ and calculate $3.1^2=9.61$. Still too small, so try $x=3.2$, and so on.

On average, this needs five guesses to get each decimal digit correct, so would need about 50 tries for ten decimal place accuracy. Modern computers would do this almost instantly, but we can do better with a simpler, more visual approach. The method is to start with any guess, x, and then get a better guess as follows. First, calculate $10/x$ and then notice that we have two sides of a rectangle with area 10. We would like a square with area 10, so one side is too long and one is too short. It doesn't matter which is which: the average of the two will be in between and so will make a better guess.

The formula is

$$\text{`new' } x = \frac{1}{2}\left(x+\frac{10}{x}\right)$$

We stop when our old guess and new guess agree to however many decimal places we need.

We can write a few lines in Python to do this. A program and sample output are given below. As you can see, we get 6 decimal places after only five guesses. This is much better than simple trial and improvement.

```python
accuracy = 6 #Number of decimal places
num = 10
x = num/2
oldx = x+1
n = 0
notdone = True
while notdone:
    print(n,x)
    if round(abs(x-oldx),accuracy+1)==0:
        notdone = False
    else:
        oldx = x
        x = 0.5*(x+num/x)
        n += 1
print("After",n,"iterations")
print("sqrt",num," =",round(x,accuracy))
```

```
0 5.0
1 3.5
2 3.178571428571429
3 3.162319422150883
4 3.1622776604441363
5 3.162277660168379
After 5 iterations
sqrt 10  = 3.162278
```

ESCAPE THE CLASSROOM

Claire Wicher shares her escape game resources – an exciting way of encouraging students to problem-solve, promote communication, and develop their teamworking skills

Originally developed for educational visits to Bletchley Park, Escape the Classroom combines the problem-solving approach to escape games with computing knowledge, such as using binary numbers and cryptography. I use the Breakout EDU Kit, which costs $150, but you can also buy basic locks and make a kit for less, if you're willing to shop around.

The challenge has two parts: breaking into boxes and escaping the room, and solving a mystery. I use a list of teachers who might have stolen the school bell, and my clues either help narrow down the list until only one person remains, or allow students to open one of the locks on a lockbox. Lockboxes contain more clues and puzzle pieces, and the last box provides the key to escape the classroom and report the perpetrator of the crime!

There are hundreds of different ciphers you can use to decode clues, and these can be changed to suit the ability of each group. Good ciphers to start with are substitution ciphers (hieroglyphics or Morse code), a Caesar cipher, a pigpen cipher, or a mobile phone code. You can also be creative with the way you supply clues, such as writing on the wall in invisible ink, and using UV lights. My favourite is placing information on a USB pen drive, as this allows me to also include video and audio files, or even Scratch or Python programs, as part of the game.

My escape game focuses on ciphers to support my lessons on cryptography, network security, and encryption, but you could use the mystery element to have students research or investigate any topic. The topic could be from your classes, a link to the focus for another subject, or even whole a school initiative.
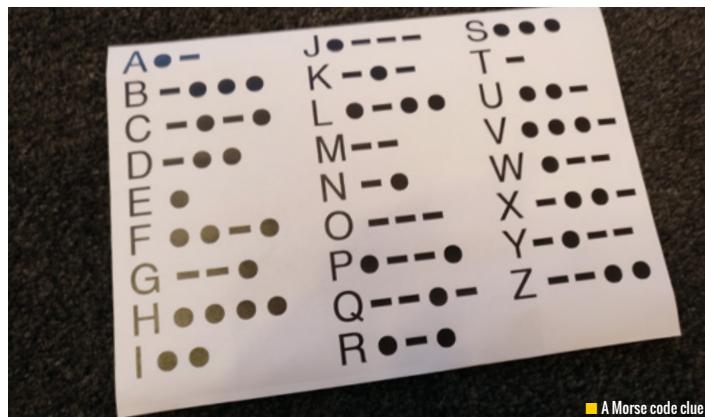
I've provided a fully completed escape game (**helloworld.cc/ escapetheroom**) that should be straightforward to set up. However,

THE CHALLENGE

✔ Someone has stolen the school bell!

✔ There are ten teachers on your suspect list

✔ Solve clues to narrow down the list of suspects, and find the culprit

✔ Once you know who it is, escape the classroom to report them

✔ The team with the fastest escape time *and* the correct suspect wins!

✔ You have a maximum of 45 minutes to escape

■ An example of one of the challenges I set my students: a mobile phone clue and puzzle

■ Claire's students working on a variety of cipher problems

■ A Morse code clue

> ## THERE ARE HUNDREDS OF DIFFERENT CIPHERS YOU CAN USE TO DECODE CLUES

it could also be used as a starting point, and you could change the puzzles or the mystery to suit your needs, your students, and your school. Once you're comfortable with how the games work, switching out puzzles, clues, and changing the theme is simple.

My students loved this activity; it helped them to learn about the different methods of encrypting and decrypting secret messages. The challenge aspect really helps to increase motivation, and being able to practise cracking ciphers in an engaging and practical environment ensures this is one lesson they won't forget anytime soon! **(HW)**

CLAIRE WICHER
Claire (@GirlGeekUpNorth) is a Newly Qualified Teacher at MEA Central in Manchester. She is founder of CodeUp UK and a Raspberry Pi Certified Educator, Pi-Top Champion, and National Coding Week Ambassador.

ALTERNATIVE ACTIVITY IDEAS

How to adapt these resources

Cross-curricular

Escape games are great for working with other subjects. Themes can be changed to fit in with other subjects. While cryptography is great to link to your history department's World War II topic, you could provide clues or answers in another language to work with the languages department, add in physical skill puzzles to introduce an element of PE, or simply theme your mystery around a fact-finding mission related to any topic!

Differentiation

You can easily adapt the games for different attainment levels or age groups. There are a few things you can quickly tweak to add or remove complexity:

- ✔ Colour-code resources so students can see which clue solves which puzzle; remove this to make it harder
- ✔ Provide only one clue/puzzle at a time, so students stay focused and there's less chance of confusion
- ✔ Use basic substitution ciphers for primary students, and complex ciphers for high-achieving A level students
- ✔ Split larger groups and provide multiple puzzles to work on simultaneously, to ensure no one is left out
- ✔ Run multiple games simultaneously by preparing identical games and removing the need to move around the room
- ✔ Recreate the escape game in digital format for each student to complete on their own/in pairs online for full classes

FURTHER READING

- ✔ **Download the Escape the Classroom resource:** `helloworld.cc/escapetheroom`
- ✔ **Visit Bletchley Park with your students:** `helloworld.cc/bletchley`
- ✔ **Cryptography tools from dCode:** `dcode.fr`

INTERACTIVE FICTION WITH PYTHON

Nicholas Provenzano explains how he introduced Python to students in his literature class, bridging computer science and literacy

L iterature classes seem like the last place you would find students coding, but interactive fiction has been around for decades. Students love to play computer games, and the very best games have amazing stories. This project will allow students to create their own piece of fiction and then use Python to turn it into a text-based computer game. Students will have a chance to create their own hero and monsters, treasures and traps and so much more while being introduced to Python. Students that love to write, and students that love to code, will love this lesson.

