(Hello World)

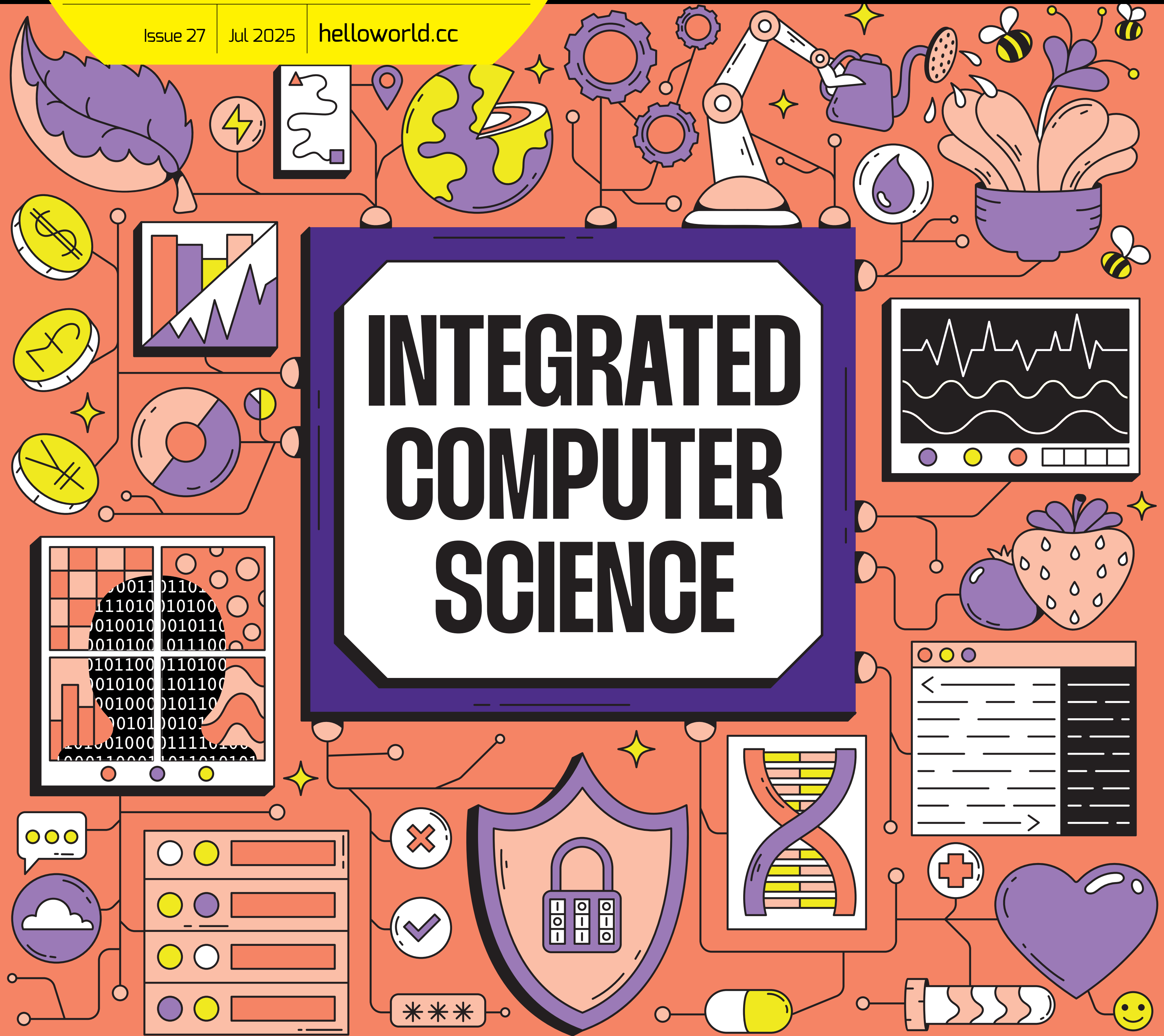
THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS

Issue 27 | Jul 2025 | helloworld.cc

LEARNING TO CODE IN THE AGE OF AI
Why we still need to teach children how to code

CS ACROSS THE CURRICULUM
The power of teacher collaboration

IDEAS FOR INCORPORATING CS
Approaches in dance, geometry, and modern languages



INTEGRATED COMPUTER SCIENCE

PLUS **EXPERIENCE CS** · AUTONOMY · **CODYSSI: A STUDENT-LED COMPETITION** · LANGUAGE, ABSTRACTION, AND AI IMPACT IN INDIA · **LEARNINGS FROM THE AI SYMPOSIUM** · MEET THE CODE CLUB · **COOLEST PROJECTS USA** · **PRIMMDEBUG** · SUPPORTING NEURODIVERGENT STUDENTS · **NEW QUICK READS** · STREAM ON A BUDGET

Raspberry Pi Foundation

Teach and learn with the Raspberry Pi Foundation

Free for everyone anywhere in the world

Teachers' resources

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

raspberrypi.org/teach

Project library

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

projects.raspberrypi.org

Research to practice

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

raspberrypi.org/research

Coding at home

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

raspberrypi.org/learn

Raspberry Pi Foundation, UK registered charity 1129409

HELLO, WORLD!

Here in England, computer science (CS) is taught as a mandatory stand-alone subject, but in many countries, it is integrated into the school curriculum. We've heard from educators who say that teaching computer science as part of other subject lessons can be overwhelming, especially if you don't have a CS background.

On the other hand, integrating computer science into other subjects can offer a more accessible entry point for young people to learn and engage with it — an approach that can broaden participation and can make computer science education more appealing. Also, as we know, computer science encourages critical thinking and problem-solving skills, and that can enhance the learning of any subject.

This issue features a range of practical articles with ideas for integrating CS over a variety of subjects at the primary, elementary, and high-school levels. On pages 28–29, James Abela shares how to code across the curriculum, using the concepts of sequence, selection, and repetition. Meanwhile, whether we can effectively teach two things at once is explored in Jane Waite's article on the Autonomy framework (pages 20–21).

In his article on pages 26–27, Jake Baskin says, "If you're a teacher who is implementing CS principles in your classroom, you are a computer science teacher." Whether you are trained to be a computer science educator or not, tell us more about how you teach computer science. We'd love to hear from you!

NEW PODCAST
TURN TO PAGE 43

Meg Wang
Editor

FEATURED THIS ISSUE

LAURIE GALE

Laurie is a third-year PhD student at the Raspberry Pi Computing Education Research Centre in the UK. On pages 18–19 he introduces PRIMMDebug, a new way of teaching debugging.

JESUALDO MARTÍNEZ MOLINA

Jesualdo is an assistant head teacher at The Premier Academy, Bletchley, UK. He presents practical examples of weaving computing into Spanish on pages 40–41.

TIFFANY JONES

Tiffany is a high-school computer science and cybersecurity teacher in Georgia, USA. She shares how she ensures that students can see themselves in tech on page 82.

(HW)

Hello World is the official magazine of the Raspberry Pi Foundation

Raspberry Pi Foundation

EDITORIAL

Editor
Meg Wang

Contributing Editor
Dominick Sanders

Subeditors
Louise Richmond and Gemma Coleman

Subscriptions
Sean Sayers

Social Media
Sean Sayers

DESIGN

criticalmedia.co.uk

Designer
Lee Allen

Photography
The Raspberry Pi Foundation, Adobe Stock

Graphics
Rob Jervis

Cover
© Veronika Kiriyenko

CONTRIBUTORS

James Abela, Leah Aiwohi, Sophie Ashford, Jake Baskin, Hedy Brown, Justine Chavez-Crespin, Philip Colligan, Michael Conterio, Fiona Coventry, Andrew Csizmadia, Catherine Elliott, Laurie Gale, Tracy Gardner, Mark Griffiths, Seung Woo (Tony) Han, Teresa Harris Boag, Graham Hastings, Dan Jones, Tiffany Jones, Patrick Kennedy, Janine Kirk, Kat Leadbetter, Sam Lovatt, Sofia Mohammed, Jesualdo Martínez Molina, Rujeko Moyo, Greg McDonough, Rajchid (Pip) Purttiponthanee, James Robinson, Sue Sentance, Zahra Shelke, Alex Sinclair, Bryan Twarek, Peter Wairagu, Jane Waite, Claire Walmsley, Joee Winter, Kay Wojnarski, Seema Zerafa

Contributing Partner
bcs Academy of Computing
CAS

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

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

(HW) CONTENTS

26–42 COVER FEATURES

INTEGRATED COMPUTER SCIENCE
Cross-curricular CS integration, whether you're a computing specialist or not

NEWS, FEATURES, AND OPINION

6 NEWS
Introducing Experience CS, impact in Telangana, Coolest Projects USA showcase, digital literacy in Kenya, new AI Pedagogy Quick Reads

16 BENEFITS OF INTEGRATED COMPUTING FOR ALL
Integrating CS can help make learning more inclusive for a wide range of learners

18 PRIMM DEBUG
A new way of teaching debugging

20 AUTONOMY
A framework for cross-curricular analysis

22 WHY KIDS STILL NEED TO LEARN TO CODE IN THE AGE OF AI
The necessity and benefits of teaching young people to code even with the advancements of AI

24 SUBSCRIBE

26 INTEGRATED COMPUTER SCIENCE

26 INTEGRATING CS
The power of teacher collaboration and the community of CSTA

28 PRIMARY CODING ACROSS THE CURRICULUM
Coding ideas for different school subjects

30 ELEMENTARY SCHOOL CODING AND ROBOTICS
How one school has incorporated coding and robotics into general-education classrooms

32 FROM SILOS TO SEAMLESS
How CS helps students connect the dots in every classroom experience

34 REIMAGINING CS REQUIRES INTEGRATION
How to integrate foundational and advanced CS content at the high-school level

36 DANCE AND COMPUTER SCIENCE
Building real-world skills across the arts, media, and digital technology

38 INTEGRATING SCRATCH INTO GEOMETRY
Reflections on incorporating Scratch projects as assessments in an advanced geometry class

40 CODING IN SPANISH
Cross-curricular computing and modern foreign languages in the primary classroom

42 AI IS EVERYWHERE – EXCEPT THE SCHOOL CURRICULUM
An opinion on why AI requires a more integrated and interdisciplinary approach

52
GIRLS IN CS
Small steps to enhance girls' interest in computer science

© Sanele Gobinduku/peopleimages.com/stock.adobe.com

62
GENAI IN PRIMARY SCIENCE
The findings of a small-scale research study

© Seventyfour/stock.adobe.com

54
EXPERIENCE AI
Connecting global communities

© Latvian Internet Association

44 CODYSSI: AN INTERNATIONAL STUDENT-LED CODING COMPETITION
Hear from a student on his inspiration for creating a competition, and from his teacher on how to support exceptional students

46 PROGRAMMING IN PARADISE
Teaching computing in rural Kaua'i

48 CODE CLUBS IN SOUTH AFRICA
Contexts, challenges, and insights from visiting Code Clubs in Durban and Cape Town

50 ADVOCATING FOR NEURODIVERGENT STUDENTS IN CS EDUCATION
Recent initiatives to support diverse talents in computing

52 HERE COME THE GIRLS!
Tips for making computer science more appealing to girls

54 CONNECTING GLOBAL COMMUNITIES WITH EXPERIENCE AI
How education can serve as a tool to connect communities in understanding and engaging with AI

LEARNING

RESOURCES & LESSON PLANS

59 GETTING THE MESSAGE ACROSS
A lesson plan for exploring the features of clear messaging in digital media

62 USING GENAI TO PLAN PRIMARY SCIENCE
Findings of a small-scale research project on using ChatGPT to provide ideas for teaching primary science

66 SMARTER IRRIGATION FOR A THIRSTY WORLD
A classroom-ready, project-based learning activity for building an automated irrigation system

69 A WHOLE-SCHOOL APPROACH TO AI
How schools can engage parents, trustees, and school staff in developing an effective AI strategy

CONVERSATION

72 STREAM MAGIC ON A BUDGET
How embedding STREAM into the curriculum is transforming learning and boosting creativity

75 BEBRAS
A fun computational thinking challenge

76 MEET THE CODE CLUB
A conversation with Code Club Collège Jeanne-Sauvé in Winnipeg, Canada

78 LEARNINGS FROM THE 'TEACHING ABOUT AI' SYMPOSIUM
Navigating the world of AI without a map

80 EVOLVING COMPUTING
Comparing language acquisition in children and in large language models

82 BELONGING IN TECH
Connecting culture, creativity, and careers in the CS classroom

EXPERIENCE CS: A NEW WAY TO TEACH COMPUTER SCIENCE

Sofia Mohammed, US executive director at the Raspberry Pi Foundation, explains how Experience CS integrates computer science concepts into subjects like maths, science, social studies, and the arts in a standards-aligned curriculum

Sofia Mohammed

Experience CS is a brand new, free, integrated computer science curriculum for elementary and middle school educators (working with students ages 8 to 14). Launched in June, one of the most important design principles for Experience CS is that it can be used by any educator. You don't need a computer science qualification or previous experience in teaching CS classes to deliver engaging and creative learning experiences for your students.

■ Educators can use Experience CS to teach computer science through an integrated curriculum, and all for free

A cross-curricular approach

Experience CS enables educators to teach computer science through a curriculum that integrates CS concepts and knowledge into core subjects such as maths, science, and social studies. This cross-curricular and integrated approach is one of the most effective ways of providing younger students with an introduction to computer science and is increasingly important as the impact of digital technology reaches every corner of our lives. We know that embedding CS in real-world contexts helps make it meaningful and relevant for students, which is essential if we are going to inspire young people from different backgrounds to want to learn more about computer science and technology.

There are six units currently available to download, with more on the way. The lessons are aimed at learners in grades 3 to 8 (ages 8–14). Each unit provides guidance on the related subject areas. For example, in 'The me project', grade 4 students (ages 9–10) explore the basics of Scratch, personalise sprites, and develop programs to create an animation that tells a story all about them. This could be integrated into language arts lessons, enabling young learners to explore visual representation and write their own unique story. In the 'Smart communities' unit, students in grade 6 (ages 11–12) explore ways in which computing and technology can be used to create environments that are responsive to the needs of community members; this could be included within science or technology lessons.

Each of the six units consists of an overview giving a summary of the topics covered, as well as a series of six to eight downloadable lessons, with the last lesson providing a chance for students to showcase projects to their classmates. Students will learn to code their own programs, make exciting projects, and build their computational thinking skills — all taught through the context of core subjects.

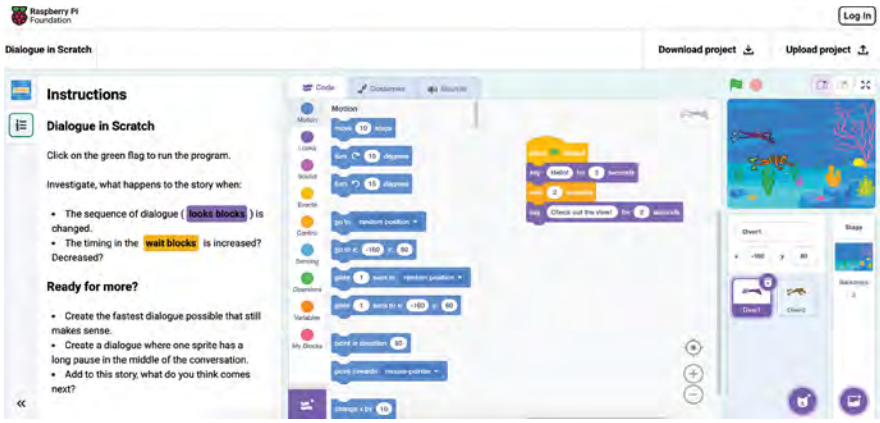
Ashly Tritch, computer science immersion specialist at Olson Middle School in Bloomington, MN, USA said, “Cross-curricular computer science is important because it shows students how coding and tech skills can be used in other subjects like math, science, or even art. It helps make learning more interesting and helps kids understand how computer science connects to real life. The lessons that the Raspberry Pi Foundation is creating will be super engaging, with fun and creative activities that keep students curious and excited to learn.”

Initially, the curriculum and resources have been mapped to national and local standards in the US and Canada, including the CSTA K–12 Computer Science Standards, but they are available for teachers and students anywhere in the world to use for free.

DESIGNED WITH SCHOOLS IN MIND

We have designed every part of the Experience CS platform with school environments in mind, making it easier for teachers to manage and for students to use.

- **Fully integrated platform:** everything students need is built into the Experience CS platform, including Scratch, lesson resources, student materials, and project templates. There is no need to visit other websites.
- **Simple access:** teachers generate class codes so learners can jump straight into activities, with no student email address required.
- **Automatic progress saving:** students’ projects are saved in the platform and linked to their class.
- **Teacher control:** teachers have full visibility of student activity, and what students see and do stays within the classroom environment.