" A WAY TO BRING CODING TO A LITERATURE CLASS

■ A 12-year-old student works on her story in her literature class

I've been thinking a lot recently about to ways to bring computer science into the literature classroom. I set out exploring the Raspberry Pi projects page, where I saw a project that allowed the user to create their own text-based computer game using Python. I thought this would be a great way to engage students in reading, writing, and programming.
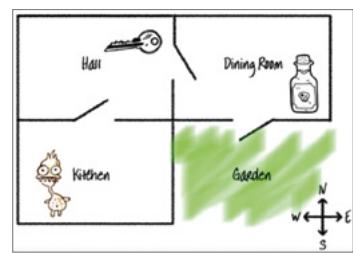
Students create their own piece of fiction and then use their stories to create an amazing text-based computer game based on the role-playing game (RPG) tutorial from Raspberry Pi: **helloworld.cc/rpg**. From the first day working on this project, my students fell in love with the writing and the coding. They couldn't wait to create their game and share them with their friends.

The project is best introduced with a focus on creative writing, where students should create an outline for their own adventure story. With that in hand, introduce the students to the Raspberry Pi RPG tutorial. It is much easier for students to create their game if they draw out the rooms on paper to help them visualise the game they're creating. The more time they are given to create their game, the more complex it can become. Students will be able to fully explore the code while creating a fun game they can share with others.

This project is the perfect way to bring coding to a literature class. Students that love to write

will be introduced to text-based programming, while students that love to code will have an opportunity to explore fiction through their own writing.
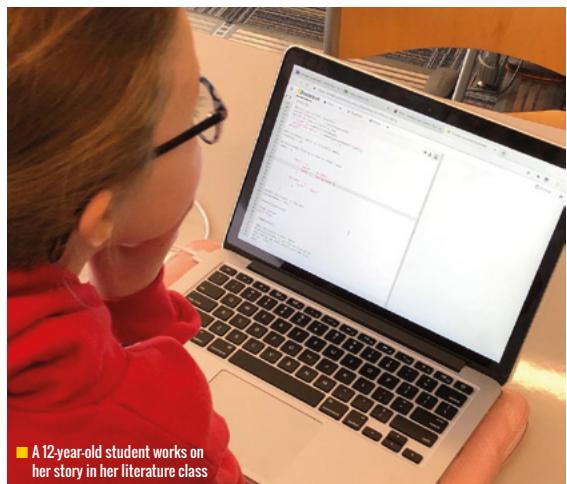
My students were excited to spend their time creating a complex story, and an even more complex game to challenge their friends and their teacher. Students who struggled with the code were helped by other students who'd already moved ahead. We spent a week on this project, but you could spend


■ Students design their own mazes for their characters to move through

longer, depending on the breadth of the stories and games. Watching students use their critical thinking skills to plan out a maze for their players was great to see.

The best part was watching students who do not normally engage in reading and writing lessons become leaders as they embraced the coding and were excited to turn their story into a game and share it with everyone. This project will become a mainstay in my teaching for years to come. **(HW)**

THE CHALLENGE

✔ Introduce students to short adventure stories

✔ Have students start to formulate a rough outline for their very own adventure story. Make sure that the stories have the following elements:
  • A protagonist
  • An antagonist
  • A special item
  • A happy ending

✔ Once the students have their stories, introduce them to the RPG tutorial from Raspberry Pi

✔ When the students open their Trinket page, make sure to have them create an account and save their work. They can lose their work if they do not create an account and save

✔ Students will need to spend a couple of days working on the tutorial

✔ They will be tempted to go off-script and create their own game as they do the tutorial, but it is important for them to complete the tutorial before creating their own game

✔ The most common errors the students will make will have to do with missed commas, quote marks, and incorrect indentation

✔ Once the students have finished the tutorial, they should start to create their own game that includes the following:
  • Six rooms
  • Three items to pick up
  • A way to lose
  • A way to win

✔ Students can be given a few hours of lesson time to complete their games

✔ Once the students have completed their games, they should share them with others

ASSESSMENT

✔ How are players able to move to different rooms?

✔ How would you add stairs to the game?

✔ What was a common error you encountered?

NICHOLAS PROVENZANO
(@thenerdyteacher) is a Makerspace Director at University Liggett School in Michigan. He is also an author, international speaker and consultant. Nicholas is a Google Certified Innovator, ASCD Emerging Leader, Raspberry Pi Certified Educator, Adobe Education Leader and a TEDEd Innovative Educator. His best-selling books, *Your Starter Guide to Makerspaces*, and *The Maker Mentality*, are read by educators around the world.

ALTERNATIVE ACTIVITY IDEAS

One of the great things about this project is that it can be used by many different age groups. Here are some ideas on how to bring this lesson to more students.

14-19 years - Text-based programming

✔ Students add different levels to their game
✔ The older the students, the more complex the stories can be

FURTHER READING

✔ RPG project from Raspberry Pi: `helloworld.cc/rpg`
✔ Example RPG game: `helloworld.cc/rpg_trinket`
✔ Nicholas's game: `helloworld.cc/werewolf`
✔ Text adventure games for inspiration: `textadventures.co.uk`

AGE RANGE

11-13

RESOURCE TYPE

Project-based learning

REQUIREMENTS

- helloworld.cc/ applab_sow

MAKE MOBILE PHONE APPS WITH APP LAB

Ben Garside shares a scheme of work based on App Lab, an engaging tool to make shareable mobile apps using JavaScript blocks or text

I have written a set of lessons where students create phone apps using the coding environment App Lab from **code.org**. I have designed the unit to make sure that it incorporates lots of elements of the UK's Key Stage 3 programme of study (details are included with the resources at **helloworld.cc/applab_sow**). **The lessons also cover many elements of the US Level 2 CSTA K-12 Computer Science Standards.**

The resources encourage students to work in pairs to play the roles of the inventors, designers, problem-solvers and, most importantly, the coders.

During the unit, they'll learn how to use the coding environment, then modify a pre-existing app, before the exciting challenge of building and sharing their own app with classmates, family, and friends.
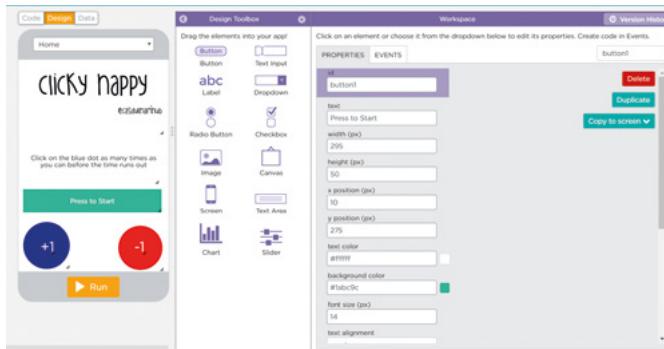
The interface

The interface is split between three windows: Code, Design, and Data.

The Design window works by dragging and dropping design elements on the screen so that students can position and format items to get the exact effect they are after, without needing to know HTML or CSS.

The Code window allows you to power the app and, very much like Scratch or MakeCode for the micro:bit, you can drop blocks of code into the coding area of the screen. Gentle red warnings will appear if you've placed blocks that will stop the app from running. More confident students will be able to navigate to the 'Show Text' button and code using JavaScript without the blocks.

The Data window is where you can set up a database, if required, to store data entry from the app. You can also see what data has been entered via this window if your app has gone live.

■ App Lab interface and coding blocks

■ App Lab in Design Mode, showing customisable properties using the GUI

LESSON 1: GETTING STARTED WITH A CLASS

I recommend that you sign up as a teacher on **code.org** to set up a class, and print out login cards in advance of your first lesson. I've made a short tutorial on how to do this can be found at: **helloworld.cc/garside_setup**.

Once you're set up, I allow the students to work in pairs to follow the initial tutorial on App Lab designed by **code.org** for Hour Of Code. The built-in tutorials will, in a short space of time, get the students familiar with the interface and give them a flavour of what is possible when coding an app using App Lab.

PEDAGOGY

These lessons make use of paired programming and PRIMM (see issue 5 of *Hello World*).

LESSON 2: REMIXING AN APP

Once students are comfortable with the interface, the second lesson allows the students to take a shell of an app that has been built for them and add more functionality to it. App Lab allows you to look at the code written for this app and then select the 'Remix' button. This duplicates the app into students' accounts for them to edit. The app they are given has a 'Start' button that, when clicked, opens a new screen with a new button to tap. The task is to change the UI so that it looks better and is clear what the purpose of the app is. Then they must code the circle so that when it is clicked, it moves to a new random position on the screen.

■ Example of some of the PRIMM questions for the students to explore

| Code/Screen	PRIMM Question	Your Answer	Were you right? Did it do anything that you weren't expecting? What was different to your prediction?
	What do you think will happen to the variable "score" once the game is running?		
	What do you think this block of code will do? What is the significance of the "5000"?		

Evaluating the lessons
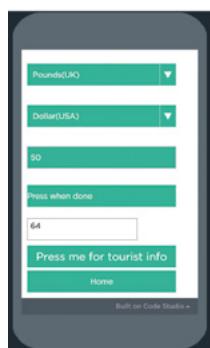
What I found worked nicely about this unit was that there is something for all students to be able to thrive at. Some students enjoyed the design elements whereas others enjoyed the coding side (this makes getting the pairs right in your classes more important). Regardless of where their strengths lie, they are all completing tasks that involve key elements of computational thinking.

Paired programming at the beginning of the unit worked really well, particularly with our students who lack confidence in their coding ability.

Not all of our groups ended up with a fully functional app, but all of my students were able to decompose the problem, then design and create an app that at least had a couple of screens, with a button that was coded to perform an action.

Finally, my unit of work is planned for five hours of lesson time. I think that although all students produced an app of some description, six to seven hours would have been more beneficial to the students. **(HW)**

LESSON 3-5: PROJECT DEVELOPMENT

Project 1 – Grandparent/Teenager Translator

- The app will allow grandparents to open the app and select some key phrases that teenagers would say and the app with translate the terms so that they can understand them.

- You should try and expand on this idea by including things such as translating the other way round (grandparents into teenage speak), or adding your voice to the app saying the phrases, or you might want to even give the user a choice of UK region. It's important that the phrases you choose for the app are kept appropriate for the audience

■ The grandparent to teenager translator app

Now the students are ready to start planning their own app! It might be a good idea to give them a choice of projects to develop, rather than giving them free rein to decide that they want to make the next first-person-shooter hit app. Examples of projects you might give them could be a grandparent/teenager translator app, a logo quiz, or a biscuit clicker game. Get the students, in their pairs, to decompose the problem, add success criteria, and plan their time over the remaining lessons. Encourage the students to draw on each other's strengths by getting one student to complete screen designs whilst the other half of the pair starts the code. Depending on your class, you could leave it in their hands as to when and who to get feedback from, before implementing changes as a result of the feedback.

Project 1 – Success Criteria

Below are mandatory success criteria, you should add between 2 and 4 additional success criteria that you would like your project to be judged on

Criteria	Met?	Developing	Secure	Exceeding
1. App must have a welcome screen with a button to start the program				
2. App must have a minimum of three phrases to be translated				
3 . App must allow a phrase to be selected before the answer is revealed				
4.				
5.				
6.				
7.				

■ Students add their own ideas by giving themselves success criteria for which they want their app to be judged at the end of the unit

RESOURCES

The full set of resources for this unit, including an outline of each lesson, is available to download at:
helloworld.cc/applab_sow

BEN GARSIDE

Ben Garside is a teacher at Durham Johnston School and a CAS Master Teacher. He is soon to join the Raspberry Pi Foundation as a Learning Manager.

■ Examples of the apps that students from Durham Johnston Comprehensive School have made

GET SPARKLING WITH CRUMBLES

Get started using the Crumble controller robotics board to teach and learn physical computing. Pablo Joyce shows how to start with some lights!

Ever woken up one morning with the insane desire to make an all-powerful robot? Well, every great creation starts with the simple baby steps and what could be a better way to go than the easy-to-use, low-cost Crumble controller board to get started?

The Crumble controller robotics board is a really accessible base for creating simple, controllable projects using LEDs and sensors – as well as complex robotic creations with motors and servos – that can be programmed on a PC, Mac, and now, by popular request, on Chromebooks.

We are going to look at first steps with this fantastic board, plugging in, powering up, and lighting up some lights. Lighting an LED is a great way to start physical computing, but the Crumble does this in style with its easily connected and coded RGB Sparkles – tri-colour LEDs made especially to work with the Crumble controller.

Getting connected

OK, let's get started! If you haven't already done so, you will need to download and install the Crumble software: **helloworld.cc/crumblesw**. This is a Scratch-like intuitive, block-based coding app custom-made
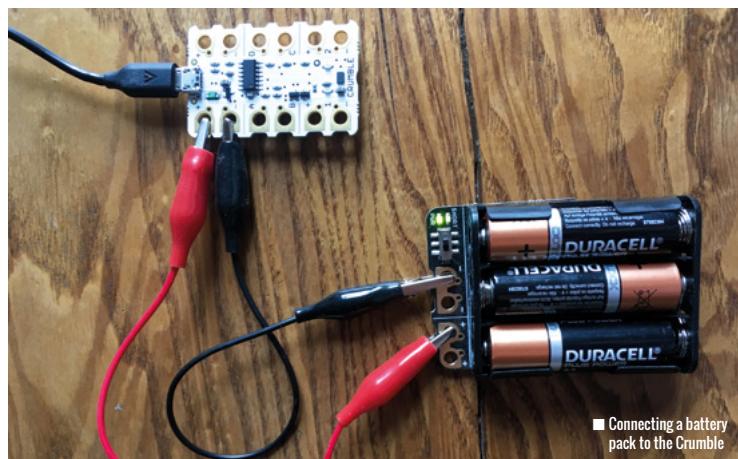
for the Crumble by the maker, Redfern Electronics.

Once we have the software ready to go, we will connect the Crumble controller to our computer using a USB to micro USB cable. The computer is where we will write our code and transfer it to the Crumble board. If I plug it into a Windows PC, I get the standard beep to tell me a USB device has been connected, but on a Mac there is no sign of life at all. No beep. No LED. Nothing appears in Finder! It is a little unnerving, but it is easy to check that your Crumble is present and correct: just open up the Crumble software and click the play button (green arrow). Assuming your Crumble is connected, you will get a confirmation, 'programming successful', even if there is no programming. If not, you will get a error telling you that no Crumble is connected.

Power up!

Now that we're connected, we need to connect some power to the Crumble controller board. The Crumble starter kit comes with a 3×AA battery pack that has a switch and 'crumble-friendly' croc-clip connectors, just like the ones on the Crumble controller. This makes connecting everything up so simple. If only we had more battery boxes like this for science circuits learning! Just make sure that you connect positive to positive (4.5 V to 5 V) and negative to negative (often called ground or GND). I have used the correct colour cables for clarity, but any colour cables will work just the same.

The power supply to the board gives a constant output to be used with the things that you connect that need external power, such as Crumble Sparkles, motors, and servos.

■ This message tells us that our Crumble controller is connected

▶ ■ ↻ Programming successful

■ Connecting a battery pack to the Crumble

■ Connecting a Sparkle to the Crumble controller

Get your Sparkle on

We're connected and powered up, so now it's time to add our first Sparkle. Crumble Sparkles are tri-colour (RGB) LEDs made especially to work with the Crumble controller. This means that we can program the red, green, and blue values to get pretty much whatever colour we want – over 16 million different combinations!