Experience CS runs in a closed, classroom-ready environment that supports safeguarding policies and fits with school filtering systems

A safe, creative way to teach computing

Experience CS has been built from the ground up to support safe, confident computing lessons in real classrooms. It includes self-directed creative projects using the popular programming language Scratch. We have built a version of Scratch especially for schools. That means it doesn’t have the community and sharing features that are central to the full Scratch platform. Instead, everything runs in a closed, classroom-ready environment that supports safeguarding policies and fits with school filtering systems. Simple and intuitive learning management features enable teachers to create accounts, set assignments, and review progress.

No matter your experience or skill level, the Experience CS content has been designed to be easy to use. However, we do also offer professional development (PD) opportunities, to help build confidence in teaching computer science. This is initially focused on supporting schools in the US and Canada, including working with a fantastic network of educational partners. We also offer free online courses on creative ways to teach programming and Scratch, plus accessible summaries of our research-backed teaching approaches (Quick Reads, see page 14) to enable you to learn more about the pedagogy behind the Experience CS curriculum.

> “ EMBEDDING CS IN REAL-WORLD CONTEXTS HELPS MAKE IT MEANINGFUL AND RELEVANT

Help shape Experience CS

The team behind Experience CS includes educators with significant experience of teaching CS in elementary and middle school settings, and is based on extensive classroom testing and research. We will continue to develop and improve the curriculum and resources in response to feedback from teachers and students. If you would like to help shape the future of Experience CS by testing new features and providing valuable feedback to improve the programme, sign up via the website: **rpf.io/exp-cs-hw27**.

You can also get a head start, ready for the next school year by visiting **rpf.io/exp-cs-hw27** to register for a free account and download lesson resources. (HW)

IMPLEMENTING A COMPUTING CURRICULUM IN TELANGANA

A partnership in action

Fiona Coventry

In Hello World issue 23, Mamta Manaktala wrote about the Raspberry Pi Foundation's partnership with the Telangana Social Welfare Residential Educational Institutions Society (TGSWREIS) in Telangana, India (helloworld.cc/23, pages 56–57). The primary objective of TGSWREIS is to provide quality education to under-resourced young people, particularly children from India's most disadvantaged socioeconomic groups (helloworld.cc/TGSWREIS).

Since our September 2023 launch, we've been working with TGSWREIS to develop and implement a computing curriculum at two TGSWREIS locations — the Coding Academy School and the Coding Academy College. When our partnership began, there were around 800 students being taught across both institutions, including grades 6 to 12 in the school, and undergraduate students in the college. However, the school now has grades 6 to 9 only.

Our impact team is conducting an evaluation to gain a deeper understanding of the tangible effects of our work and to identify areas that may require further attention. Read on to find out more about the partnership and what we've learnt from our 2023–2024 evaluation.

Aim of the partnership

The aim of our partnership is to enable students in the school and undergraduate college to learn about coding and computing by providing the best possible curriculum, resources, and training for teachers.

Students working together at the Coding Academy School

The partnership is strategically important for us at the Raspberry Pi Foundation because it helps us to test curriculum content in an Indian context, and specifically with learners from historically marginalised communities with limited resources.

Adapting our curriculum content for use in Telangana

Since our partnership began, we've developed curriculum content for students at the Coding Academy School, which is in line with India's national education policy requiring coding to be introduced from grade 6 (age 11). We've also developed curriculum content for the undergraduate students at the Coding Academy College.

In both cases, the content was developed based on an initial needs assessment. We used the assessment to adapt content from our previous work on The Computing Curriculum (**helloworld.cc/the-computing-curriculum**). Local examples were integrated to make the content relatable and culturally relevant for students in Telangana. Additionally, we tailored the content for different lesson durations and to allow a higher frequency of lessons. We captured impact and learning data through assessments, lesson observations, educator interviews, student surveys, and student focus groups.

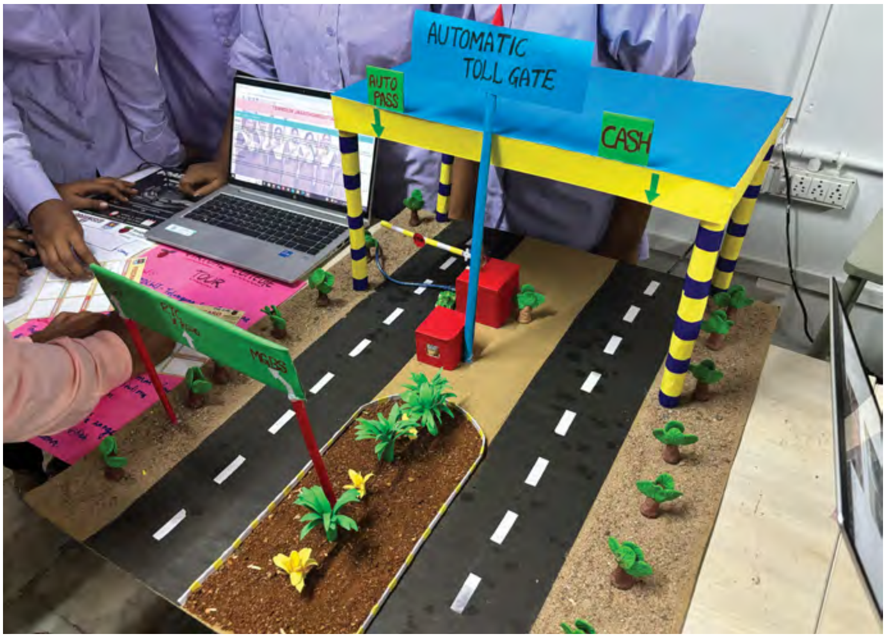
Curriculum well received by educators and students

We have found that the partnership was succeeding in meeting many of its objectives. The curriculum resources have received lots of positive feedback from students, educators, and observers.

In our 2024 survey, 96 percent of school students and 85 percent of college students reported that they've learnt new things in their computing classes. This was backed up by assessment marks, with students scoring an average of 70 percent in the school and 69 percent in the college for each assessment, compared to a pass mark of 40 percent. Students were also positive about their experiences of the classes, and particularly enjoyed the practical components — a third-year undergraduate student at the Coding Academy College said, "My favourite thing in this computing classes [sic] is doing practical projects. By doing [things] practically we learnt a lot." A student focus group facilitator at the Coding Academy School explained, "Since their last SA [summative assessment] exam, [grade 9] students have learnt spreadsheet [concepts] and have enjoyed applying them in activities. Their favourite part has been example codes, programming, and web-designing activities."

However, we also found some variation in outcomes for different groups of students and identified some improvements that are needed to ensure the content is appropriate for all. For example, educators and students felt that improvements were needed to the content for undergraduates specialising in data science — there was a wish for

The entrance to the coding lab at the Coding Academy School



A project by the undergraduates at the Coding Academy College

the content to be more challenging and to prepare students more effectively for the workplace. Some amendments have been made to this content and we will continue to keep this under review.

In addition, we faced some challenges with the equipment and infrastructure available. For example, there were instances of power cuts and unstable internet connections. These issues have been addressed as far as possible with Wi-Fi dongles and with educators adapting their delivery to work with the equipment available.

Our ambition for India

Our team has made some improvements to our curriculum content in preparation for the subsequent academic years, and we will make further improvements based on the feedback received.

The long-term vision for our work in India is to enable any school in the country to teach students about computing and creating with digital technologies. Over our five-year partnership, we plan to work with TGSWREIS to roll out a computing curriculum to other government schools within the state, and have now begun training teachers in other Telangana schools.

Through our work in Telangana and Odisha (**helloworld.cc/26**, pages 26–27), we are learning about the unique challenges faced by government schools. We are designing our curriculum to address these challenges and ensure that every student in India has the opportunity to thrive in the twenty-first century. If you would like to know more about our work and impact in India, please reach out to us at **india@raspberrypi.org**. (HW)

Students at the Coding Academy College listening to a presentation

Mila's 'Sports Compilation' project

CELEBRATING YOUNG INNOVATORS AT COOLEST PROJECTS USA

Highlights from Coolest Projects USA 2025

Sophie Ashford

In April earlier this year, the Science Museum of Minnesota buzzed with excitement as young tech creators from across the country came together for a celebration of creativity, curiosity, and coding in the form of Coolest Projects USA.

Around 40 young people showcased their projects, sharing their work with friends, family, and the wider coding community. With hands-on tech activities, project demonstrations, fun swag, and certificates of achievement, the day was packed with energy and enthusiasm.

Coolest Projects is all about empowering young people to show off what they have built, no matter their skill level or experience. From first-time coders to budding engineers, everyone was welcomed to the event and learnt something from the day.

Celebrating impact: Broadcom Foundation Coding with Commitment®

One of the standout moments was the presentation of the Broadcom Foundation Coding with Commitment® special recognition prize, which highlights a project that uses code to solve real-world problems.

The award went to Kaila for her project 'Car vs Road', a self-driving car simulation designed to explore how natural disasters

Coolest Projects USA at the Science Museum of Minnesota

Kaila's project won the Broadcom Foundation Coding with Commitment® award

like hurricanes, floods, and tornadoes might impact autonomous vehicle technology. Kaila explained, "My project was inspired by Waymo. After hearing about Coolest Projects through the Minnesota State Science and Engineering Fair, I decided to attend — and absolutely loved it! The atmosphere was welcoming, the judges were so involved, and the entire experience was truly unforgettable."

Broadcom Foundation's partnership with Coolest Projects USA highlights how coding can be a powerful tool for tackling global challenges, from health to climate change, and champions digital literacy for all youth.

Celebrating creativity across categories

With seven categories available for entries, the judges had their work cut out for them choosing their favourite projects. Here are some that particularly captured their attention on the day.

Mila's 'Sports Compilation' Scratch project

Mila's project, 'Sports Compilation', tells a story through animated sports scenes that inspire people to try new things and embrace their rebellious sides.

"I wanted to show my mom that I could code — even though I'm not a full-time coder. I kept working on my original sports scene and added more to create something bigger.

"My biggest challenge was making it reset automatically. At first, I didn't understand the instructions I found online, but then I figured out how to use timing and broadcasts to make it work.

"Coolest Projects was so fun. My teacher hyped it up, and it lived up to the excitement!"

Amazing young innovators at Coolest Projects USA

Ethan's 'Coding AI to Enhance Speech Therapy' project

Ethan's project was inspired by his experience with speech therapy. He used machine learning to help people identify and correct different incorrect pronunciations of the 's' sound. Ethan shared his motivations behind attending:

"I went to Coolest Projects USA to get the opportunity to present my work and receive feedback from a large audience of judges, participants, and other viewers on how to improve my project and the presentation of it.

"I had fun at Coolest Projects USA, seeing the different ways that young people like me are taking advantage of technology to try and help the lives of others, and also seeing the creativity and determination that we have."

A big thank you

We want to say a huge thank you to every young creator who presented a project, to the families who cheered them on, and to our amazing volunteers and partners. We also want to give a special shout-out to our sponsors, Broadcom Foundation and Best Buy Inc., for their incredible support. We couldn't have done it without you.

Thank you for making Coolest Projects USA 2025 such a special day! (HW)

CSTA 2025

Connecting with educators from across the US

Hedy Brown

We always get so much from attending the CSTA conference in the USA. From inspiring talks, to participating in engaging workshops, to meeting Hello World writers, connecting with and learning from educators is what we love to do.

Of course, we also want to share our latest resources to support your work. This year, the team from the Raspberry Pi Foundation will be presenting three sessions.

Engaging students' interest in CS using 3...2...1...Make!

Breakout session, room 21
8 July 2025, 3.30 pm–4.30 pm ET
Discover how educators are engaging student interests and empowering them to create with digital technologies in Code Clubs all over the world using the 3...2...1...Make! model. Learn how this model, developed from research by the Raspberry Pi Foundation and their work within the Code Club and CoderDojo communities, was created and how it works. See how educators are using Code Club resources, designed using the 3...2...1...Make! model, in their classrooms and coding clubs to help their students invent projects that matter to them (**helloworld.cc/3-2-1-guide**).

Experience CS: a new way to teach computer science

Breakout session, room 21
9 July 2025, 9.00 am–10.00 am ET
Experience CS is the Raspberry Pi Foundation's free curriculum designed to integrate computer science into other content areas across grades 3–8. This curriculum is designed to support teachers with ready-to-use learning materials, a standalone version of the Scratch platform in our Code Editor for Education, and research-backed computing pedagogical practices. Educators will explore the unique features of Experience CS and identify their own opportunities to integrate computer science into their work. To read more, turn to pages 6–7 or visit **experience-cs.org**.

Sprouting fresh ideas with hydroponics and micro:bits

Workshop, room 5
11 July 2025, 12.00 pm–3.00 pm ET
In this hands-on workshop in collaboration with Northern Lights Collaborative for Computing Education, participants will explore an innovative approach to integrating computer science into elementary and middle-school classrooms by building and programming a microchip-controlled hydroponic system, the HydroBitBucket. Using a micro:bit as the control unit, participants will program the system with MakeCode to collect real-time data through sensors and control the light source and water pump. This project demonstrates how to connect computing to real-world applications, offering students an authentic and engaging way to explore computer science while investigating the biology of plant growth.

Whether you're attending the conference or reading this from elsewhere, check out our free resources to support your journey as a computer science educator at **raspberrypi.org** or get in touch at **rpfna-team@raspberrypi.org**. (HW)

The team at CSTA 2024 in Las Vegas

HEAR MORE FROM CSTA TEACHERS ON OUR PODCAST WATCH THE EPISODE HERE
helloworld.cc/pod27-tips

DIGITAL LITERACY

Empowering Kenya's future

Peter Wairagu

At the Raspberry Pi Foundation (RPF), our mission is to put the power of computing and digital making into the hands of people all over the world. We believe that digital literacy is a fundamental skill that can unlock immense potential and opportunity. This belief is what fuels our commitment to supporting digital literacy programmes globally, and we are incredibly proud of the progress being made in Kenya.

In Northern Kenya, we have partnered with the Frontier Counties Development Council (FCDC; **fcdc.or.ke**), a consortium that brings ten Northern Kenya county governments together from Lamu, Tana River, Garissa, Wajir, Mandera, Marsabit, Isiolo, Turkana, Samburu, and West Pokot.

Turkana

In Turkana County, over 8000 learners from fourth to ninth grade (ages 9–14) will be equipped with digital literacy skills, including coding, through the Advancing Computer Skills in Schools Project, which includes 12 primary and junior schools in the county.

The project began in July 2024 with ten schools in Lodwar. As of the beginning of the second term in May 2025, the project has expanded to include two new schools and five additional teachers.

West Pokot

Training sessions also began in West Pokot in May 2025. Using community trainers, these sessions aim to train teachers to integrate digital tools into education and everyday problem-solving. Expert facilitators provide technical support, mentorship, and align the training with international best practices.

The programme covers areas such as basic computer science, coding, and integrating ICT in the teaching of non-computing subjects. By introducing technology in a hands-on, accessible way, the initiative aims to nurture a new generation of innovators, create pathways to STEM subjects, and support FCDC's broader mission of fostering resilient, self-reliant, and digitally inclusive communities in Kenya's frontier regions.

Laying the foundation

We've been working with the FCDC for over a year now to understand how to better serve the communities in Kenya. Feedback from young people during learner focus groups has shown us the potential for everyone in Kenya to develop digital skills:

"In my computing lessons, I've learnt how to use a computer safely and properly. I learnt how to type, use the mouse, and open programs. We also learnt about coding, which is really fun because we can make things happen on the screen by giving the computer instructions. I've also learnt how to create a simple document using software like excel sheets. I really enjoy using the computer to solve problems and make things work".

Through this partnership, the Raspberry Pi Foundation and the FCDC are not only enhancing educational outcomes but are also laying the foundation for long-term economic development, social inclusion, and technological empowerment across some of the most historically underserved areas of the country. (HW)

Rachel Bennett, RPF managing director, and Peter Wairagu, RPF country manager, Kenya, present certificates to the new community trainers

AI PEDAGOGY QUICK READS

New, free, short guides on AI in computing education

James Robinson

What does outstanding computing education look like in the age of AI? The Raspberry Pi Foundation has released a new series of Pedagogy Quick Reads exploring this vital question. Focusing on three aspects of AI in computing education — computational thinking (CT) 2.0, anthropomorphism, and feedback literacy — these short guides offer practical insights and new strategies for your classroom practice.

Introducing our AI Pedagogy Quick Reads

Each Quick Read is designed to help educators explore, understand, and apply one area of research evidence. They include:

- An introduction to the topic or idea, putting it into context
- A summary of the key concepts and takeaways for educators
- Sections elaborating on each key concept and relevant research
- A diagram presenting the same ideas in visual form
- Links to resources for further reading

The feedback literacy Pedagogy Quick Read

Computational thinking 2.0

"Without CT 2.0, today's learners will remain passive consumers rather than informed participants in a world increasingly shaped by data-driven AI technologies."

This Quick Read explores how the concept of CT is evolving, particularly in the context of AI. It offers guidance on how to teach CT skills that are relevant to, and enhanced by, AI technologies (**helloworld.cc/computational-thinking-quick-read**).

> WHAT DOES OUTSTANDING COMPUTING EDUCATION LOOK LIKE IN THE AGE OF AI?

Anthropomorphism

"If young people see this technology as innately human-like, we run the risk of impacting their … sense of agency … safety … social connection … curiosity."

As AI becomes widely used, it's important to consider how students understand and view these technologies. This Quick Read discusses anthropomorphism (attributing human-like qualities to AI) and provides strategies for avoiding some of the common problems around viewing AI systems as human (**helloworld.cc/anthropomorphism-quick-read**).

Feedback literacy

"How do we ensure that all students get the most out of AI system-produced feedback? Feedback literacy is a theory-driven framework that can help … answer this question."

Effective feedback is important for student learning, especially in a rapidly changing field such as AI. This Quick Read examines how to develop feedback literacy in both educators and students, enabling them to give, receive, and use feedback more effectively (**helloworld.cc/feedback-literacy-quick-read**).

More Pedagogy Quick Reads

This new AI series joins our bank of existing Pedagogy Quick Reads covering a wide range of topics related to computing education. These resources are based on up-to-date evidence of the best ways to teach and learn computing, all organised into twelve pedagogy principles for computing education (**helloworld.cc/pedagogy-principles**). You can find these resources and more on our pedagogy page (**helloworld.cc/pedagogy-resources**). (HW)

HEAR MORE FROM JAMES ON OUR PODCAST, WATCH THE EPISODE HERE
helloworld.cc/pod27-code

COLUMN

BENEFITS OF INTEGRATED COMPUTING FOR THE INCLUSIVE CLASSROOM

Integrating computing into other subjects can benefit a diverse range of students and help make learning more inclusive

There are a number of opportunities for bringing other subject areas into computing lessons, and inversely, teaching computing and computer science concepts and skills through other subjects. This is fairly straightforward at the primary level (ages 5–11) or in special schools if the same teacher delivers all of the subjects, but it can be more challenging at the secondary level (ages 11–18). However, it is worth taking time to explore opportunities for integrated computing and computer science, as there are a number of benefits in terms of promoting an inclusive classroom, from practising and consolidating key vocabulary to harnessing authentic contexts and providing multiple ways for students to express their learning.

Consolidating key vocabulary

There are a large number of technical terms that students need to understand and use in computing, from 'algorithm' to 'zip file', which can be overwhelming to learners with English as an additional language (EAL), as well as those with additional learning needs and disabilities. This essential subject-specific vocabulary, often referred to as 'tier 3 words', should be taught explicitly to provide a foundational understanding and to avoid misconceptions. A number of approaches to doing this are outlined in a blog by teacher and writer Alex Quigley (**helloworld.cc/vocabulary-pillars**). Having the opportunity to hear and use the key vocabulary repeatedly, and in different contexts in other areas of the curriculum, will help learners to remember the terms and build a more robust mental model of concepts and how they are connected. In primary lessons, you could ask pupils to write 'algorithms' to explain how to solve maths problems, or 'debug' spelling and grammar in sentences in English lessons after covering these concepts in computing. In secondary schools, consider sharing a list of key words and definitions with colleagues in other subject areas who are using technology in their lessons or teaching computer science, to ensure consistent use.

Catherine leads the new Computing & Digital Innovation Centre in Sheffield, UK (**helloworld.cc/learn-sheffield**), supporting schools with computing and the wider implementation of educational technology. She has over 20 years' experience of working with students with special educational needs and disabilities and their teachers, and is passionate about making computing accessible and inclusive for all learners.

Authentic, real-world examples

I have written about the Universal Design for Learning framework from CAST previously (**helloworld.cc/CAST**). These guidelines offer a set of concrete suggestions that can be applied to any discipline or domain to ensure that all learners can access and participate in meaningful, challenging learning opportunities. One of the guidelines looks at increasing engagement in learning by optimising relevance, value, and authenticity (**helloworld.cc/welcoming-interests**). Authentic examples that arise from working in a more integrated, cross-curricular way can engage learners through solving real-world problems, harnessing culturally relevant contexts, and allowing them to recognise the breadth of computing-related career pathways. These aspects benefit all students, but in particular can help us to improve access to the subject for traditionally underrepresented groups.

© Seventyfour/stock.adobe.com

A diverse range of students can benefit from the integration of computing into other subjects

Examples of authentic, real-world projects:
Modern languages: create a bespoke translation or assistive technology tool using the translation and text-to-speech extensions in Scratch (see an example here: **helloworld.cc/scratch-french**).

Physical education: explore and discuss the role of machine learning and AI in detecting injuries in elite athletes (see an interesting article that explores the benefits and limitations of this here: **helloworld.cc/ai-sports-injuries**), then use the machine learning tool developed for the micro:bit to analyse movement in a particular sport (**helloworld.cc/microbit-ml-tool**).

Art: use the Processing programming language (**processing.org**) to design and code generative art, and discuss the ethics around the use of AI to create art.

Music: program complex compositions using a text-based language in Sonic Pi (**sonic-pi.net**) or use Chrome Music Lab's Song Maker with younger learners (**helloworld.cc/chrome-music-maker**).

History: investigate the role of Alan Turing and Bletchley Park during the Second World War and set students code-breaking activities linked to cybersecurity. Teaching London Computing has a number of activities here: **helloworld.cc/code-cracking-puzzles**.

Multiple means of action and expression

Another principle of universal design is to design multiple means of action and expression, to allow learners flexible ways of expressing what they have learnt in the most effective way. Students who are slow writers due to physical difficulties, or who struggle with spelling and grammar, may produce better work when they can record themselves speaking or use a more creative output. Technology is a great tool for this, and skills learnt in computing lessons can be practised and used in other curriculum areas to produce meaningful outputs. For example, primary pupils can create a stop-motion animation to illustrate the life cycle of a frog or retell a Greek myth. In secondary geography lessons, students can create a multimedia e-book to present information on an area being studied. Students can design and create websites or apps as revision tools in religious studies, or program a quiz to practise maths problems.

Integrating computing across the curriculum offers opportunities to foster inclusive practice by reinforcing key vocabulary, engaging students with real-world contexts, and offering flexible ways to demonstrate learning. This can support learners with a wide range of additional needs, as well as including learners who may not traditionally engage with computing. As educators, how might you collaborate across departments to create meaningful, inclusive learning experiences that integrate computing in authentic ways? (HW)

#INSIGHTS

PRIMMDEBUG: A NEW WAY OF TEACHING DEBUGGING

STORY BY Laurie Gale

If you have programmed before, you've almost certainly struggled to resolve an error. If you teach programming, you've probably seen your students struggle in the same way. This is because debugging, the process of finding and fixing errors, is tough. It can take up hours of our time and completely infuriate us.

Yet debugging has so much potential as a skill. Through it, students learn to problem-solve, troubleshoot, and build resilience while learning — all skills important in everyday life.

But how do we teach students how to become good debuggers? My PhD has been focused on this question, particularly for secondary-school students (aged 11–16), and in this article I'll be sharing something I have made to teach debugging.

Debugging can be challenging

Why is it so common to struggle with debugging? Well, successful debugging can require lots of knowledge that beginners are unlikely to have. Debugging a difficult error may require good understanding of program flow, typing, or other mechanics of a programming language. These are difficult to keep in working memory at the same time, especially when learning to program.

Related to this, many beginner programmers use ineffective debugging strategies which often resemble a trial-and-error process. Near-random changes to a broken program are unlikely to have a positive impact, and they might even make things worse. After a while, this can easily become frustrating; most people don't enjoy being completely stuck on an error. This can make debugging an emotional process too.

Although teachers are on hand to help in the classroom, they may be rushing around helping to debug other students' programs (**helloworld.cc/Michaeli**), so it is not always possible for struggling students to depend on the teacher for debugging support.

Teaching debugging

So, how can we help all students to debug successfully? Well, having a toolkit of good strategies is important. There are already several great approaches that can help to develop this toolkit:

- Cheat sheets illustrating correct syntax can help students debug simple errors
- Students can get used to common errors by having their programs 'sabotaged' (**create.withcode.uk/kpride**)
- Teachers can model their debugging behaviour by getting students to 'stump the teacher' (**helloworld.cc/Kerslake**)
- Large language models can be used for lots of debugging-related support, such as decoding complex error messages

As well as these approaches, it is important for students to understand the debugging process as a whole. This is particularly true for more complex errors that require more thinking. In research, this is often referred to as systematic debugging. There's even a textbook on how to do this (**whyprogramsfail.com**), which helped to shape this work.

Introducing PRIMMDebug

PRIMMDebug is a teaching aid for teaching debugging to secondary students (**primmdebug.web.app**). This is based on the PRIMM (Predict–Run–Investigate–Modify–Make) (**helloworld.cc/primm-quick-read**) approach to teaching programming. PRIMMDebug promotes a reflective approach to debugging by making the process explicit. Let's look at how it works.

The PRIMMDebug process is similar to PRIMM, with some extra scaffolding tailored to debugging. We start with a PRIMMDebug challenge, which contains the following information:

- A program with a single error that students are likely to make

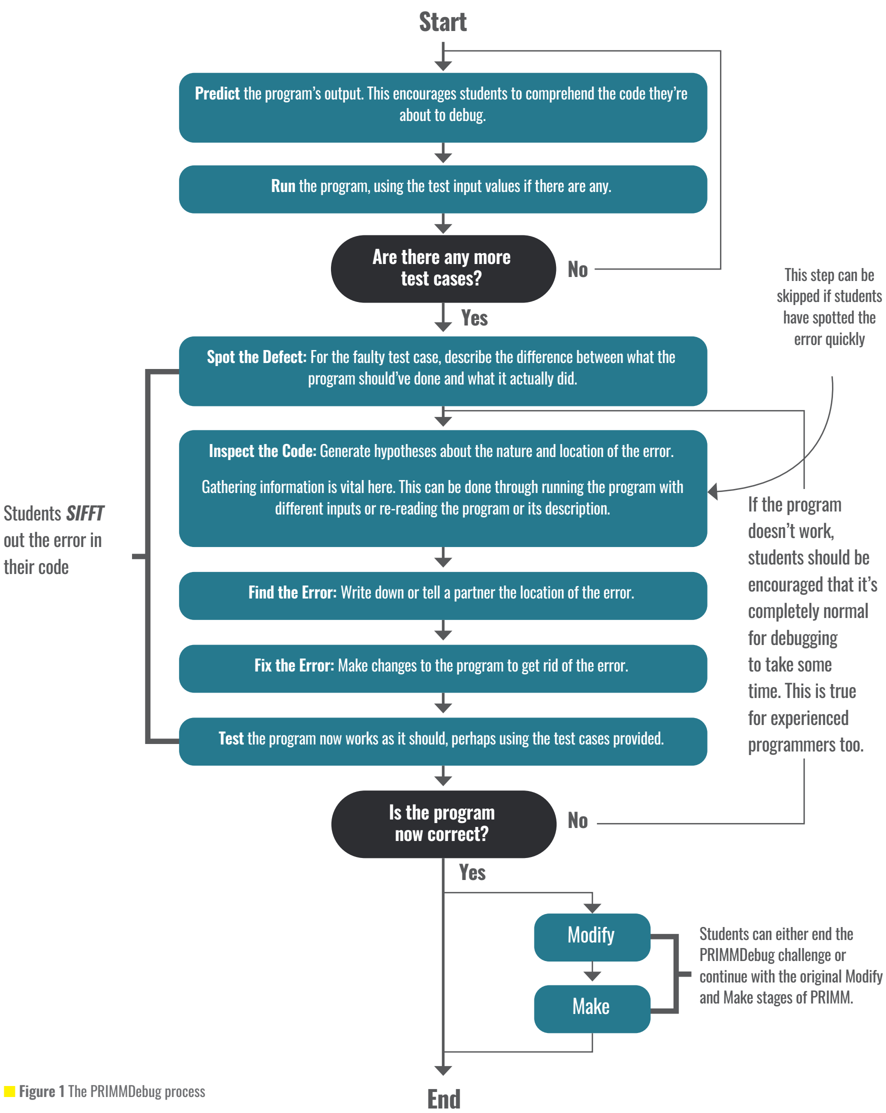


Figure 1 The PRIMMDebug process

- A description of what the program should do
- Optionally, some test cases (sets of input values to test the program with, mapped to the program's corresponding expected output); at least one test case should expose the error

The process is best explained with a diagram (Figure 1). This approach is quite long-winded, so might take some time to complete. Not all the steps are specific to debugging, either, so students are not expected to remember the whole process. However, the middle five steps are generic to the debugging process, which spell out the acronym SIFFT. PRIMMDebug doesn't need to be explicitly communicated, but you can encourage your students to SIFFT the errors out of their programs by following these steps.

The PRIMMDebug tool

The PRIMMDebug process can be modelled in the classroom in several ways. One secondary teacher who has tried out PRIMMDebug used it as a whole-class plenary, without telling their students about the error in the program.

But it is also important for students to have a go themselves. The tool is a simple web app that takes students through the PRIMMDebug process step by step. It's designed to be simple and was informed by feedback from teachers and researchers.

Some of the key features of the tool help to push the behaviours that PRIMMDebug aims to encourage. A good example is the 'articulation pane' on the right-hand side of the tool, where students write down their responses to the questions for each stage. Students can't run the program until they have written down a prediction of the program's output, and they can't edit the program until they have completed all of the stages before 'Fix the Error'.

Other scaffolding is also provided by the tool. The test cases are displayed at certain stages, as well as hints if students don't successfully resolve the error the first time.

The tool is available online at **primmdebug.web.app** and is equipped with a set of ready-made PRIMMDebug challenges. It is still being researched and developed, but do check it out if you are interested.

PRIMMDebug is designed to expose students to the whole debugging process. A big part of this process is staying calm and persevering if things don't go right the first time. Talking about this can help to build a positive error culture in the classroom. Errors are a completely normal part of programming and not something to be ashamed of. Hopefully, this will help students to fix errors more reliably — and maybe even enjoy the process. (HW)

FURTHER READING

Kerslake, C. (2024, March). Stump-the-Teacher: Using Student-generated Examples during Explicit Debugging Instruction. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (pp. 653–658). **helloworld.cc/Kerslake**

Michaeli, T., & Romeike, R. (2019, April). Current status and perspectives of debugging in the K12 classroom: A qualitative study. In *2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1030–1038). IEEE. **helloworld.cc/Michaeli**

Zeller, A. (2009). *Why programs fail: a guide to systematic debugging*. Morgan Kaufmann. **whyprogramsfail.com**

Check out the KPRIDE auto-sabotager tool developed by Pete Dring **create.withcode.uk/kpride**

For more information on PRIMM, check out this Quick Read from the Raspberry Pi Foundation: **helloworld.cc/primm-quick-read**

#INSIGHTS

AUTONOMY: A FRAMEWORK FOR CROSS-CURRICULAR ANALYSIS

STORY BY Jane Waite

Conducting research can be an unpredictable journey; one never knows when an idea will resonate deeply, become a transformative theory, or leave a lasting impact. Over the last ten years, I have been very fortunate to work with Professors Paul Curzon and Karl Maton on a strand of research that I believe has this kind of long-term potential. This work is on bringing Legitimation Code Theory (LCT), a general sociological set of frameworks, to computer science education (helloworld.cc/legitimation-code-theory).

Initially, Paul, Karl and I focused on one LCT dimension, that of semantics, which has already gained traction in computing education (**helloworld.cc/semantic-waves**). For example, Ofsted (a department of government in England which inspects education service providers) has recommended using semantic wave principles in computing pedagogy. Recently, we have shifted to a second dimension: Autonomy. While still in its early stages of research in computer science, Autonomy could prove equally valuable, perhaps even more so for primary education research.

A framework for cross-curricular analysis

In September 2024, I presented with Paul on our first computer science Autonomy research study at the ITiCSE conference in Milan. The research was a case study, exploring and evidencing how to use the Autonomy framework in computing practice. The theory was very well received. Several researchers came to us later asking for more details and talking about how they were going to use the theory to analyse their work, and teacher trainers told us that they would introduce the framework to their trainee teachers.

At its core, Autonomy is a framework that can be used for analysing when one teaches more than one thing at once. For example, in computer science, we might teach about programming and science, or spreadsheets and maths, or digital media making and art (the list is endless). While these interdisciplinary approaches hold promise, they often fall short, with students failing to grasp one or both subjects effectively. Why is that?

Autonomy gives us a way to analyse these situations. It introduces a formal, structured language for talking about cross-curricular teaching, helping educators to pinpoint areas where things are going wrong. In our case study, we analysed the teaching of algorithms through the teaching of magic, and the results revealed important insights about how to optimise such lessons for better learning outcomes.