The Crumble works so well in primary schools because connecting things is so easy, and the Sparkles are a great example of this. The croc-friendly connections are clearly marked and consistent. The positive connects to positive, the negative connects to negative, and the third connector, marked D, needs to connect to the 'D' connection on our Crumble controller.

PABLO JOYCE
Pablo (@pablisch) is a primary computing and STEAM teacher at The Devonshire Hill Primary School, computing teacher at The Rosemary Works, RPi Educator, and compulsive maker.

Note: All Sparkles connect to the 'D' connector. You can connect them in a long line – up to 32 Sparkles on one Crumble.

Each connected Sparkle is independently controllable with its assigned number. It might be a little confusing at first that the first Sparkle is designated as 0, the second as 1, the third as 2, etc. but this is a helpful start to the idea that things start at 0 rather than 1.

When laid out straight, the croc clips connecting your Crumble and Sparkles should run parallel and not cross. Connecting the positive and negative wrongly could destroy your Sparkle.

■ The six Sparkle code blocks

Time to shine!

Time to code our first Sparkle and light the light! In the Crumble software, the first thing we need is the block in Basic called 'program start', this is like the Crumble version of Scratch's green flag. This is the block that all our Crumble code will start with and currently there can only be one of these blocks. Anything that is not connected to the 'program start' block will not run.

In the Sparkles code block palette, there are three basic options: specify a colour visually, turn Sparkles off, or specify a colour by giving the RGB values from 0 to 255. Each of these options can be used for a single Sparkle or all Sparkles.

Pull out the block 'set sparkle 0 to red' (where the red is a red box), connect it under the 'program start' block, and run the program (click the green play arrow). Your Sparkle should have lit up red. Woohoo!

Click on the red colour box and choose a colour for your Sparkle. The colour picker looks and works

❝ AN ACCESSIBLE BASE FOR CREATING SIMPLE, CONTROLLABLE PROJECTS

differently depending on the OS you are using, but pick whatever colour you want, including saturation and brightness, and then in the Windows version you will need to click OK. Run the program again. Has your Sparkle changed colour?

Now let's play around with the RGB values. Throw the 'set sparkle to colour' block away by dragging it into the blocks palette and drag out the 'set sparkle 0 to 0 0 0' block, connecting it to the 'program start' block. Of course, we need to leave the first 0 at 0 because that is telling us which Sparkle we are lighting, but the following three 0s are giving the red, green, and blue values as shown by their box colour. Values are from 0 to 255, where all three at 0 means black (unlit) and all at 255 means white. What do you think you will get if you put red and blue at 255 and ▶

green at 0? Try it out! Put in the values you want to try and run the program. How about green and red at 255 and blue at 0?

■ A forever loop containing a sequence of colour changes and waits

Colour sequences

One colour, or one of 16 million colours is fine, but now we can move on to sequences of colours. The simplest way to do this is by telling the Sparkle to be one colour, wait, change to a new colour, wait, etc. This could be a simple sequence that ends, or in a loop so that it goes on forever.

Once you have chosen a sequence of colours in an order you like, drag out the 'wait' block from the Control palette. You can choose to have your wait in either seconds

■ Controlling the colour of two sparkles

TAKE IT FURTHER CHALLENGES

Ideas to stretch your students

- ✔ Use a 'random' block to show random colours
- ✔ Use variables to make stepped colour changes
- ✔ What could you make using many Sparkles?
- ✔ What would you make by adding motors too?

FURTHER READING

Redfern Electronics, the maker of Crumble controllers, has a website with project ideas and a downloadable Getting Started guide - helloworld.cc/redfern.

Phil Bagge has produced a range of resources for the Crumble, and has even published a Crumble Creations book full of Crumble STEM projects to get stuck into - helloworld.cc/codeit.

or milliseconds – so good to have the choice!

Put a 'wait' block after every colour change. If your are looping them, you will need a wait after the last one too! To make this loop endlessly, simply grab a 'do forever' loop from Control, put all your colour changes and waits inside, and connect to the 'program start block'... and run!

You want more?

One Sparkle is fun, but we can connect up to 32 of these little beauties together, so let's look at getting a second one up and running right away! It's very simple: we just use more croc clips connecting from the free connectors on the right side of our connected Sparkle to the corresponding ones on the left side of the next Sparkle and carry on like that for any extra Sparkles we connect. Always check that you are connecting positive to positive, negative to negative, and signal (D) to signal (D). Your first Sparkle will always be 0, the second will be Sparkle 1 and so on. You can code them individually or altogether as needed.

We've just had a brief look at the simplest colour coding, but there is so much more that we can do. In Operators, we can use the 'random' block along with the RGB value setting block to create random colours. We can use the same

■ You can connect up to 32 Sparkles

■ Using random colours
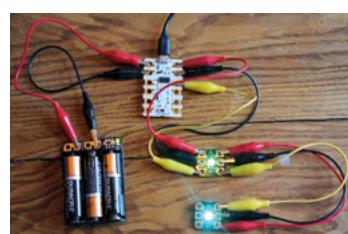
'random' block to select a random Sparkle to change or randomise the wait time between changes. If we start using variables, which you may have used before in Scratch, we can create sequential colour changes to, for example, move from yellow to red. Even with just Sparkles, there is so much coding to explore, either through tinkering or having a certain effect you want to achieve and working towards that.
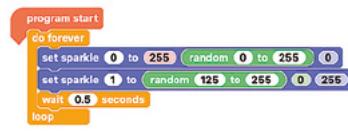
■ Pablo's Iron Man silhouette with Sparkles for eyes

Using the light

Of course, how you end up programming your Sparkles will probably depend on your purpose in having lights and, let's face it, light is a big part of our real world as well as any number of imaginary ones. What is your Sparkle going to become? Car lights, a volcano spark, the end of a magic wand, a hideous beast's menacing eyes? So many possibilities. Having a target outcome can really drive any making and coding project just by creating an achievable, purposeful goal that you are working towards. There may be some parameters for a project set by you or it may be completely open for your students to decide. I have a silhouette of Ted Hughes's Iron Man appearing off the cliffs, but even within the topic of *The Iron Man*, there are many scenes I could have chosen. Maybe I'll make a space bat angel dragon next time? Maybe I'll use motors too, so that he has flapping wings? **(HW)**

PARSON'S IN PRACTICE

Parson's puzzles require students to organise jumbled lines of code. **James Robinson** outlines their benefits and shares some ideas of how to apply them in your lessons

Parson's puzzles can help beginner programmers learn how to code. Learners are presented with a program - either in pseudocode, blocks or text, that has been split into sections and then jumbled up.

Have a go at solving this 'number guessing' puzzle and construct a program which will ask you to guess a random number and give you a hint to let you know if your answer was too high or too low.

```
print("Too low")
```

```
while
|
|
|
|
|
endwhile
```

JAMES ROBINSON
James Robinson (@LegoJames) is a Senior Learning Manager at the Raspberry Pi Foundation.

```
print("Well done, you guessed
correctly.")
```

```
guess = input("Guess a
number")
```

```
if guess !=number then
|
else
|
end if
```

```
number = random_number()
```

```
print("Too high")
```

```
guess = input("Try again")
```

To increase the difficulty, you could split up the if, else, endif, while, and endwhile. The solution to this puzzle is on page 94.