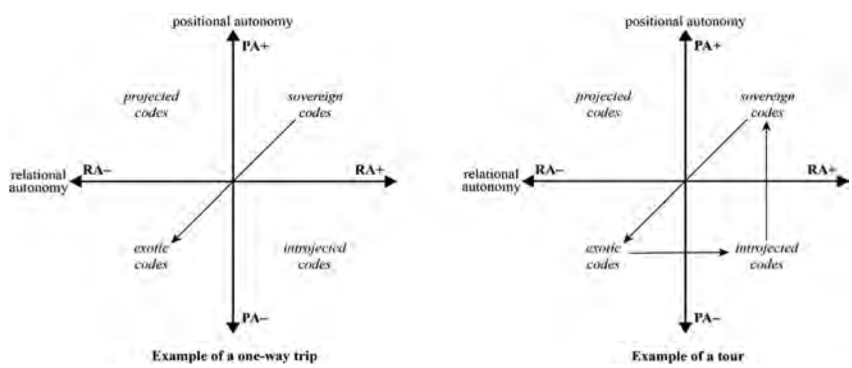


Figure 1 An example of a one-way trip and a tour pathway. (Maton, K. and Howard, S. K. (2018) Taking autonomy tours: A key to integrative knowledge-building, LCT Centre Occasional Paper 1: 1–35.) helloworld.cc/maton

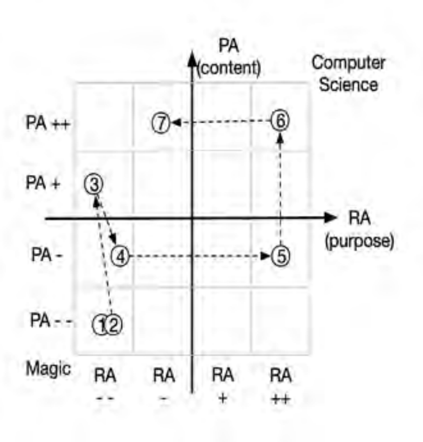


Figure 2 A one-way trip in our analysis of a computer science activity case study (Curzon, Waite, & Maton, 2024)

How does an Autonomy analysis work?

An Autonomy analysis focuses on two key dimensions of a learning event:

1. **Relational autonomy (RA):** the purpose of a teaching step. Why is this step being taught? Is it intended to teach the principal subject or the cross-curricular topic?
2. **Positional autonomy (PA):** the content of the teaching step. What is being taught at this moment?

In Autonomy analysis, a learning event is broken down into steps, and the dimension of autonomy is determined for each step. Simply put, you analyse why you are teaching that step, and what the content of the step is. For our case study, we first determined whether the purpose of each step was to teach magic or to teach about algorithms (or something else). Secondly, we analysed what was taught in that step, and whether the content was focused on the magic trick or computer science content (or something else). Armed with this analysis of the steps, we could then draw a diagram that showed the changes in purpose and content over the course of the lesson. By breaking a lesson into discrete steps and analysing each step through these lenses, educators can visualise the shifts in purpose and content over the course of a lesson (**Figure 1**). The analysis can then be visualised in a diagram that maps these changes or pathways, offering insights into the effectiveness of cross-curricular teaching.

Types of pathway

Autonomy diagrams can show two different types of pathway (**Figure 1**):

1. A one-way trip that takes you from one quadrant on the diagram to another and does not return, potentially leaving key concepts underdeveloped.
2. A tour (like a guided tour on holiday) that starts and ends in the same conceptual place. Research from other disciplines suggests that tours are more likely to result in successful learning experiences.

The four principal autonomy codes are named (from top right clockwise in **Figure 1**) as sovereign codes, introjected codes, exotic codes, and projected codes. In the context of our computer science education research example, 'sovereign codes' were lesson steps where the content and purpose were aligned with the target of computer science and algorithms topic being taught. 'Introjected codes' were those steps that were aligned with the purpose of teaching about computer science and algorithms but the content taught was about other subjects, such as magic here. 'Exotic codes' were steps that were associated with other contexts (not computer science and algorithms) for both content taught and purpose — in our example, mostly for teaching about magic for the purpose of learning about magic. Finally, 'projected codes' were steps in the lesson where the content taught was not about target computer science and algorithms, rather other contexts (such as magic), yet the purpose of the steps was to teach about computer science and algorithms.

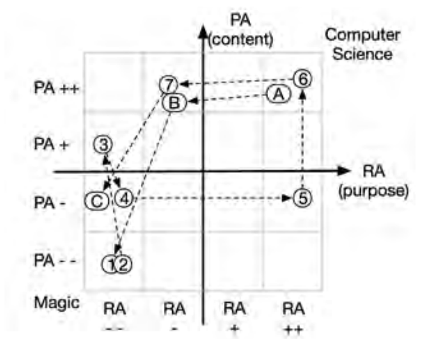


Figure 3 Our improved computer science activity case study, now a tour (Curzon, Waite, & Maton, 2024)

We found that our case study lesson was initially a one-way trip (**Figure 2**). Students started learning about magic for the purpose of learning about magic, and ended learning about algorithms but for the purpose of learning about magic. To improve the lesson, we added extra steps to make it into a tour (**Figure 3**). We changed where the lesson started, with a step clearly about algorithms for the purpose of teaching about algorithms, and returned and overlapped with existing steps, and added an extra step to clearly link back to magic. While this adjustment showed promise in reflections following lesson delivery, further empirical studies are needed to evidence the effects of our changes.

Why does the Autonomy framework matter?

Teaching computing is widely regarded as challenging, especially in cross-curricular contexts. Lessons that aim to teach computer science through and with another topic have often been found to be unsuccessful. Spending time thinking deeply about what is going on in these lessons using Autonomy analysis may help reveal what is going on and put us in a better position to improve them.

Our journey with Autonomy is just beginning, but we believe that it holds incredible potential for improving cross-curricular teaching in computer science and beyond. If you want to find out more about Autonomy, please read our paper (**helloworld.cc/autonomy**), or you can read about the theory on the LCT website (**helloworld.cc/lct-computing**). (HW)

FURTHER READING:

Curzon, P., Waite, J., & Maton, K. (2024, September). Teaching CS with and through other forms of knowledge. In *Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research* (pp. 1-4) (**helloworld.cc/autonomy**).

#INSIGHTS

WHY KIDS STILL NEED TO LEARN TO CODE IN THE AGE OF AI

STORY BY Philip Colligan & Mark Griffiths

What do you say to a young person who is being invited to write `print("Hello World!")` for the first time and is asking what the point of it is?

It's a fair question. The timetable is packed and every hour spent teaching one thing is time not spent on another.

Let's put the challenge as forcibly as we should: why learn to program in a world where, if ChatGPT 4.0 were a student in an introductory computer science course, it would be in the top 6 percent in some exam results (**helloworld.cc/denny**) and where programmers in the workplace are increasingly using tools such as Copilot to generate code?

Are the traditional arguments for learning to code, and even program, still valid? Based on what we know, and the tools we have available now, we at the Raspberry Pi Foundation argue in a recent paper (**helloworld.cc/why-code**) that they are.

More than that, we argue that in a society and an economy increasingly being shaped by artificial intelligence (AI), it is even more vital that all young people have access to high-quality learning opportunities to learn to code and program, and through that, to create, make, and act. Coding in schools remains the best lever we have to spread these opportunities.

We can teach young people how to use technology to fulfil human purpose

Human agency

Throughout our paper is the theme of human agency. One way this is enacted is when our imaginary learner writes a first line of code: in this moment, the learner is experiencing, through interacting with a computer, that computers can be made to do what you want!

The opaqueness of the data-driven models that drive AI technologies doesn't change the fundamental point. Developing the conceptual apparatus to understand how these technologies are built helps shift us from a deterministic view of technology, or even seeing them as akin to magic, to asking questions such as: What data is being used? What assumptions underpin the model? What are the real-world consequences? How can they be used to serve human purposes and create value?

Human agency is also needed if we are to effectively use generative AI (GenAI) tools which, we shouldn't forget, are probabilistic systems designed to generate statistically acceptable outputs. There is no guarantee of accuracy or relevance. These tools are best used to augment human intelligence. In our context, the outputs they propose need to be interrogated critically for safety, quality, their relationship to other parts of code, and most of all, their effectiveness at addressing real-world problems.

Importantly, the expertise to use GenAI is best built up, in part, by the act of coding. For example, it is through writing code that a young person gets first-hand experience of how computers behave and, over time, develops the mental models that allow them to translate a real-world opportunity (such as creating a language-learning app) into components amenable to a computational approach and, eventually, a solution.

Currently, we don't have a better way of building these skills than by directly interacting with the computer through coding. That is one source of our agency over them. For example, a fascinating study of knowledge workers' use of GenAI (**helloworld.cc/lee**) found that critical thinking was used more when the workers had higher confidence in their ability to complete the task without GenAI tools.

Once acquired, this expertise continues to be a powerful way for young people to achieve what is important to them. For some, that will eventually be a high-paid job, or forming their own business. For others, it may be conducting data-driven investigations in the public interest (**helloworld.cc/sds2025**) or making an app that entertains their friends.

Skilful shapers of AI

There is a wider point here: in a time of economic uncertainty, when it is very easy to feel overwhelmed and depleted, a good position for a young person to place themself in is to be on the right side of the macro-trend for powerful computational approaches to be used across society and the economy.

To evidence this, the OECD's international assessment of what 15-year-olds know and can achieve (**helloworld.cc/pisa2025**) recognises the importance of "students' capacity to engage in an iterative process of knowledge building and problem-solving using computational tools".

A recent interim report into England's national curriculum (**helloworld.cc/uk-curriculum-review**) talks of "the opportunities that will be available to those who can become the most skilful shapers and operators of AI" and can, in turn, integrate that with their domain-specific knowledge.

Integrating computational thinking

Schools in Alabama, USA, for example, are integrating computational thinking (**helloworld.cc/talladega**) into their English, history, science, and maths classes, and when they visit the library.

To extend this, in a business studies class, young people could support an imaginary retail company to raise its sales by using data science techniques to analyse customer purchasing data. A history student could use natural language processing libraries to analyse how English writers at the time described the events of the French Revolution, which they could then integrate into their wider knowledge. A politics student may analyse Reddit forums to identify the adjectives most commonly used to describe a politician. They could go on to integrate these with their understanding of the role of conceptual metaphors in politics.

The range of action is enormous, because the range of motivations and interests young people have is enormous. This is mass agency through problem-solving, discovery, and creation. We are not fighting against technology, but using it with understanding and critical awareness to fulfil our human purposes.

So, to the young person asking, "Why do I have to learn this?" our response, in a slogan, is that programming gives you the power to create, innovate, discover and shape the world. And that's something worth learning.

Supporting educators

Of course, this all needs to be enabled through the agency of the teachers and volunteers that work with young people — and of the school systems that support them. For too long, we've asked teachers to teach a subject that they didn't study themselves. We almost always fail to give them the time and investment in their professional development that they need to succeed.

School systems themselves have been battered by the effects of the pandemic; they are wrestling with persistent and unacceptable learning poverty; and, in many parts of the world, simple access to devices and the internet remains a challenge.

However, we shouldn't forget that during the Second World War, 25 percent of recruits — around 700,000 men — were rejected by the US army because they were unable to meet basic literacy qualifications. The response to this is salutary: well-designed course materials, skilled teaching, and sufficient funding (**helloworld.cc/soldiers-literacy**).

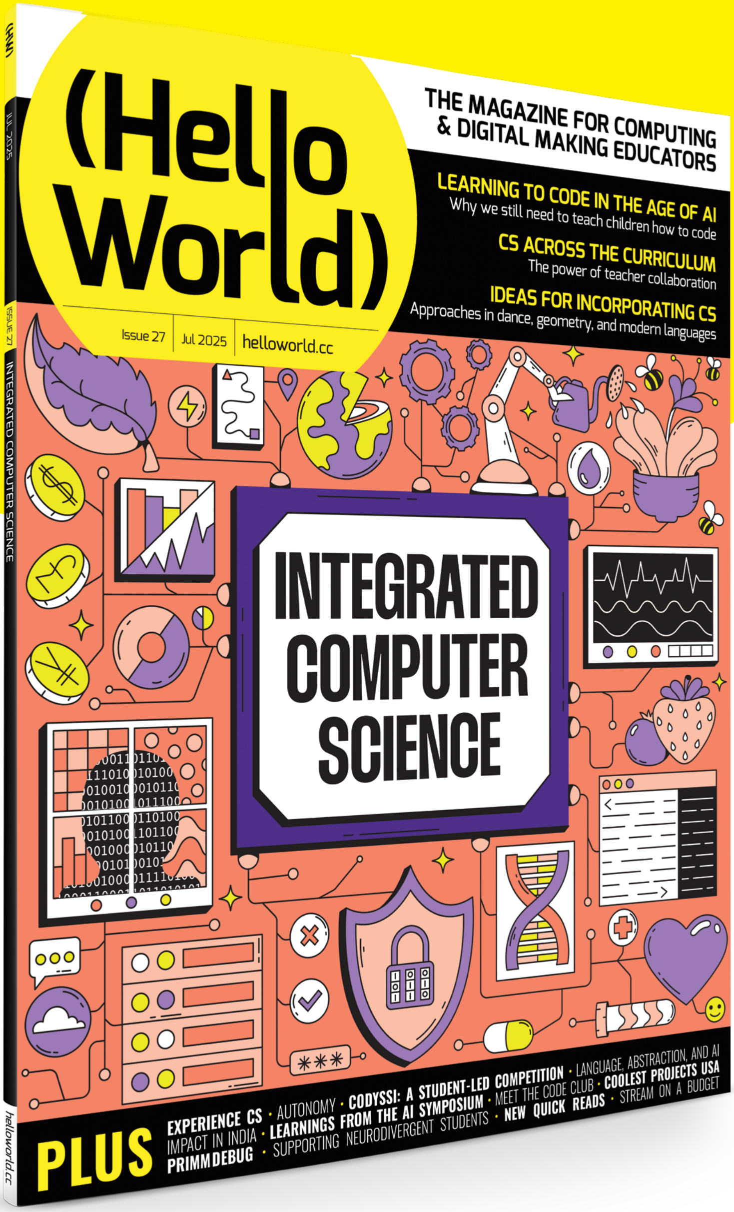
Being able to code and program is — like reading, writing, and maths — a literacy. It helps us participate fully, and as informed citizens, in the world. It provides agency.

As the opportunity space of using digital technologies expands and deepens, now is the time to provide all parts of the learning system, and learning systems around the world, with what they need to enable all young people to thrive in an age of powerful and ubiquitous digital technologies. For now, coding remains essential to that. (HW)

HEAR MORE FROM PHILIP ON OUR PODCAST, WATCH THE EPISODE HERE
helloworld.cc/pod27-code

(HW)

SUBSCRIBE TODAY



FREE IN PRINT for UK-based educators

Why subscribe?

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


FEATURE
INTEGRATING CS ACROSS THE CURRICULUM
WELCOME TO THE CSTA 2025 ANNUAL CONFERENCE!
IF YOU'RE A TEACHER WHO IS IMPLEMENTING CS PRINCIPLES INTO YOUR CLASSROOM, YOU'RE A COMPUTER SCIENCE TEACHER


(Hello World)
THE MAGAZINE FOR COMPUTING & DIGITAL MAKING EDUCATORS
LEARNING TO CODE IN THE AGE OF AI
Why we still need to teach children how to code
CS ACROSS THE CURRICULUM
The power of teacher collaboration
IDEAS FOR INCORPORATING CS
Approaches in dance, geometry, and modern languages
Issue 27
Jul 2025
helloworld.cc
INTEGRATED COMPUTER SCIENCE
PLUS

INTEGRATING CS ACROSS THE CURRICULUM

Jake Baskin looks at the power of teacher collaboration and the community of CSTA, the world's leading association for K–12 computer science teachers

As a former computer science (CS) teacher, I know how important access to teacher-created content is. I still remember so clearly what it meant to be a new high-school CS teacher over a decade ago. As I've shared in many other places, one of the most influential things for me was being introduced to the Exploring Computer Science curriculum and professional development that shaped how I grew as a teacher and leader (exploringcs.org).

When I think about my experience, a specific case stands out. I taught two sections that included students from our school's severe and profound autism programme. Working with the expert special education teachers in our school, I quickly realised we needed to adapt our curriculum to meet the specific needs of these students. Every teacher has unique individuals in their class and brings incredible expertise and wisdom about how to support students where they are at that moment. No single curriculum can address all these needs — that's why creating tools and sharing resources across a community of teachers is so vital to our work. Resources developed by teachers that can be modified by other teachers are essential for meeting students where they are.