How to use Parson's puzzles

You might want to present a puzzle to the whole group as a starter activity for discussion, perhaps to address a misconception, such as that a variable can store multiple values or a history of values. You can present a short program with multiple variable assignments but jumble up the statements and ask students to rearrange. This works with blocks too!

For longer programs, you may like to deploy a tangible activity to pairs or small groups of learners. In this scenario you can print a program onto paper, slice it into sections like a card sort activity, and ask learners to put them programs together correctly. After a period of time, learners share their solution with the wider group, addressing any issues that arise.

You can even apply this approach to assessment, by creating a drag-and-drop-style activity that learners complete on their own. This gives you, the teacher, a quick formative assessment tool.

There are several online tools that can be used to create your own Parson's puzzles.

What the evidence says
The benefits of Parson's puzzles include:
- Learners find it easier to reassemble chunks of code, over writing programs from square one
- The puzzles encourage learners to focus on the concepts in a program rather than the syntax
- They require learners to read and comprehend code, which scaffolds their learning
- They are inclusive of different abilities and approaches

Why not try this approach in your lessons and let us know how you get on? Email us at **contact@helloworld.cc** or Tweet us **@HelloWorld_Edu** to share your ideas. (HW)

NO HEADACHES
HOMEWORK

Alan O'Donohoe suggests a strategy to make sure
computing homework never causes any headaches again

Some of the challenges facing computing teachers can be mitigated through an effective homework strategy. This guide will suggest a simple, yet highly effective, strategy for homework that will help all computing teachers manage the challenges, as well as other people's homework expectations. The guide starts with a simple description of the strategy, and then explores each step in detail to help make computing homework a success in your school.

Computing teachers have voiced concerns such as:

- Computing lessons are too short; by the time any theory has been taught, there is little time remaining in lessons for problem-solving activities, programming and practical work
- When there is too much knowledge content in lessons, computing lessons can become dull and dry
- Teachers are expected to set homework regularly and to provide meaningful feedback frequently to students
- It can be difficult to set homework tasks that rely on computers, particularly if some students do not have access to the technology required

The value of homework

I'll admit that, in the early part of my teaching career, I really struggled to see any value in setting homework. I have since changed my thinking. I have learned that, when managed well, an effective homework strategy can actually nurture a very positive attitude among students, as well as alleviating some of the constant pressures on teachers – creating the effect of having increased contact time in lessons for the things that matter most.

I realised that the secret to achieving a state of 'homework nirvana' relied on me, the teacher, making some bold decisions from the start, then establishing regular habits for my students to develop as well as some new habits for me too. The thing about habits is, once

you've really got into the habit of doing something regularly – you can do it repeatedly without reminder and without requiring a great deal of effort.

The simple explanation

Explained most simply, this homework strategy operates with three strong features:

- Homework list: Teachers share a long list of homework topics with their class in advance
- Pupil responses: Students respond by producing single-page summaries per topic, per week
- Regular checks: Peer checks regularly take place in class, measured using a rubric, and check-scores are recorded

This guide goes on to explain the strategy in more depth:

The homework list - a list of topic headings, shared in advance

The teacher starts by creating a full list of topic headings for the full year ahead, prepared long in advance of the lessons. While this might seem like quite an undertaking, if you share the task among your teaching colleagues you'll be able reduce the burden on yourselves and potentially have a more interesting and varied collection of homework topics.

Prescribe a single topic per week to your students, e.g. Week 1 – Topic 1, Week 2 – Topic 2, and so on. This means that, in a state school for example, there may be a total of 39 weekly topics, to take into account the number of weeks in an academic year. One interesting consequence of this approach is that your students will no longer have the excuse to inform you, "I was absent – so I didn't know what the homework was." If you receive requests to set work for students who are on long-term absences from school, you can simply suggest that they follow the homework programme at an appropriate pace.

■ Provide mini exercise books for students to write single-page summaries

Suggested examples of homework topics

For each topic, create a one-page summary that uses a combination of words and images to respond to the topic:

■ Suggest how to choose a strong password
■ Suggest how to select a suitable file name
■ Explain the difference between the World Wide Web and the internet
■ Describe the key features of a web browser
■ Explain what encryption is used for
■ Explain what a variable might be used for
■ Describe some advantages of a block-based programming language

> ## THE HOMEWORK STRATEGY CAN SERVE AS A USEFUL BACKUP STRATEGY

When creating your homework topic list, you could choose to follow a particular structure or have the topics follow certain themes related to areas of study, e.g. software, hardware, networks, programming, ethical issues.

It's important to recognise that the homework topics need not closely match the intended learning content of every lesson. It is considerably more difficult to plan for every homework topic to match the lesson topic for any given week, since that would require the

teacher to know a long time in advance what will be taught in each lesson. Instead, the topics could more broadly encompass the learning of the whole year, and may even touch on some topics that would not normally crop up in class.

There is no need to slavishly stick to the list every week. As the teacher, you always have the option available of deviating temporarily from the prescribed topic for a particular week. You may exercise this option when there is a more pressing need to set an alternative in order to reflect something that is current – but the list always provides the default topic.

In the very busy periods we often find ourselves in as teachers, the list becomes a real time-saver. By getting into the habit of setting every class their topic homework each week, you are far less likely to forget to set homework and will not struggle to choose an appropriate homework task under pressure. I would question whether the topic list needs to be that varied for different age groups – if the topic descriptions are written in a more general manner, they are more open to interpretation by learners.

A useful contingency

This homework strategy can serve as a useful backup strategy for when the planned taught lesson cannot proceed as intended. Imagine a scenario during which at the last minute, your Computing class have to be redirected to another classroom with no computers available. The homework strategy provides an alternative plan for a lesson until normality can be resumed. You might decide to lead a more traditional 'chalk and talk' lesson focusing on the current or next homework topic, while the issue keeping you out of your classroom is resolved. ▶

TRY TO FIND A SMART WAY TO COLLECT THE CHECK-SCORES

▶ Alternatively, if your travel to school has been disrupted or delayed, your colleagues could ask those students already in class to start on the week's homework topic until you arrive.

Ensure that the list is easily accessible

Take steps to share the list publicly with your students and others, where appropriate. By making the topic list easily accessible, it ensures there are a minimum number of barriers to prevent students from completing homework. Additionally, you can also share it with parents. If you host an online version or cloud-hosted document, e.g. a Google Doc, you can share the list easily and then use a custom URL, e.g. TinyURL or Bitly, to help keep it memorable.

An alternative to maintaining an online list may be to print copies out for students to glue into their homework books.

■ With a sound strategy in place, homework is no longer a headache

Provide exercise books for the purpose

Provide mini exercise books specifically for the purpose of homework. Lots of exercise options are available, including non-ruled. We chose a mini exercise book, A6 in size, which worked perfectly well. It was just compact enough to fit in a blazer pocket, and did not require a huge amount of effort from students to fill a single page.

While some computing teachers may baulk at the suggestion of using exercise books for homework, I found that it led to a more level playing field and reduced the chances of things going wrong. My experience has been that requiring all of the students I taught to all be able to use technology for the homework just created additional reasons for excuses, such as "I couldn't log in", "My device needed charging", "My laptop is broken", "I forgot the webpage", "My account has been suspended". I told them that if they didn't have their book in class, they would score zero in the peer checks.

Single-page summaries per topic

With this strategy, students are expected first of all to study the topic independently. Ideally, this activity takes place out of lesson during homework time. Students then summarise their findings using sketch notes. We required our students to create one single-page summary for each individual topic every week. We worked on the principle that if a student couldn't fit their summary on a single page, then they were doing too much or it was not an effective summary. It's important to stress that there is much more activity involved in the homework than simply filling a page, since some prior research is required. The single page summary is just one way of evidencing that the research has actually taken place.

In my experience, some teaching groups have grasped this approach more readily than others. When certain classes or students have required more support to achieve the desired standard, I've devoted some lesson time toward achieving that. This has included sharing examples of the expected standards.

Peer-check in class using a checking rubric

The trick to reducing the marking burden on the teacher is in using the skills of the teacher to do the things that only the teacher can do. While there is a lot of learning potential in requiring students to peer-mark each other's work, the additional bonus is that it saves the teacher from unnecessary marking. To keep the peer checks smart and efficient, we used a simple rubric that awarded a check score out of four, depending on the quality of the response.

The first time the peer checks take place, it will require a certain amount of explanation. After using this approach in lessons for a few weeks, you'll find that the peer-checking exercise runs like a well-oiled machine and can be conducted in around five minutes. You may choose to peer-check homeworks every week, or every fortnight, and do two together at the same time.

A simple example of a checking rubric:

1 mark for correctly recording the title and date of the homework

1 mark for written notes

1 mark for inclusion of diagrams/sketches

1 mark for exceptional presentation

0 marks for no evidence of homework

I would periodically take photographs of samples of responses for the purpose of having a gallery of examples to refer to.

ALAN O'DONOHOE

Alan (@teknoteacher) has more than 20 years experience teaching and leading Technology, ICT and Computing in schools in Northern England. He runs exa.foundation, delivering professional development to inspire and engage digital makers, support the teaching of Computing, and promote safe, secure and appropriate use of technology.

Scores are collected into the teacher's mark book. Try to find a smart way to collect the check-scores, using the fastest, most efficient manner available to you. You may need to experiment with a few different approaches until you find the technique that suits you best of all. The approach that suited me involved using some voting software which would display the number options on all screens; students would simply select the number that matched their check score. This took a manner of seconds and saved me manually entering the scores. This system also assisted me to administer the attendance register, since it doubled up as a speedy way for me to check who was in class.

Share progress on a scoreboard

There is a principle that 'teams perform better when someone is keeping score'. After I collected in the peer-check scores, I displayed the raw scores in a chart and shared this with the class. I told students that I would be happy with a score of 3 for any homework, where 4 was the maximum score. However, any homework summaries which scored 2 or less would be below the expected standard and as such need to be rectified before the previous lesson. In some cases, the issue could be remedied there and then, ready to be re-marked at a later date.

Identify issues early on and intervene

Once the teacher is in the habit of recording the peer-check scores regularly, it facilitates the reviewing of pupil progress. This means that issues can be identified early on and rectified. Some issues may require high levels of intervention, e.g. facilitating additional opportunities for students who are lagging behind expected completion and/or below the expected standards.

If you'd like some free, friendly advice about planning your computing homework strategy, contact author Alan O'Donohoe at **alan@exa.foundation**. (HW)

■ Voting software is used to display students' check-scores on all screens

YOUR QUESTIONS

Q WHEN MY STUDENTS DON'T SEEM TO HAVE FULLY GRASPED A CONCEPT, UNDER WHAT CIRCUMSTANCES IS IT BETTER TO MOVE ON OR STAY ON THAT TOPIC? **@JOSH_CHEEK VIA TWITTER**

A It depends on how much that topic underpins other parts of the curriculum you are trying to cover. The basic building blocks of programming will be used and reused time and again throughout everything you do, so there may be ample opportunity to offer recaps for students who need them without holding others back or forcing them to repeat material. Another strategy I like comes from experienced teacher Katie Vanderpere Brown, Director of IT at Saffron Walden County High School (@vanderpere) who recently tweeted, "I like to make connections between what they already know really obvious to students when I'm teaching theory. For example, teaching encoding links to binary and hex. If you weave the curriculum together, you are constantly reinforcing knowledge acquisition".

LAURA SACH (@CODEBOOM)
SENIOR LEARNING MANAGER, POST 16
RASPBERRY PI FOUNDATION

Q DO YOU HAVE ANY EXAMPLES OF EFFICIENT WAYS OF ASSESSING COMPUTING AT PRIMARY LEVEL WHICH DON'T REQUIRE COLOURING EXCEL CELLS OR KIDS COLOURING SMILEY FACES? **@COMPUTINGMISSA VIA TWITTER**
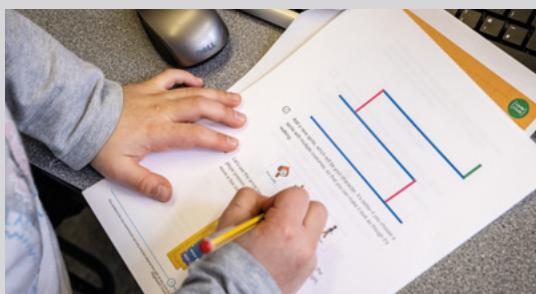
A I adapt the Project Quantum questions (**helloworld.cc/diagnosticquestions**) to create a quiz at the end of each six-week block of lessons. I take the questions and put them into a Google Form, to make sure that my students don't get lost in trying to navigate the website! As they're all multiple-choice questions, Google Forms automatically marks them and tells me how well the class is doing overall, with options to see individual child performance if necessary.
However, I would also suggest you think carefully about your motivations for assessment.

The Project Quantum questions are summative, and can only be used to inform planning on a year-by-year scale. Of course, alongside this there are regular formative assessment tasks, questioning strategies, and discussions. If you're looking at diagnostic assessment to inform planning, then you'd be better off planning activities at the beginning of a unit to know where to direct teaching.

I have found these can be as simple as a True/False or Always/Sometimes/Never statements such as 'A loop, or repetition, used in a program will always do something at least ten times.' The resulting discussion can often give you far more information than a written test.

SWAY GRANTHAM (@SWAYGRANTHAM)
LEARNING MANAGER, RASPBERRY PI FOUNDATION
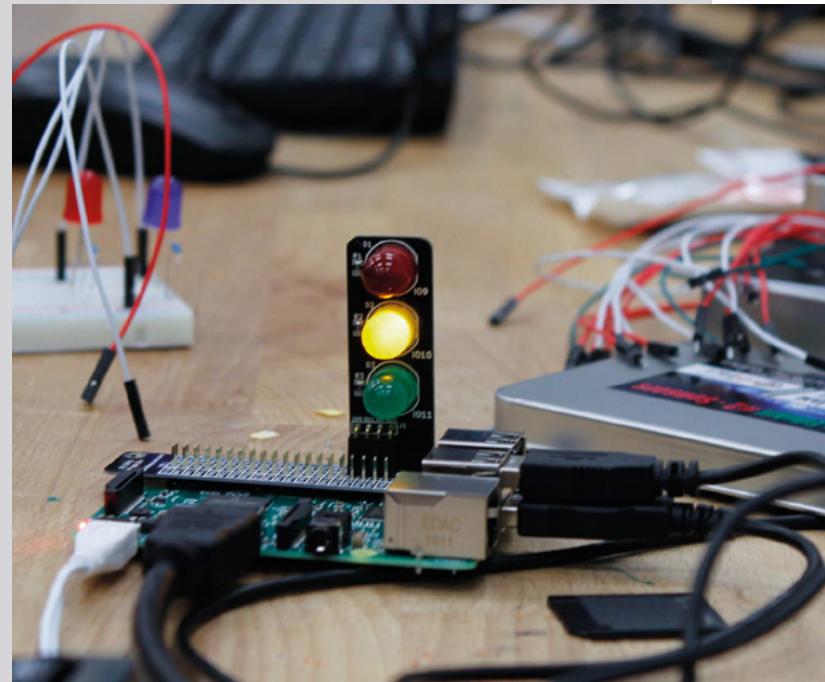
Q WHAT'S THE BEST WAY TO START PHYSICAL COMPUTING WITH MY CLASSES? WHAT KIT DO I NEED? WHAT EXPERIENCE? @LANCASTERSCICS VIA TWITTER

A Start simple and decide what your focus you want to have: is it the electronics, the design and making, or the programming? Being clear about the objectives will help select the right activity for your learners and context. While using individual components might help reinforce some electricity theory, it may overload the learners if your focus is the programming, and vice versa.