Integrating computer science

Every student needs to be prepared for a world powered by computing, and that work starts as early as kindergarten. CSTA's work on the Reimagining CS Pathways report (**reimaginingcs.org**) found that combining CS with other subjects has benefits that can reach more students and build on other CS courses — including helping students better understand the relevance and usefulness of CS.

When thinking about specific skills that integrate well into elementary education, it's important to recognise how the foundations of computational thinking are already present in how elementary students engage with the world. One early example often cited is the concept of sequencing — learning how to follow step-by-step processes, whether that's putting on shoes or getting ready for recess. These experiences connect directly to the decisions we make in computing and understanding how to follow a set process to achieve a task.

AI presents unique opportunities for integration across the curriculum. A key example is helping students understand how both humans and technology sense the world around them. Even our youngest students can begin to understand their own senses, while being introduced to the idea that technology also 'senses' in different ways, and how that information feeds into the technological systems they interact with on a daily basis.

> IF YOU'RE A TEACHER WHO IS IMPLEMENTING CS PRINCIPLES INTO YOUR CLASSROOM, YOU'RE A COMPUTER SCIENCE TEACHER

WELCOME TO THE CSTA 2025 ANNUAL CONFERENCE!

As an organisation for teachers by teachers, I'm thrilled the Computer Science Teachers Association (CSTA, **csteachers.org**) is partnering with Hello World, a magazine for CS educators by CS educators. Hello to our CSTA members who are reading this issue at the CSTA 2025 Annual Conference in Cleveland, USA, and hello to those of you who are reading from around the world.

It's been incredibly valuable to get the research-backed resources created by the Raspberry Pi Foundation in front of CSTA members, so that teachers can bring innovations from research into their classrooms. I'm particularly proud that a physical copy of the magazine is available to every teacher at the conference, because it makes tangible the immense work and power of teachers and researchers working together to improve what's happening in the classroom. I'm also excited to see how we may collaborate in the future, to continue having an impact on students around the world. Enjoy the conference!

Come join the world's largest computer science department at CSTA

The Reimagining CS Pathways report digs into fascinating ways in which we can align and integrate concepts across curricular areas. As someone who originally wanted to be a creative writing major before entering computing, I love thinking about integration with English language arts. There are interesting ways to analyse text with computers and to compare that with other analytical approaches, examining where computational analysis might provide insights, or where it might fall short.

There's also clear overlap with the fine arts, for example using technology to create new instruments for music, unlocking new ways of creating visual art, or exploring ethical and societal challenges around what it means to create art with technology and where such approaches are appropriate.

So many ways to collaborate

If you're a teacher who is implementing CS principles into your K–8 classroom (ages 5–14), you are a computer science teacher, and CSTA is here to support you. From on-demand and live professional development opportunities, to the searchable 'Resources Library', to the CSTA Virtual Community where you can connect with other teachers in real time, CSTA has the resources to support you with integrating CS into your classroom. Head to **csteachers.org** to learn more about our community of teachers.

If you're thinking about integration, especially in the elementary space, I encourage you to connect with your local CSTA chapter and our virtual affinity groups where educators come together to discuss what's working in elementary grades. As you review the most recent drafts of the updated CSTA standards, you'll find numerous ideas and opportunities to bring computer science standards and content into your elementary classroom in new and different ways.

While it may feel like you're the only one doing this work in your school or district, you are not alone. Come join the world's largest computer science department! I hope to see you soon as a part of our community. (HW)

JAKE BASKIN

Jake is the executive director of the Computer Science Teachers Association, the world's leading association for K–12 (ages 5–18) computer science teachers. He is a former high-school computer science teacher, department chair, and professional development provider with Chicago Public Schools (**@csteachersorg**).

PRIMARY CODING ACROSS THE CURRICULUM

Code in any lesson to build your students' problem-solving skills, using the concepts of sequence, selection, and repetition

Too often, we talk about coding and problem-solving as isolated skills, but in fact, when these skills are applied, they can help adults and children to solve almost any problem. From simple matters such as working out what to put in your bag to politically complex issues such as keeping peace in the playground, coding can often provide a refreshing solution.

You may not think that children's programming tools could have such a powerful impact, but in a previous school when I was teaching Year 3 (ages 7–8), we had an issue with students arguing and fighting in the playground, and naturally, when you have a problem, you turn to the computer science department!

We quickly realised that students did not actually know the rules of many common games. Ask students around the world for the names of the games they play and you will get lots of answers and hundreds of derivatives. What is essentially the same game can be called everything from 'tag', 'tig', 'tigger' or 'squid chase' to 'manhunt'.

We asked the students to decompose every game down to its core rules. They then created simple flowcharts which were put in the playground to stop these arguments! Similarly, teachers use choose your own adventure stories to help neurodivergent students simplify the complex world around them (**helloworld.cc/social-stories-guide**).

The importance of coding has never been greater, both as a tool to use and as a tool to help us understand our digital world. Coding has a potential use in every subject, and thanks to Scratch and OctoStudio, children can now code on virtually any platform, including iPads and Android devices. No accounts or internet-access is required, and OctoStudio does not collect any data for any purpose, making it a safe space for even very young students to create their own games.

If you want to reduce stress, for example, you can make a 4-7-8 breathing-technique game to encourage students to take a breath (**helloworld.cc/4-7-8-video-tutorial**), or you could make a convertor game to help with science, or a times table quiz. These were all once the domain of professional game creators, but can now be made by seven-year-olds.

Sequence, selection, and repetition

With the three simple concepts of sequence, selection, and repetition, you can make code for a wide range of subjects (**Table 1**). With each concept, you will encourage thinking skills and creativity, and also deepen the learning of the chosen topic.

For example, linguistics and grammar in English lessons are prime candidates for coding. Let us look at conditional sentences, for example: if + [present action], then + [future action].

SUBJECT	SEQUENCE	SELECTION	REPETITION
Art	Creating animations or patterns	Colour mixer	Pattern generator (Islamic art)
English	Storytelling or grammar exercises	Spelling bee helper, teaching conditional sentences	Rhyming word generator, sentence builder
Geography	Erosion, volcano formation, urbanisation	Weather report	Climate data analyser
History	Simulating historical events	Era classification	Comparing cultural calendars
Mathematics	Algorithmic thinking	'Odd or even' game	Multiplication table generator, counting game
Music	Composing music	Instrument chooser	Beat maker
Physical education	Sports or dance routines	Exercise selector	Workout routine
PSHE (personal, social, health, and economic education)	Steps to organise yourself in the morning	Friendship advice	Mindfulness breathing
Science	Photosynthesis simulation	States of matter	Food chain simulation, chemical reaction timer, water cycle

Table 1 Coding ideas for different school subjects

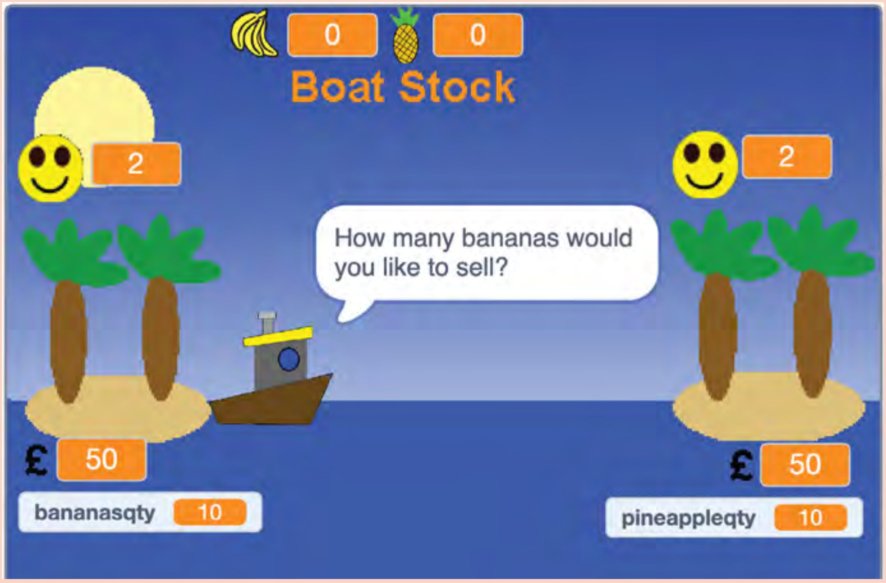
HANDING IN WORK

OctoStudio uses the 'share' function to let students send their files. Scratch relies on students signing in and remembering a username and password. As a teacher, you can get a teacher account. This means you can change student passwords as needed, so they will never lose their account. You need to apply for this account, and it can take a few days to come through (**helloworld.cc/scratch-educator-accounts**).

There are various ways to teach conditional sentences, but imagine the excitement in the room when they see: 'If I tap the frog, then it will jump 50 pixels.'

The children know that they will be working with conditional sentences, and because of the coding, the target language is likely to come out quite naturally. In effect you are teaching the computing principle of selection and the language of conditionals. With this simple change we have created a multisensory experience which might even include videos that can be shared with parents.

The tools that come with OctoStudio enable students to create animations, use speech-to-text and text-to-speech, record their voices, and even create videos. This enables students to express themselves in many ways. A nice introduction to characters and storytelling is to make knock-knock jokes. Here is a video tutorial (**helloworld.cc/knock-knock-video-tutorial**) which is also available as a lesson plan for Scratch (**helloworld.cc/jokes-scratch-lesson**).



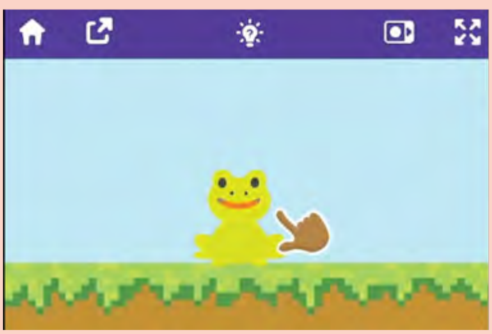
Students can learn complex ideas in a simple way with OctoStudio and Scratch; this animation shows how the balance of payments works

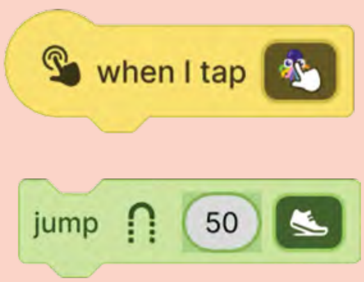
Maths and science

For maths and science lessons, it is great fun to create code to make animations of solar systems and planets that spin. The maths available in OctoStudio is much simpler than that available in Scratch. Scratch covers some fairly advanced mathematics (**helloworld.cc/math-functions-scratch**), but these limitations can be a part of the challenge, and solving problems using fundamentals is enjoyable. It encourages real decomposition! See this tutorial of the moon orbiting the earth in OctoStudio: **helloworld.cc/moon-orbit-video-tutorial**.

You can also think about ways to encourage problem-solving, such as seven-block challenges. With this kind of challenge, you could pick seven specific blocks, or just ask, 'What is the most exciting code you can make with just seven blocks?' These simple problems encourage students to think about scarcity. This is critical to understanding economics, and these tools can help students understand important issues such as the balance of payments. See this Scratch game: **helloworld.cc/balance-of-payments**.

Understanding the building blocks of computing and problem-solving is a timeless skill our students need if they are to make valued contributions to society. (HW)





OctoStudio code to make a frog jump; this animation could help to teach conditional sentences

JAMES ABELA

James is director of digital learning and entrepreneurship at Garden International School in Kuala Lumpur, Malaysia. He is founder of the South East Asian Computer Science Teachers Association and **ReadySetCompute.com**.

ELEMENTARY-SCHOOL CODING AND ROBOTICS

Learn how one school has incorporated coding and robotics into general-education classrooms

If you're an elementary school teacher, you've probably said it before: "There's just not enough time in the day!" Between maths lessons, recess duties, school initiatives, and the occasional assembly, it often feels as if there's no room for the things you're already doing, let alone adding something new.

Yet coding isn't content that schools should put on the back burner. Learning to code improves algorithmic thinking and problem-solving skills. In addition, it develops skills such as collaboration and working through failure. Coding and robotics are also amazing outlets for creativity. Students stretch their imaginations to make their visions come to life, and often have to find original solutions to problems that emerge throughout their projects. Unfortunately, according to Code.org's State of Computer Science Education Survey (**helloworld.cc/state-of-cs**), less than half of elementary schools in the USA have formal computer science opportunities for students.

We can expand access without radically changing elementary-school schedules. At Lake Forest Country Day School, where I teach, we have made a concerted effort to weave computer science into the school day through cross-curricular projects. By adjusting coding to fit the content students are already learning, we can greatly expand access to computer science instruction.

Adding coding to projects leads to additional class time for teaching algorithmic thinking and block coding. Still, one of the most exciting parts of routine coding projects is the cascading effect of regular exposure to the content. The more students work on projects that involve coding, the less class time needs to be dedicated to teaching the tools, and students with previous experience can frequently elevate projects using their prior knowledge.

Access to robots and coding tools is another perceived barrier for educators. Teachers interested in implementing computer science in their classrooms should look for robots already in their building or district that they can borrow for the duration of a unit. There are also grant-giving organisations that can help. In Illinois, IDEA (the Illinois Digital Educators Alliance) offers mini grants for technology goals (**helloworld.cc/mini-grants**); and BirdBrain Technologies also has a robot loan programme for schools in the USA (**helloworld.cc/loan-programme**). Teachers can get Raspberry Pi Picos for as little as $8 per device. At my school, we frequently have students work in groups for coding projects, use robotics as a learning station or as one of a few summative project options, or use built-in simulators in coding platforms to reduce the need for full-class sets of robots.

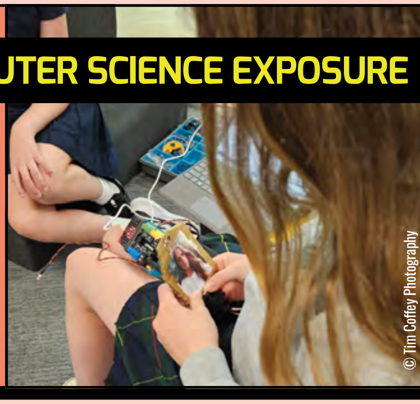
Below are a few of the project ideas we have used with great success, and which other schools could easily implement with a variety of different robotics tools.

Build a map

Creating uniform grids for robots to traverse is a great opportunity to get students aged as young as four into coding and has nearly limitless applications. In Mrs Dohnke's first-grade class (ages 6–7), students learnt about South America as part of their social studies unit. Each student chose a country, read and wrote about it in English language arts (ELA), and created a 15x15 centimetre square with their chosen country's name and

THE IMPORTANCE OF COMPUTER SCIENCE EXPOSURE

Despite 60 percent of high schools in the United States offering computer science courses, only 6.4 percent of students take a foundational class in high school. Of that number, only 32.5 percent are girls. By creating opportunities for exposure in K–8 schools (ages 5–14) we can build a passion for computer science that will encourage all students to participate.



© Tim Coffey Photography

Cardboard and Hummingbird Bits allow students to explore content in a new modality

its important aspects. We then combined all their squares into a class grid.

Students then used Cubetto, a story-based robot, to move across the map. When Cubetto arrived at their square, they explained their country to their classmates. We extended the learning (and coding access) by having the first-graders bring Cubetto to junior kindergarten classrooms and having the junior kindergarteners code Cubetto across the map, with help from the first-graders.

In Mr Hedund's fourth-grade class (ages 9–10), students used cardboard and Hummingbird Bits to build and model a castle. The learning was jigsawed; each student learnt about a different element of castle living. Students designed, built, and coded a character who would live in their portion of the castle. In Ms Vivirito's second-grade class (ages 7–8), students learnt about animals by creating a physical model using a micro:bit and craft materials.

> INCORPORATING CS IN GENERAL-EDUCATION CLASSES CAN HAVE A MULTIPLYING EFFECT ON OUR STUDENTS

This project could work across multiple curricula and with multiple robots. Any bot that has a consistent rotation could be modified to work across a map (including Bee-Bot, Code-a-Pillar, and Code & Go Robot Mouse). We've done similar grid projects to demonstrate events across time, and to help students show their knowledge of the Mood Meter, our school's social and emotional learning tool.

Creating a setting, scene, or character

Text and history analysis is huge in any ELA or social studies classroom, and coding can be a fantastic and adaptable way to demonstrate that learning.

These projects allowed students to explore content they had learnt in an entirely new modality. Adding coding concepts to assessments encourages deeper thinking about course material, as students make more deliberate choices as they code.

Science and non-fiction extensions

Robotics is one of the best avenues for taking conceptual ideas and making them real. In Mrs Rabjohn's third-grade classroom (ages 8–9), students create several projects at the culmination of non-fiction reading and writing units throughout the school year. Students used servomotors to demonstrate different kinds of joints during their human body unit by making functional models in small groups. During a unit about light and sound, students used sensors to measure the sound and light in the room.

Taking concepts students learn in class and aligning them with coding projects takes abstract concepts and makes them concrete. Importantly, it also encourages students to play as their understanding is assessed. Explaining how sound waves travel on a paper and pencil test isn't nearly as impactful as creating a cardboard boombox that lights up when music is played near to it. These projects could be completed with any microcontroller with inputs and outputs. There are possibilities for interdisciplinary units involving energy, motion, moisture, and more that could be accessible to students in second grade and above.

One of our third-grade students had so much fun with our human body unit that she went home and got a micro:bit for her and her younger sister to continue exploring coding and robotics! Moments like this show the impact of coding in the classroom and how incorporating computer science in general-education classes can have a multiplying effect on our students. (HW)

GREG MCDONOUGH

Greg is the innovation space director at Lake Forest Country Day School, Illinois, USA. He teaches maker education to students from preschool through eighth grade (**@mcdonough_greg**).

FROM SILO TO SEAMLESS: MAKING COMPUTER SCIENCE A NATURAL FIT IN EVERY CLASSROOM

Computer science helps students connect the dots in every classroom experience

For too long, computer science has been siloed as a stand-alone subject, often reserved for high-school electives, after-school clubs, or specialised programmes. This approach has unintentionally reinforced the idea that CS is only for a select few: the tech-savvy, the maths-minded, or those planning to major in STEM. But this couldn't be further from the truth.

When CS is integrated into the core curriculum, it becomes more than just coding. It transforms into a dynamic, interdisciplinary tool for building essential skills like problem-solving, collaboration, and creativity. It becomes a powerful equaliser in a world where access to opportunity is too often unequal.

CS isn't just for coders: it's for everyone

Computer science fosters transferable skills — like critical thinking, creativity, and persistence — that are essential in nearly every profession, from healthcare to the arts to skilled trades. Whether students go on to become software engineers, entrepreneurs, or community leaders, CS helps them develop the adaptable mindset and digital fluency the modern workforce demands. But perhaps even more importantly, it helps students see the world differently.

Whether you're teaching fractions, fiction, or social justice, CS can help students connect the dots, ask better questions, and imagine bold solutions. Algorithms can illustrate patterns in poetry. Data can amplify student voices in discussions about equity. Programming can model real-world maths problems or simulate ecosystems. When teachers intentionally weave computer science into their subjects, it becomes a lens for understanding and reshaping the world.

A story from the classroom: CS in a New Mexico middle-school science lesson on water

In a middle-school Earth science class in New Mexico, Mr Ortiz kicked off a unit on the water cycle and regional precipitation patterns. But instead of handing students a diagram and a textbook to read, he gave them a challenge: what can we learn about our local climate by analysing data about rainfall over time?

Using spreadsheet software and beginner-friendly coding tools like Python, students accessed open-source precipitation data from the USA's National Weather Service. They wrote simple scripts to clean, sort, and visualise the data — comparing rainfall levels across decades, tracking seasonal trends, and

Educators don't have to be tech experts to bring CS into their classrooms

Computer science helps students see things differently

even identifying signs of drought years.

As students worked in groups to code their own graphs and interpret the results, the conversation shifted. They weren't just completing a lab — they were becoming data scientists in the environment they reside in. One group discovered that there had been a sharp decline in rainfall over the last 15 years, and began asking deeper questions about water access, agriculture, and equity in their own communities. Another group connected the data to current events, such as restrictions on water usage during wildfire season.

Mr Ortiz didn't have a computer science degree. He had attended CS professional development and simply integrated free, accessible tools and framed CS as a way to deepen scientific inquiry. His students walked away with a stronger understanding of precipitation and a new confidence in their ability to use data and code to explore complex, real-world issues.

Integration over isolation

Stories like this one remind us that integrating CS doesn't mean reinventing the wheel. It means looking at what's already happening in classrooms and asking, 'Where could computer science help students go deeper? How can we use these tools to bring learning to life?'

That's where the Computer Science Teachers Association (CSTA) comes in (**csteachers.org**). CSTA is a home for all educators — not just computer science teachers — who want to bring CS into their classrooms. With diverse resources, professional development, and a welcoming educator network, CSTA supports teachers at every stage of their journey, no matter their subject area or experience level.

We normalise CS integration by making it feel doable. That means supporting teachers with easy-to-use tools, cross-curricular lesson ideas, and — most importantly — community.

A call to action

You don't have to be a tech expert to get started. You just have to be curious and open to exploring how CS can enrich your practice and empower your students.

Computer science belongs to everyone. It's time to bring it out of the silo and into the spotlight — not as a subject to be feared or reserved for the few, but as a vital, vibrant part of learning for all. (HW)

JUSTINE CHAVEZ-CRESPIN

Justine is the professional learning and content manager at the Computer Science Teachers Association. Justine remains focused on bridging the gap in CS education. She empowers underrepresented students, and aims to create a supportive environment where everyone feels valued.

REIMAGINING COMPUTER SCIENCE REQUIRES INTEGRATION

How to integrate foundational and advanced computer science content at the high-school level

Our world is powered by computing, and recent advances in AI are only accelerating our collective reliance on it. The need for early, universal computer science (CS) education is becoming ever more important, as all students need quality CS education if they are to become informed citizens and confident creators.

Yet most students still do not learn CS. According to the 2024 State of CS Report (**helloworld.cc/state-of-cs-2024**), 60 percent of public high schools in the US offer at least one CS course, but just 6.4 percent of high-school students took one last year (if all students took one CS course during high school, we would expect to see enrolment of around 25 percent). There are also persistent disparities in who takes CS courses.

I have always been a proponent of stand-alone CS instruction to promote quality instruction and equitable implementation. To reach all students, though, it is necessary to integrate computer science into other subject areas. Adding another subject to an already overflowing curriculum is daunting, and the number of existing instructional requirements gives little wiggle room, especially for students who require more support (for example, multilingual learners who also take an English language development course, or students who need to retake a course). Many schools cannot add CS courses without displacing other important courses in a wicked zero-sum game. This is especially true at smaller schools, where there is typically a narrower course menu; for them, integration may be the only option.

When done thoughtfully, integrated instruction can be meaningful. De-siloed instruction enables relevance, and application that is more similar to what we want students to be able to do in the real world. Ideally, schools would not choose one approach, but offer both. This enables students to develop a strong foundation akin to that in other disciplines, while also allowing them to apply and extend this foundation in relevant and engaging ways.

Integration in high school

While much existing work on integrating CS focuses on elementary grades, it is just as important in high school. This can take the form of an integrated course such as Bootstrap:Algebra (**helloworld.cc/bootstrap-algebra**), which integrates CS into algebra courses, or SFUSD's Creative Computing course (**helloworld.cc/creative-computing**), which integrates CS into an introductory digital arts course. Reimagining CS Pathways defined several compelling X+CS pathways too (**helloworld.cc/xcs-pathways**), for example, computational journalism.

Foundational CS can also be meaningfully integrated into existing coursework in smaller ways, such as within a single lesson or unit. **Figure 1** shares some examples of how CS can enhance instruction in other subject areas.

Beyond the foundation

When all students develop a strong foundation in CS, teachers can leverage and extend their CS knowledge and skills beyond the foundation level and into other non-CS courses. Integration is thus helpful not only in the teaching of the foundation, but also in the teaching of advanced content. Here are some examples illustrating how more advanced AI content can be integrated into other courses, after students have learnt foundational CS:

Science: students use sensors to gather data about a chemical process and analyse it using an AI library.

- **Mississippi Career-Readiness Standards for Science (helloworld.cc/**

REIMAGINING CS

Over the past two years, the Computer Science Teacher's Association (CSTA), the Institute for Advancing Computing Education (IACE), and our partners have sought to reimagine CS education. The Reimagining CS Pathways (**reimaginingcs.org**) project resulted in a community definition of foundational computer science content. It differs in a few notable ways from current curricula, including a greater focus on social impacts and ethics, algorithms and computational thinking, data and AI, emerging technologies, and the need to connect CS learning to careers. This article highlights integration guidance from the Reimagining CS Pathways report.

SUBJECT	SUBJECT CONCEPTS	CS CONCEPTS	SOURCE
English language arts: using a text file of *Romeo and Juliet*, students record counts for each character's dialogue and then visualise that data. Using the visualisation, students look for patterns in the data and then use the patterns to confirm what is known about the play and generate new questions about the text. Students also assess word frequency per scene to look for patterns in the text.	Close reading for meaning and tone	Types of data, data cleaning, data analysis, and visualisation	Integrated Computational Thinking (**helloworld.cc/integrated-CT**)
Mathematics: students create flags using a combination of maths and CS concepts. First, students sketch the image on graph paper, then they experiment with predefined functions to decompose elements of national flags and compose additional flags.	Ratios, coordinates, scaling	Functions, decomposition, image manipulation, comments	Bootstrap:Algebra (**helloworld.cc/making-flags**)
Science: students develop and experiment with computational models to explore the behaviour of a forest fire and its impact on the forest ecosystem.	Ecosystems, evolution, patterns and systems, using models	Clearing, analysing, and visualising data; troubleshooting and debugging	CT-STEM (**helloworld.cc/ct-stem**)
Social studies: students explore patterns in population change across countries and time spans. They create multiple data visualisations by using a specialised tool to adjust parameters to generate the appropriate visualisation, which can then be analysed.	Population growth patterns, data literacy	Function parameters, data visualisation	Data Visualization for Learning (**helloworld.cc/teaspoon-programming**)
Fine arts: students create a song by using predefined functions with the appropriate parameters, as they practise using music concepts and terminology.	Elements of a song (tempo, measures, sections)	Functions, parameters	EarSketch (**helloworld.cc/earsketch-tutorials**)

■ **Figure 1** How CS can enhance instruction in other subjects

mississippi-science): 'Students will use mathematical and computational analysis to evaluate problems.'

■ **AI concepts:** sensors, perception, and classification; using data: collection, cleaning, data types, validity, bias.

Social studies: students develop a plan of action related to the environmental costs of developing LLMs.

■ **New York Learning Standards for Social Studies (helloworld.cc/ny-social-studies):** 'Prepare a plan of action that defines an issue or problem, suggests alternative solutions or courses of action, evaluates the consequences for each alternative solution or course of action, prioritises the solutions based on established criteria, and proposes an action plan to address the issue or to resolve the problem.'

■ **AI content:** using AI tools to solve problems; ethical frameworks, philosophy, psychology, bias.

Mathematics: students analyse the output of unsupervised learning models that categorise data.

■ **Texas Mathematics Essential Knowledge and Skills (helloworld.cc/texas-math):** 'Students will extend their knowledge of data analysis and numeric and algebraic methods.'

■ **AI content:** using data sets, regression, probabilistic thinking; representation and reasoning, KNN, vectors.

Fine arts: students explore similarities and differences between how AI models and artists make use of the intellectual property of others, as well as the ethical and legal ramifications of such use.

■ **Nevada Visual Arts Standards (helloworld.cc/nevada-visual-arts):** 'Demonstrate awareness of ethical implications of making and distributing creative work.'

■ **AI content:** biases in data collection, analysis, and reporting; AI programming.

English language arts: students write prompts using techniques such as few-shot prompting.

■ **Illinois English Language Arts Learning Standards (helloworld.cc/illinois-english-language-arts):** 'Produce clear and coherent writing in which the development, organisation, and style are appropriate to task, purpose, and audience.'

■ **AI content:** natural interaction, semantics, chatbots; prompt engineering.

When all students are prepared for a world powered by computing, they will be empowered to utilise foundational CS knowledge and skills in their personal lives, communities, and careers. The more we can teach the meaningful integration of CS, the better students will be prepared to continue to do this on their own. (HW)

This work is supported by the National Science Foundation (NSF) under Grant No. 2311746. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

BRYAN TWAREK

Bryan ('BT') is head of research and innovation for the Computer Science Teachers Association (CSTA). He led the Reimagining CS Pathways project alongside Dr Monica McGill, Jake Koressel, and Dr Julie Smith (**@BTwarek**).

MOVE THROUGH THE CURRICULUM WITH DANCE LIVE!

Building real-world skills across the arts, media, and digital technology

Dance Live! is a high-energy, performance-based competition in which our students can shine on a professional stage, while making real-life connections and cross-curricular links across all subjects.

Our school is proud to be taking part in Dance Live! (**dancelive.co.uk**) for the third year running. It's about more than just performance. Behind the scenes, students take on key production roles including choreography, filming, editing, stage management, sound engineering, and event coordination — gaining hands-on experience in every aspect of live-event production.

A role for everyone

Primary and secondary students combine dance, digital media, and storytelling. Each school creates a seven-minute dance routine enhanced by student-designed digital backdrops, lighting, and special effects. Performances can be based on any theme — historical, fictional, or real-world — allowing students to creatively express their learning. No prior technical experience is needed, as training is provided on the day of the event.

Entry requirements

- Open to primary and secondary schools in the UK
- A minimum of 20 students in the performance
- A group of student digital leaders to support lighting and videography
- Teams must create a seven-minute performance with a digital backdrop
- Students must be involved in both performance and production
- Schools must submit a two-minute journey video documenting their creative process
- Schools must attend a regional heat to perform live in front of an audience and judges

In our first year, our performance was inspired by the Suffragettes — a powerful historical story about women's rights. The students had been exploring this topic in their core subjects, making it a meaningful and relevant theme for their dance. In our second year, our performance was inspired by the novel *No Ballet Shoes in Syria* by Catherine Bruton. The students had been studying the novel in English, focusing on the emotional journey of a young refugee and the challenges faced in search of safety and belonging.


Dance Live!

Skill	Description
Literacy	Writing scripts for performance
Digital	Designing LED backdrops with apps
Dance	Choreographing routines for storytelling
Drama	Expressing emotions through movement
Art	Creating stage props for performance
Music	Mixing soundtracks for the performance
History	Basing stories on historical topics

Figure 1 The wide range of subjects connected to Dance Live!

Cross-curricular links

From our very first experience with Dance Live!, we quickly saw how the project sparked meaningful cross-curricular connections. What started as a collaboration between dance and digital media soon evolved into a whole-school initiative, with staff from various subjects joining in to support the students' final performance. As the project progressed, it became clear just how many areas of the curriculum were involved. **Figure 1** highlights the wide range of subjects that were meaningfully linked through this creative journey.

Real-life experiences through Dance Live!

Creating a student dance performance may not sound like a project that involves a lot of technology, but especially for some girls, who may have a tendency to view dance and creativity as very separate to computing, Dance Live! has been an excellent way for our students to gain real-life technical experience. Planning lighting and camera work, and editing music and video, all demonstrate how creative professionals use their digital skills in real-world careers.

Dance, digital skills, and more are all a part of creating a performance for the Dance Live! competition

Our students also gained valuable life skills working on the competition, as described in the competition stages below.

Rehearsals and training

Preparing for the competition involved rehearsals in which students refined their technique, learnt choreography, and built performance stamina. The girls showed outstanding commitment, energy, and teamwork — demonstrating resilience and a strong work ethic throughout, highlighting our school values.

Performing for live audiences

Performing live helped students build confidence, develop stage presence, and learn how to engage an audience — valuable skills for a wide range of future careers.

Camera work

Filming the performance was an exciting technical challenge that pushed our digital leaders to think creatively and strategically. They carefully analysed the routine to plan camera angles and capture key moments — sharpening their problem-solving and technical skills. Despite having limited prior experience, they rose to the occasion with support from the team on the day, turning it into a valuable learning opportunity.

Lighting design

Lighting brought the performance to life. Our students explored both artistic and digital techniques to design lighting states that enhanced each scene. Operating the lighting console during the seven-minute performance required focus and precision, but no prior experience was needed. With guidance from a professional lighting technician, students quickly learnt the ropes and gained valuable hands-on experience in live production.

Movie making and editing

Students designed a seven-minute interactive digital backdrop and documented their process in a two-minute video. They built skills in digital storytelling, video editing, and creative tech. Using free tools like iMovie, Canva, and Adobe Express, they proved that innovation doesn't require expert software — just creativity and collaboration.

Music editing

Students carefully selected music to match the performance and storyline, then blended the tracks into a single cohesive soundtrack. Using free apps like GarageBand and Canva, they developed skills in music editing, sound design, and syncing audio with visuals, showing how creativity can thrive without professional software.

Overall, the feedback from students about this project has been truly outstanding. They not only had the chance to showcase their talents on stage and take part in a high-energy competition but also gained a deep understanding of the entire production process, from multiple career perspectives.

Now, when students attend a theatre, concert or event, they do so with a new level of curiosity and appreciation. They notice the lighting, the videography, the time invested in scripting, and the teamwork taking place behind the scenes.