A good starting point might be something like the Robot Antenna project from the Raspberry Pi Foundation (**helloworld.cc/rpi_robot_antenna**). It provides a nice balance of electronics, making, and programming and can be easily adapted. The activity focuses on Scratch, but is equally easy using Python. And while the project uses the Raspberry Pi device, the same result can be achieved with other products.

You don't need any experience to begin with, just a willingness to experiment and to see what works for you. Adding any hardware to a project adds complexity and an extra dimension of debugging; however, the payoff is worth it. To develop your skills and confidence, why not take this online FutureLearn course? **helloworld.cc/physical_computing_course**

JAMES ROBINSON (@LEGOJAMES)
SENIOR LEARNING MANAGER, PEDAGOGY AND TRAINING
RASPBERRY PI FOUNDATION

Q I WOULD LIKE TO GET A UNIT OF WORK INTO YEAR 9 USING BLENDER AND/OR UNITY. DO YOU HAVE ANY ADVICE OR EXAMPLES? CAROLINE ELLIS

A Code Club has a whole sequence of Blender projects (**helloworld.cc/cc_blender**) which cover everything from manipulating basic pre-prepared models to creating and rendering animated scenes. If you're looking to go down the Unity route, CoderDojo have a good starting project (**helloworld.cc/cd_unity**) which helps learners understand the basics of controls and physics.

PHILIP HARNEY (@PHILIPHARNEY)
CONTENT LEAD AT CODERDOJO

YOUR LETTERS

Our letters page is a place for you to join our conversation. If you've got a comment, a question, or an announcement to share, contact us on Twitter via **@HelloWorld_Edu.** Alternatively, email us with 'Teacher Letter' in the subject line (**contact@helloworld.cc**).

Calorie counting

Dear Hello World,

I've just read the *Hello World* article about dieting with genetic algorithms (Issue 7, Page 64). I know it's not directly suggested as a student activity (and what it's trying to demonstrate is interesting), but I feel quite strongly that calorie counting is not an appropriate theme for programming activities. As a teacher, I have seen far too many students struggle with food issues and I think that needs to be kept in mind.

Name supplied

Thank you very much for your letter, you make an important point. When planning which articles and resources to feature in *Hello World,* we should always be mindful of the contexts in which they could be used by educators, and the myriad of pressures young people are under. This is something we will strive to bear in mind in future issues of the magazine.

SOLUTION TO THE PARSON'S PUZZLE ON PAGE 87

```
number = random_number()
guess = input("Guess a number")
while guess != number
    if guess > number then
        print("Too high")
    else
        print("Too low")
    endif
    guess = input("Try again")
endwhile
print("Well done, you guessed
correctly.")
```


We'll print another selection of your correspondence in *Hello World* 9. Until then, as always, thanks to everyone who's got in touch.

CREATIVE CODING
IN PYTHON

These beautifully presented projects were a hit with the
granddaughter of John Stout, our book reviewer

My six-year-old granddaughter, Amy, loves this book. It was the look of it that first got her attention, then the art project later on in the book, but the book starts with a brief introduction to coding. It guides you through installing Python, you write your first line of code, make your first (deliberate) error, and get quick introductions to pseudocode and flow charts.
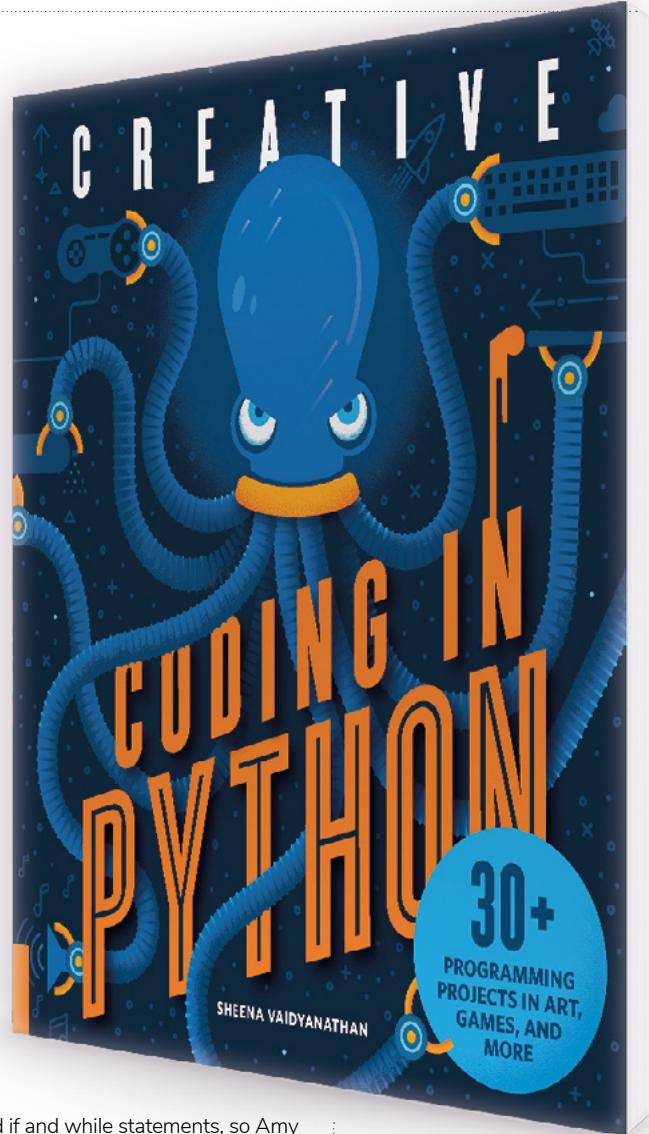
There are five chapters, each with its edges in a different colour. Every chapter starts with 'Big Ideas', e.g. variables, then a 'Project' based on the theme of the chapter and all the big ideas met so far. Importantly, each chapter ends with a 'Next Steps' section, where learners 'Experiment' and 'Extend' what they've done so far.

In the first chapter, 'Create your own chatbots', the Big Ideas are variables, inputting and outputting data, and comments. You then create a simple chatbot in five steps, testing the program entered so far at the end of each step (one of Kernighan and Pike's Testing rules!) so you don't get dispirited when faced with errors in the complete program.

A turtle called Shelly

The chapter 'Create your own art masterpieces' has turtle graphics (with a turtle called Shelly), for loops, and lists as its Big Ideas. Loops make programming much more attractive to Amy: so much for so little!

'Create your own adventure games' introduces Booleans, logical operators, and if and while statements, so Amy can first write a 'Guess the number' game with just one guess and feedback, then improve it.

The book ends with 'What's Next': MicroPython on the micro:bit, gpiozero on the Raspberry Pi, and an introduction to dictionaries, exception handling, files, and packages such as Pygame (this would be useful to teachers new to programming).