Our aim is to provide students with innovative, real-world experiences that ignite their passions and prepare them for the future. Projects like this don't just build confidence — they open doors to new interests, career pathways, and a lifelong appreciation for the arts and technology. (HW)

SEEMA ZERAFA

As head of digital learning at Pembridge Hall, London, UK, Seema teaches computing to students from Year 1 to Year 6. She supports innovative cross-curricular projects, collaborates closely with specialist educators, and designs bespoke, real-world learning experiences that ignite creativity, foster critical thinking, and cultivate a deep love for project-based learning.

INTEGRATING SCRATCH INTO GEOMETRY

Joee Winter shares how she integrated Scratch projects as assessments in an advanced geometry class, instilling programming knowledge while also deepening students' understanding of geometry concepts

As a maths teacher, my computer science teaching journey began with a year-long professional development experience focused on integrating computational thinking and coding into existing classes.

During my professional development, I taught an advanced geometry course for ninth-grade students (14- to 15-year-olds). A couple of maths teachers who had previously completed the professional development mapped out project ideas based on content covered in a geometry course. I used this opportunity to shake up my approach to the course I was teaching.

This was an opportunity to shake up my approach to the course I was teaching

I used Scratch as the platform for these project ideas, as the block-based interface was accessible to all students and allowed those who were already comfortable with programming a chance to add complexity to their programs.

Traditional maths teaching practices use sequencing, where students learn some content, take a quiz, continue to practise, learn more content, and then take a summative test. While I still wanted to continue formative assessments in my course, I knew I could utilise Scratch projects as performance assessments, allowing students a new and different way to show their content knowledge.

Student learning with Scratch

Students started the school year off by creating a program in Scratch using one or more of the tutorials given on the platform. The lesson's goal was to allow students to get into the platform, learn about the workspace and options for personalisation, and successfully complete a program. I wanted students to feel comfortable with the platform before I asked them to complete a program such as a geometry assessment.

The first geometry Scratch project covered conditional statements. Students needed to include multiple conditional blocks, sprite interactions, and a 'forever' loop in their program. It was interesting to observe how students approached this first geometry task. I was always surprised by which of my students pushed themselves to do more with the program than the rubric demanded.

As their comfort level with the Scratch platform grew, so did the personalisation of the programs. A unit on lines and angles followed the unit on logic and reasoning. For this next geometry project, students needed to have a sprite draw two parallel line segments and one perpendicular line segment. I also provided constraints by requiring the parallel line segments to be drawn from quadrant two to quadrant four in the coordinate plane. They were also prohibited from placing a line segment through the origin. This project challenged the students to take time to figure out the slopes of their intended lines. Students had to decompose their problems to design a program that met the specifications for parallel and perpendicular lines, aligning with 2-AP-13 of the CSTA K–12 Standards (**helloworld.cc/csta-standards**). They had to work backwards to determine where

EXAMPLE PROJECT

Here is an example of geometry Scratch project directions for students:

This project is designed to help you better understand how to calculate trigonometric ratios of a right-angled triangle and how to use operators within coding.

For your script you need to have:

- A right-angled triangle background (sides and angles labelled appropriately)
- Variables for a, b, c (with user inputs for a, b, c), Sin A, Cos A, Tan A, Sin B, Cos B, Tan B
- Use of operators in the script to determine if there is a right-angled triangle using a, b, c
- Use of operators in the script to calculate Sin, Cos, and Tan for angles A and B
- Use of an 'if–then–else' block
- Use of a reset button to reset all values to 0 (or some way to reset within the program)

a perpendicular line would start and end. Students who took a shortcut to finish as fast as they could quickly discovered there was a bit more to determining the coordinates they used in the program.

The geometry project that challenged them most with decomposition was creating a calculator for right-angled triangles. During the trigonometry unit, students had to write a program to determine whether three given sides created a right-angled triangle, and then calculate the trigonometric ratios of the sides of the triangle. Students worked to find out how to construct the Pythagorean theorem using nested operations. It was a powerful experience with the very basic concept of order of operations, and for many, it took a few tries. When students needed help debugging, we broke down what parts of the equation were calculated first, and looked at whether this sequencing matched what was expected for the Pythagorean theorem. This was a much deeper experience than calculating the theorem for triangles using pencils and paper.

Reflections on the experience

As a maths teacher, geometry has always been my favourite course to teach. This was made even more enjoyable when I could watch my students succeeding at creating their projects. They provided opportunities for students to experience productive struggle, practising persistence and resilience.

In maths courses, students often learn a set of steps and are then asked to replicate the steps for new problems. The Scratch projects that students completed allowed them to think beyond a set of steps. I facilitated conversations with the class where we worked together to determine a plan to accomplish the given tasks. Students were able to learn that there were multiple ways to write a program to meet the same requirements.

The amount of discourse surrounding these projects was astounding. I am a firm believer in conversation as a way to deepen a student's knowledge. By being able to explain their learning, students understand the concepts on a deeper level. When students needed help debugging, they had to be able to express their intent with their program and what issues they were facing. While I helped my fair share of students work through their debugging, there were also many peers who were there to help support their classmates. When asked how Scratch projects were different from other geometry assignments, one student shared, "I appreciated the collaboration on the Scratch projects. It forced us to talk to others."

Student reflections

The engagement on these Scratch Days, as my students called them, exceeded our non-Scratch curriculum days. I think it was the first time in a long while that students were able to express themselves creatively in a maths course. Many times, I found myself giggling as I reviewed and graded their work, watching intriguing sprite interactions, costumes, and backgrounds. This engagement also led former students to sign up for computer science courses now offered at our high school. One of these students shared with me that the Scratch projects "reintroduced me to computer science concepts", and they were excited to work on the projects in class.

Learning together

For teachers looking to bring computer science into their content areas, consider how programming could help students demonstrate their knowledge within the subject area. It is also important to build in time for students to learn how to use the chosen platform. If I had not taken the time for students to complete tutorials and work through the projects independently, the student experience would not have been as smooth. It is important to show students that you are also taking a risk trying something new, and that you will be learning alongside them. My working relationship with my students changed when they saw my investment in learning something new, and I gained confidence in my ability to teach computer science. I can now proudly say I teach four different computer science courses. My experiences with integrating Scratch in geometry led me to my new journey in computer science education; maybe it can do the same for you. (HW)

JOEE WINTER

Joee is a computer science and maths teacher at Austin High School in Austin, MN, USA. She has been teaching for eighteen years, and has five years of experience as a computer science teacher (**linkedin.com/in/joee-winter-962888265**).

CROSS-CURRICULAR COMPUTING AND MODERN FOREIGN LANGUAGES IN THE PRIMARY CLASSROOM

Integrating computing and Spanish can boost creativity, strengthen digital and language skills, and bring learning to life

Cross-curricular teaching deepens learning by connecting subjects and fostering critical thinking, creativity, and real-world application. It builds transferable skills like communication, collaboration, and problem-solving.

At The Premier Academy in Bletchley, UK, we embrace this by using computing to enhance learning, particularly in Spanish. This purposeful integration strengthens understanding in both areas, nurturing digital literacy alongside key language skills such as vocabulary, grammar, reading, writing, listening, pronunciation, and speaking.

This article presents practical examples of weaving computing into primary Spanish lessons. These experiences align with the curriculum and foster confident, curious, collaborative, and reflective learners.

Typing in two languages

Touch typing, a fundamental digital skill that enhances fluency and confidence, is enriched by Spanish integration. Children type Spanish words and short sentences. We also use platforms like TypingClub's (**typingclub.com**) Spanish modules to build keyboard skills while reinforcing vocabulary. This simple approach transforms routine practice into a language-rich experience, supporting both computing and modern foreign language (MFL) goals.

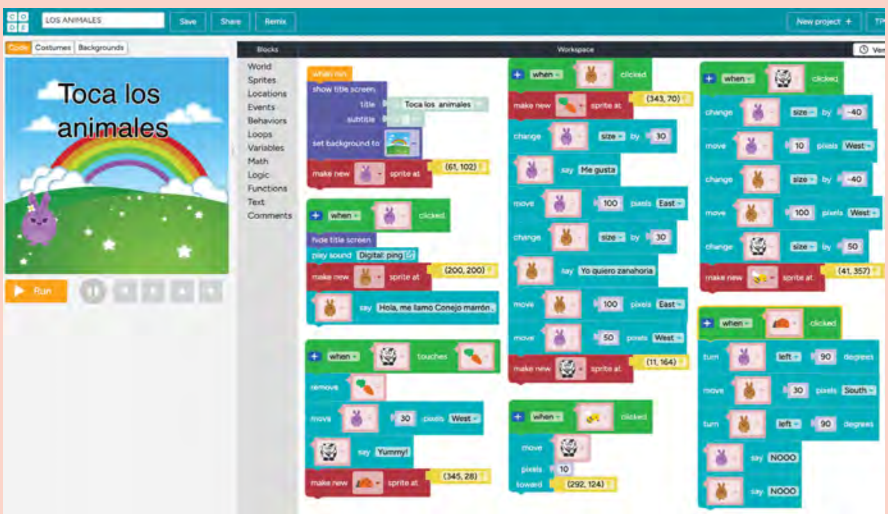
Creative coding in Spanish

Coding offers exciting cross-curricular opportunities through block-based tools such as Scratch and Code.org. Children can create interactive games, digital flashcards, and animated stories in Spanish. For example, when learning vocabulary around clothes or body parts, they can design projects in which clicking words triggers their pronunciation, or quizzes to test vocabulary. Advanced projects include animated conversations and interactive comics, or practising listening skills through embedded audio.

These activities strengthen coding logic and encourage meaningful, creative Spanish vocabulary use, fostering both computational thinking and linguistic competence.

Physical computing: hands-on Spanish

Physical computing makes abstract concepts tangible using tools like micro:bits and



An interactive Spanish game coded in Scratch

LEGO® Education WeDo 2.0. Children design, build, and program systems that respond to inputs like buttons and movement, and control them using sensors and motors.

Spelling out Spanish words on a micro:bit or creating a *piedra, papel o tijeras* (rock, paper, scissors) game reinforces vocabulary through repetition and fun. The micro:bit radio feature also enables the exchange of short Spanish messages, encouraging real-time language use. A weather project uses the temperature sensor in the micro:bits to display '*Hace frío*' or '*Hace calor*', linking science and Spanish.

WeDo kits involve building models and adding Spanish vocabulary to algorithms, like adding greetings or topic-related words to match programmed actions, making vocabulary memorable and purposeful.

In addition, collaborations with Spanish schools teaching physical computing and robotics offer opportunities to share our creations via platforms like Skype or Zoom. These sessions give pupils authentic opportunities to practise Spanish, explore cultural differences, and develop global curiosity and confidence.

Data handling with cultural relevance

When learning about databases, children gather Spanish-related information. In one project, they surveyed classmates about favourite foods or pets in Spanish and used spreadsheets to create charts, reinforcing vocabulary in an authentic context while also learning about data analysis and presentation.

Art, Spanish, and technology unite

For World Art Day, we combined art, Spanish, and computing with the theme 'Recreating Spanish Masterpieces with Technology'. Children used pixel art and digital drawing apps to reimagine works by artists such as Picasso, Dalí, and Miró, learning about the artists and Spanish culture, making art history accessible and fostering creative expression through technology.

Spanish animations

In animation units, children create short clips enacting Spanish dialogues or vocabulary-based scenes, building narrative skills, practising spoken Spanish, and enhancing digital storytelling. They learn animation principles and look at how digital tools create movement, linking these skills to Spanish topics, making the animations purposeful and engaging for both objectives.

AI for language analysis

We have begun to explore AI tools for language analysis. Children use translators and grammar checkers to analyse sentence structure and identify errors, reflecting on their writing and the differences between Spanish and English.

They also learn about the limitations of AI translation accuracy and the importance of their own understanding, providing opportunities to discuss ethical technology use and the limits of automation.

Digital presentations and word processing

Children use presentation tools for Spanish slideshows on topics such as numbers, food, transport, or weather, often adding voice-overs for pronunciation practice. They also use word processors to design leaflets for Spanish speakers, fostering research, editing, persuasive writing, and visual communication skills while also reinforcing Spanish vocabulary and grammar.

Game creation with a purpose

Game design is a favourite topic, in which children apply computing skills to create Spanish-language games. Children explore the history and impact of game development, learning what makes a digital game engaging, such as characters, objectives, rules, and interactions, applying these principles in their own designs and acting as testers. We use Purple Mash's tools (**helloworld.cc/purple-mash**) to build Spanish games tailored to different levels. Year 4 children (ages 8 to 9) create pairs games based on sports vocabulary, matching images with words or simple sentences. Older kids design maze games collecting *frutas* and avoiding *transportes*, recording Spanish words for in-game audio and writing instructions in Spanish, providing authentic language use.

A creative, connected classroom

Integrating computing with Spanish has transformed engagement at The Premier Academy, making abstract concepts concrete, building digital and linguistic confidence, and highlighting the interconnectedness of learning. These activities support our curriculum goals and cater to different learning styles through repeated exposure and feedback. This has made learning effective and enjoyable.

This approach gives computing skills purpose, enhancing communication, creativity, and inclusion through visuals, audio, physical interaction, and creative tools. Looking ahead, we're excited to explore new cross-curricular opportunities, like creating online safety posters in Spanish to display around the school.

Though this may be easier for a combined coordinator, computing and language specialists can collaborate effectively, sharing expertise and enriching their practice. While this article focuses on Spanish, the approach is easily adaptable to any MFL. My advice: start small, stay curious, and empower learners. The tools are accessible, and the possibilities are endless. So why not give it a try? Let's do it! (HW)

JESUALDO MARTÍNEZ MOLINA

Jesualdo is an assistant head teacher and the computing and languages coordinator at The Premier Academy, Bletchley, UK. He has over 25 years' experience teaching languages and computing in both Spain and the UK (**linkedin.com/in/jesualdomartinez**).

AI IS EVERYWHERE – EXCEPT THE SCHOOL CURRICULUM

Sue Sentance asserts that AI requires a more integrated and interdisciplinary approach to teaching within the school curriculum

In 2006, Larry Page, co-founder of Google, said that AI was going to be "the ultimate search engine [that] would understand everything in the world. It would understand everything that you asked it, and give you back the exact right thing instantly." In fewer than 20 years, the explosion of accessible LLM (large language model) technology has brought that vision to reality, and other capabilities of AI are impacting — and will further impact — everyone's lives, bringing both hope and nervous anticipation.

AI has been around for decades as an emerging and distant technology, but it's now part of our everyday language. This creates an urgency to ensure that young people leaving school are equipped to use AI-assisted technologies, with critical perspectives on their use, and some technical understanding.

In many of the world's curricula, we teach in discrete blocks. The question I hear all the time is, where should we put AI in the curriculum? Is it part of computer science? Or citizenship? Where should we teach fundamental data literacy? These are unanswered questions. In this short opinion piece, I want to pose the question, do we now have an opportunity to think more laterally about curriculum design and take a more integrative approach?

We need to distinguish between using AI as a tool and learning about AI, but teach both hand in hand. We should avoid making the mistakes of the past (which I won't dwell on here). AI is an interdisciplinary field, which means that to further our AI knowledge, we need people who understand maths, computer science, psychology, sociology, philosophy, linguistics, and so on. Future AI specialists will come from many different fields.

As an example, many of the current applications of AI that young people are in contact with revolve around the use of LLMs. In the context of learning our native language at school, we learn reading, writing, different genres of writing, and how to critically evaluate others' writing. This can all be related to the use of an LLM: we want young people to be able to use LLMs confidently, to learn how to write effective prompts (a new kind of literacy), and to critically evaluate the output.

To augment an understanding of what an LLM gives in response to a prompt, it might be useful to link our language lessons, where we are introduced to the effective use of LLMs, to other subjects: to probability and approximation in mathematics, neural networks in computing, and bias and fairness, from wherever ethics is taught in the curriculum, as a few examples.

Once we pull in all the threads of what an LLM is and how to use the technology effectively, horizontal coherence (**helloworld.cc/horizontal-coherence**) is needed to ensure that showing connections between subjects is timely and effective. However, I believe we are a way off from being able to do this effectively because of the discrete nature of our education systems.

What we teach about AI in the school curriculum and how we teach it will need much research and discussion: we are barely at the beginning of this journey, despite the urgency felt by many schools. We need to be patient, but also brave. My concern is that gradual tiny tweaks to the curriculum will not support what we need to do. We need to teach in a more integrated way in school. We also need to teach about AI. This moment may be a good opportunity to bring those two needs together. (HW)

SUE SENTANCE

Sue is the director of the Raspberry Pi Computing Education Research Centre at the University of Cambridge, UK. Formerly a computing teacher, she leads a number of research projects relating to computer science and AI education, including programming pedagogy and physical computing.

CODYSSI: AN INTERNATIONAL STUDENT-LED CODING COMPETITION

Student Pip Purttiponthanee and his teacher Patrick Kennedy discuss Pip's creation of Codyssi and how to support exceptional students

Are you passionate about problem-solving or interested in learning to code? If so, then Codyssi is a great opportunity for you. I am Rajchid (Pip) Purttiponthanee, and I created Codyssi (codyssi.com), an international coding competition made for anyone who is at least 13 years old.