You can probably tell that Amy is not the only one who loves this book: it's a hit with both of us! **(HW)**

THE PRACTICE OF PROGRAMMING
A CLASSIC WORK ON PROGRAMMING WELL

INFO   **BY** Brian W Kernighan and Rob Pike | **PUBLISHER** Addison-Wesley | **PRICE** £18.00 | **ISBN** 978-0201615869 | **URL** helloworld.cc/kernighan
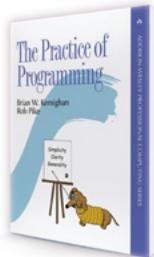
Published in 1999, this book assumes you can program, and will improve your programming in any language. It contains useful rules and 'war stories' from real programs the authors have worked on. Written by Brian W Kernighan, a contributor to the development of UNIX, and Rob Pike, a developer of the language Go, it concentrates on the underlying principles of simplicity (short and manageable programs), clarity (easily understandable programs), generality (programs that work across a range of situations), and automation (programs do the mundane work).

Chapters on Style, Algorithms and Data Structures, Design and Implementation, Interfaces, Debugging, Testing, Performance, Portability, and Notation are divided into sections, each introducing a rule with sample code; e.g. 'use descriptive names for globals, short names for locals'.

Most programmers will eventually discover many of these rules, but sharing them with learners new to programming will give them a head start.

See also *Elements of Programming Style* (1978, also by Kernighan; search YouTube for his *Computerphile* videos) and *The Art of Readable Code* (2011). Perhaps the rules need repeating every generation or so! **(HW)**

GAME CHANGER:
ALPHAZERO'S GROUNDBREAKING CHESS STRATEGIES AND THE PROMISE OF AI

INFO   **BY** Matthew Sadler and Natasha Regan | **PUBLISHER** New in Chess | **PRICE** £19.95 | **ISBN** 978-9056918187 | **URL** newinchess.com/game-changer

At first sight a specialist chess book, *Game Changer* gives insights into the game and how chess programs work – specifically AlphaZero, which learnt to play brilliant chess by playing 44 million games in just nine hours.

There's plenty to capture the interest of non-chess players, too. The book includes interviews with the developers of AlphaZero as well as Demis Hassibis, CEO of DeepMind.

The discussions around the similarities between AlphaZero's approaches and how grandmasters play chess are particularly fascinating.

AlphaZero has replicated many of the strategies of famous players, as well as discovering original approaches to playing chess. The same can be said of AlphaGo, an AI system which beat Go grandmaster Lee Sedol. AlphaGo is also the subject of a recent Netflix documentary. **(HW)**
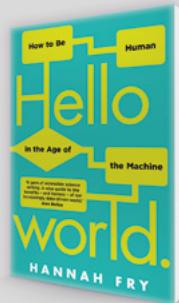
ESSENTIAL READING

Three more books on how technology is changing the way we see the world and interact with one another

HELLO WORLD: HOW TO BE HUMAN IN THE AGE OF THE MACHINE

BY Hannah Fry
PUBLISHER Penguin
PRICE £6.22
ISBN 978-0857525246
URL helloworld.cc/fry

Mathematician and broadcaster Hannah Fry takes us through some of the ways in which AI is already affecting human society. Topics include the justice system, medical diagnoses, and self-driving cars.

THE CREATIVITY CODE

BY Marcus du Sautoy
PUBLISHER Fourth Estate
PRICE £12.20
ISBN 978-0008288150
URL helloworld.cc/dusautoy

An exploration of some of the ways in which AI could demonstrate creativity, ranging from new moves in the ancient game of Go, through discovering and proving mathematical theorems, to musical composition.

OUTNUMBERED

BY David Sumpter
PUBLISHER Bloomsbury
PRICE £13.88
ISBN 978-1472947413
URL helloworld.cc/outnumbered

An accessible, but still somewhat technical, account of the algorithms underpinning social media and the sometimes unintended uses to which they're put. Includes some helpful worked examples.

MILES BERRY PRINCIPAL LECTURER IN COMPUTER EDUCATION

WHAT I'M THINKING ABOUT COMPUTATIONAL THINKING

More automated solutions, fewer jam sandwiches

Computational thinking is the golden thread running through England's computing curriculum, and has been a profoundly influential concept in computing curriculum design across the globe. There are, though, two quite different ways of thinking about computational thinking. The position you take on this isn't merely a philosophical one: it seems to impact directly on the 'what' and 'how' of computing education.

Firstly, there's the idea of taking the principles of computer science and applying them to other contexts. Jeannette Wing is credited with coining the term 'computational thinking'. Her original position, back in 2006 was:

"Computational thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science."

In this view of computational thinking, we might identify a number of the concepts that computer scientists and software engineers draw on in their work, and then teach pupils about these concepts: thus pupils might learn about algorithms through writing recipes for jam sandwiches, they might learn about decomposition by labelling the parts of a plant, they might learn abstraction through planning journeys on the underground, and they might learn to recognise patterns through spelling rules. With this interpretation of computational thinking, unplugged approaches seem to work well.

For me, though, this definition of computational thinking seems too vague to be helpful. There seems little here to set computational thinking approaches apart from those which we might label as mathematical reasoning, or engineering thinking. Without computation, this sort of computational thinking is really just thinking.

I think there's a better way. Recently, Jeannette Wing has given a more helpful definition of computational thinking, which sites it within the broader domain of problem-solving but, crucially, links this explicitly to computation:

"Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out."

I think the key distinction between computational thinking and other problem-solving approaches is that, with computational thinking, we should be searching for a solution that can be automated. Typically, this means that the solution should be one that can be implemented as code. By locating computational thinking as part of the programming process, it becomes a teachable, and assessable, thing. Computational thinking then becomes useful in solving the problems, or understanding the systems, that come up in the other subjects that students study. It is less about finding examples elsewhere in the curriculum to illustrate CS ideas, than about applying these ideas to write programs that solve problems from elsewhere in the curriculum.

While making jam sandwich recipes, labelling plant diagrams, planning tube journeys, and learning spelling rules are all well and good, they don't sit well with this view of computational thinking – in none of these cases do pupils take what they've done and implement it as code. Pupils are far more likely to learn computational thinking if they can use it to write programs to solve problems: you can't, for example, create Instagram-style filters, model the spread of an epidemic, or suggest a spelling correction without using computational thinking, and it's in looking for an automated solution that can be applied to any problem in the class that computational thinking becomes distinctive and useful. (HW)

MILES BERRY

Miles is principal lecturer in Computing Education at the University of Roehampton. He's a member of the Raspberry Pi Foundation, and serves on the boards of CAS and CSTA.

"HELLO, WORLD!"

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

Q WHAT IS HELLO WORLD?

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

Q WHO MAKES HELLO WORLD?

A The magazine is a joint collaboration between its publisher, Raspberry Pi, and Computing At School (part of BCS, The Chartered Institute for IT). *Hello World* is supported by Oracle.

Q WHY DID WE MAKE IT?

A There's growing momentum behind the idea of putting computing and digital making at the heart of modern education, and we feel there's a need to do more to connect with and support educators inside and outside the classroom.

Q WHEN IS IT AVAILABLE?

A Your 100-page magazine will be available five times per year.

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 or a bugbear you'd like to share? We'll feature your thoughts and ideas.

Tell us your story
Have you had a recent success (or failure) you think the wider community would benefit from hearing? We'd like to share it.

Write for the magazine
Do you have an interesting article idea or lesson plan? We'd love to hear from you.

GET IN TOUCH

Want to talk? You can reach us at:
contact@helloworld.cc

FIND US ONLINE

www.helloworld.cc

@HelloWorld_Edu

fb.com/HelloWorldEduMag

SUBSCRIBE IN PRINT TODAY!
PAGES 36-37

(Hello World)

helloworld.cc