When I was 15, I had already participated in many coding competitions, including Eric Wastl's Advent of Code (**adventofcode.com**) and Paul Baumgarten's Coding Quest (**codingquest.io**), and I really enjoyed solving their coding puzzles. The challenge of problem-solving and the feeling of pride when I succeeded were great experiences for me. After a supportive, inspiring discussion with my computer science teachers, Mr Patrick Kennedy and Mr Chris Hall, I decided to create and host my own coding competition.

The official 2025 contest round poster — the QR code takes you to Codyssi's website

The Codyssi coding competition

Creating and organising Codyssi was quite a tall order. Over the past year, I've worked on numerous tasks, including devising the challenges, designing the website, writing the plots for each problem, drawing the poster, producing the trailer, coordinating advertisements, and so much more! I've enjoyed the challenge of producing Codyssi — bringing it to life using a diverse range of skills is something I am proud of.

Codyssi is — to my knowledge — the first international, student-led, online coding competition from Thailand, and I'm really proud of doing something unique!

As a fun fact, Codyssi's name is a blend of the words 'code' and 'odyssey'. This is because I decided to include some themes and characters from Greek mythology in the plots of some of Codyssi's problems.

Why did I make Codyssi?

Aside from my passion for problem-solving and the challenge of creating a coding competition, I had many other motives to create Codyssi.

Firstly, I wanted to provide more opportunities for aspiring coders and students of all abilities to learn, compete, and develop their passion for coding and solving problems. For this reason, I tried my best to ensure that everyone could benefit from participating in Codyssi, by designing coding problems to engage and challenge participants at a variety of different skill levels.

In addition, I wanted to encourage friendly competition, both within and between schools. So, I added a feature allowing schools to register for Codyssi so that their students can compete on the

	Sample round	Contest round
Challenges	4	18
Participants	120+	800+
Registered schools	6	24
Countries involved	4	13

Codyssi's current stats

global leaderboard. Registered schools also receive a private school leaderboard, so that students can compete with staff and other students within their school.

Finally, I believed that Codyssi could be a source of inspiration for other students who were looking to create and develop their own personal projects.

COACHING EXCEPTIONAL STUDENTS

Some students have a natural talent for coding, and Pip is undoubtedly one of them. Having taught him computing for several years before he began IGCSE Computer Science aged 14, I was well aware of his strong interest in programming. From the start, it was clear that the standard curriculum wasn't enough to challenge him.

To push his abilities further, I encouraged him to participate in online coding competitions like Advent of Code (**adventofcode.com**), and he was immediately captivated. For context, Advent of Code is an annual online event running from 1 to 25 December, featuring daily programming challenges of increasing difficulty. Participants can use any programming language to solve the puzzles. However, Advent of Code is not designed as an entry-level competition, and is certainly not tailored for school students.

What stood out most about Pip was not just his technical ability but also his independence, curiosity, and, most impressively, his deep consideration for what others could achieve. He was driven by a desire to help others discover coding and experience the excitement of competitive programming. Recognising that many students found coding competitions intimidating or out of reach, he took the initiative to design a competition tailored for his peers. His goal was to create an engaging and accessible challenge that would inspire more students to develop their programming skills and gain confidence in problem-solving. My role was guiding him through listening to his ideas, offering occasional suggestions, and watching his passion and drive take him forward. The end result is quite extraordinary, but don't take my word for it. Try a challenge at **codyssi.com**!

Structure of a Codyssi problem

If you have looked at some Codyssi problems, you will probably have noticed that there are some common features that all Codyssi problems share.

Firstly, each Codyssi problem has a title and an associated drawing. Almost all of the drawings that correspond to Codyssi problems were drawn by my (very talented) friends that volunteered to produce art for Codyssi! Here's another fun fact: I drew one picture for a problem in the 2025 contest round — can you guess which one?

Each Codyssi problem has three parts that build on each other, and each part has an assigned difficulty level from 1 (easy) to 6 (very challenging). As a bonus, each part's challenge is embedded in a prompt that includes a short story, which you will hopefully find entertaining!

When solving a Codyssi problem, you are given input data that you will need to use to solve the problem. Each user's input data is randomly generated, so your inputs are very likely to be unique.

Generally, problems get more difficult throughout a problem set. However, this is not always true, as there could be more difficult challenges towards the start of the problem set. Each problem set also has an overarching plot that is consistent in the prompts for all of its problems.

Codyssi's 2025 contest round

During December 2024, I hosted a short sample round on Codyssi's website to test whether my systems would function as I intended. The sample round consisted of four problems, and it was really successful! Aside from Codyssi's website functioning properly, there were also over 120 registered users and six registered schools, and I received positive feedback from many participants.

Codyssi's main event in 2025 was the 'Journey to Atlantis' contest round, which began on 17 March, 2025. The competition consisted of 18 problems spanning the entire range of difficulty ratings from 1 to 6. While the overarching plot followed a perilous trip to Atlantis, the problems also featured characters, locations, and themes from the Greek epic poem *The Odyssey*. Some participants may have even recognised familiar figures or settings in the artwork for each problem.

At the time of publication, Codyssi had over 800 registered participants and 24 registered schools, from 12 different countries across three continents. These numbers highlight the competition's growing international reach, with the potential for even greater expansion in the future.

We hope that all those who participated found the experience enriching, that they learnt something new, and that they deepened their passion for problem-solving and coding! (HW)

RAJCHID (PIP) PURTTIPONTHANEE

Pip is a computer science student in Year 12 at Bangkok Prep, Thailand. He is passionate about maths and computer science, and intends to study maths at university.

PATRICK KENNEDY

Patrick is a computer science teacher and innovator, formerly at Bangkok Prep and now working at the British International School, Phuket, Thailand. He has over 25 years of teaching experience, specialising in technology and STEAM education.

PROGRAMMING IN PARADISE

Leah Aiwohi shares her experience of teaching computing in rural Kaua'i

Growing up and returning to teach on the tranquil island of Kaua'i, Hawai'i, my life was deeply intertwined with the ocean's rhythm and a strong sense of community. Our small town, linked by winding roads, fostered resourcefulness and resilience among generations of interconnected families.

Education faced unique challenges here. Outdated textbooks, delayed supplies, and a lack of technology were common. Our school lacked computer labs, reliable internet, and teachers with technical expertise. My early learning emphasised practical skills — fixing, mending, and solving problems with limited resources. Computer science was an unfamiliar concept, not due to a lack of student ability, but because it simply wasn't introduced.

A path to computer science

My journey into technology began unexpectedly in 1990, my first year of teaching. The arrival of a single desktop computer, our school's first, was a pivotal moment. Connecting it via a dial-up modem to a satellite dish on O'ahu, we initiated our first e-pal project, linking my students with a class across the ocean. This experience revealed technology's power to connect and transcend geographical boundaries, planting a seed of possibility.

My path to computer science (CS) was far from straightforward. Without formal exposure, I navigated my journey with curiosity and a desire to help others learn. An educational technology course during teacher training opened my eyes to the potential of digital tools for instruction and equitable access. While I was fascinated, a clear path into CS remained elusive.

I proactively sought out workshops, online forums, and professional development, piecing together enough knowledge to introduce basic tech to my classroom. Often, I found myself learning just ahead of my students, coding or setting up networks late into the night to bring them to life the next day. Mentorship from dedicated educators, both local and from afar, was crucial, reminding me that my understanding of my students and community, coupled with my willingness to learn, were valuable assets.

Overcoming hurdles

Access remained a significant hurdle. Slow internet, costly off-island travel for training, and budget constraints due to high shipping costs were constant challenges. The complexities of logistics, even down to essential lithium batteries, highlighted the disadvantages of our remote location. Yet with each obstacle overcome, my conviction grew: CS was essential for my students, offering opportunities equal to those enjoyed by their peers elsewhere.

My initial motivation for becoming a teacher was to give back to my community and to be the educator I had longed for.

Teaching computer science in a rural Pacific community presents unique obstacles

Returning home to teach was not a choice but a calling. I recognised the immense potential within our students, and wanted to help them realise it.

Integrating computer science

Computer science gradually integrated into my classroom, evolving from small projects to digital media lessons and STEM integration. This blossomed into a passion as I witnessed my students' enthusiasm when creating something tangible from abstract concepts. Early projects such as building websites about local legends demonstrated technology's power in sharing culture and identities.

The initial stages were challenging. Lacking formal CS training, I learnt alongside my students, navigating technical difficulties like Wi-Fi outages. Yet these challenges were interspersed with inspiring moments, such as the quiet student who excelled at creating a game about environmental stewardship. These instances underscored that CS was not merely about code, but about meaningful storytelling, creativity, and problem-solving.

Representation was also paramount to introducing CS to my classroom. The absence of individuals in tech who look like my students, or who come from similar backgrounds, can lead students to believe they don't belong. Culturally responsive teaching is therefore core to my approach. I strive to ensure students recognise their values, language, and experiences as strengths in this field. When they code with a sense of purpose, grounded in their identities, true innovation flourishes.

Unique problems and solutions

Teaching CS in a rural Pacific community presents unique obstacles. Unreliable internet, outdated equipment, and limited exposure to STEM careers are realities for our students. Some have never used a laptop outside of school, while others balance academic pursuits with significant home or work responsibilities.

However, these challenges have fuelled innovation. We've forged partnerships with local organisations to improve device access and internet connectivity. Remote learning platforms have enabled connections with guest speakers and mentors. Our projects connect coding with local issues, cultural narratives, and core values such as environmental stewardship.

One student developed a mobile app prototype for tracking invasive species, merging knowledge of native plants with data collection and interface design. Another created an interactive digital story about his ancestral land division, incorporating visuals and audio interviews. These projects became avenues for reclaiming voice, honouring heritage, and applying technology in relevant ways. Witnessing students taking ownership of projects that bridge technology and their lived experience reinforces the significance of this work and the necessity of pairing access with authenticity.

Elevating everyone

While my primary commitment is still in the classroom, I've recognised the need to influence decision-making if we are to transform access to CS. This has led to my involvement in national fellowships, conference presentations, curriculum development, and service on boards dedicated to equity in education. Recognition as Kaua'i's Teacher of the Year twice, a Lifetime Achievement Award for Career and Technical Education, and National DARE Educator of the Year are honours, but their true value lies in inspiring the next generation. My hope is that my journey empowers my students to forge their own paths and surpass my achievements.

As an educator from a rural Pacific community, I bring a crucial perspective often absent from national dialogues about CS. My role has been to emphasise that equity encompasses geography, infrastructure, and cultural relevance, not just race and gender.

I've contributed to initiatives advocating for rural inclusion in K–12 CS pathways (ages 5–18), stressing that culturally responsive teaching is essential. Mentoring educators and collaborating with national organisations are ongoing efforts to ensure CS education does not leave isolated communities behind.

My aspiration is for my students to believe they can succeed in technology without leaving their islands. I want them to recognise that innovation can originate here, rooted in their land and their stories. My hope is that CS education in rural areas will be about leading with creativity and culture.

How you can help

Policymakers and tech leaders, I urge you not to overlook rural communities. We need sustained investment in reliable internet, modern equipment, and long-term funding. Policies should allow for flexible delivery of CS education, acknowledging unique contexts, and pathways for local educators to gain CS credentials within their communities should be supported. Fellow educators, continue building bridges and advocating for culturally relevant curricula. Partner with local organisations and families to ground tech learning in local issues. (HW)

LEAH AIWOHI

Leah has been a public-school teacher in Hawai'i throughout her 34-year career as an educator. Having taught at the elementary, middle-, and high-school levels, she currently teaches media production, computer science, and STEM education.

CODE CLUBS IN SOUTH AFRICA

Rujeko Moyo explores the contexts, challenges, and insights

The contexts and challenges experienced by our Code Club partners in South Africa (SA) are quite different to those encountered by UK-based clubs (although there are some similarities). Drawing on my background of living in South Africa for over a decade and delivering educational programmes in low-income South African state schools, I felt we needed to understand how Code Clubs were really faring in South Africa in order to inform our approaches. This is my account of my tour of the Code Clubs in these communities.

From indigenous games to block-based coding: insights from Limpopo

Just before travelling, I had an insightful online meeting with 'Techno Granny' Phuti Ragophala, a retired school principal championing coding in rural Limpopo, one of South Africa's provinces. I was enlightened by Phuti's approach to training teachers.

She starts where the teachers are, connecting with their lived experiences and culture by linking coding concepts to the indigenous games played regularly by children in Limpopo. She then builds on this cultural connection with clearly and simply labelled cards that reinforce coding concepts. She further develops their understanding of coding concepts through Tangible Africa resources (**helloworld.cc/tangible-africa**), which utilise cardboard tokens, a mobile phone, and their 'Rangers' app. Many teachers in rural Limpopo own a smartphone and the app is 'zero-rated' so they don't incur mobile data costs while using it.

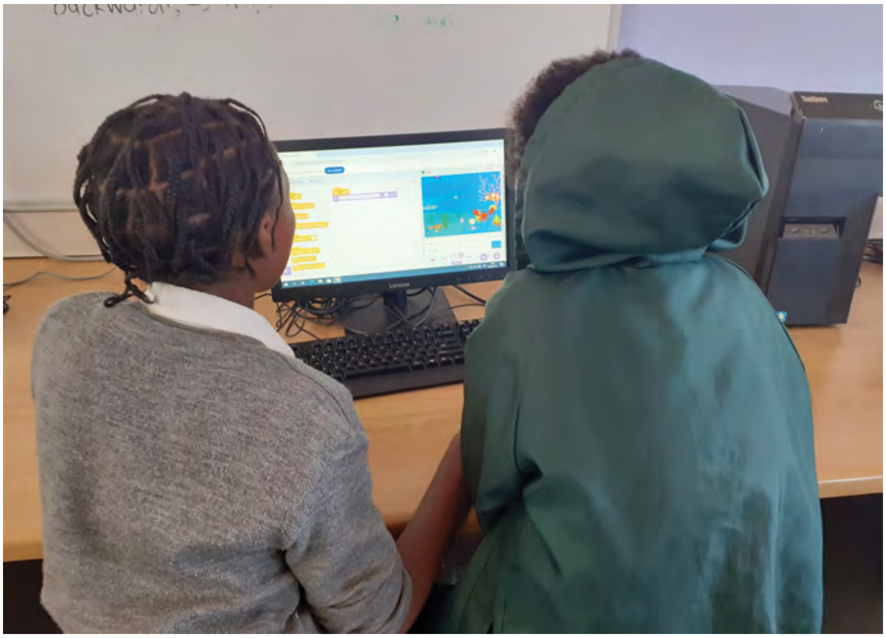
However, Phuti does not stop at unplugged coding. Her training proceeds to introduce teachers to block-based coding through open-source apps and websites like OctoStudio (**octostudio.org**), ScratchJr (**scratchjr.org**), Scratch (**scratch.org**), and MakeCode (**makecode.org**). For teachers that can access these open-source resources, having a solid foundation in coding concepts through familiar unplugged coding activities makes for easier exploration and progression through the otherwise unfamiliar world of block-based coding.

Visiting the digital democracy team at Keep a Child Alive in Durban, South Africa

Insights from Code Clubs in Durban

During the SA trip, I learnt about the contexts and challenges of Code Clubs in Durban schools while working with our Code Club partner, the digital democracy team at Keep a Child Alive (**keepachildalive.org**). We visited ML Sultans St Mary's Primary School (a mainstream school) and Open Air School (a special education school). Both schools facilitate coding sessions in well-resourced computer labs, during a rotational class period for different classes during the school day, not as an after-school club that is limited to interested learners who live locally to the school. This enables them to engage every learner in coding. The in-school coding sessions are also welcomed because they are aligned to the requirements of South Africa's coding and robotics curriculum.

The fourth-grade class (10- to 11-year-olds) of 44 learners at St Mary's Primary, who had been progressing through the 'Intro to Scratch' pathway (**helloworld.cc/scratchintro**), were collectively on the third project 'Find the bug', although a few were already independently tackling the fourth project, 'Silly eyes'. Their teacher, Mr Mahendren Reddy, led the majority of the class through the project step by step. Although it was great to observe their guided agility through the project, Mr Reddy



Coders at Ikwezi Lesizwe Primary Code Club creating a story-based coding project together

explained that it had taken several weeks to get his learners to that level of confidence. The learners had initially struggled to read through the project instructions until their teacher advised them to focus on the pictorial coding scripts. Another factor that slowed their progress was that learners only got the opportunity to code once a week during a coding lesson at the computer lab, and even then, some took turns to code while sharing a device. The majority did not have access to computing devices outside of the one-hour coding lesson in the ICT lab.

Despite these challenges, it was very encouraging to observe the school leadership's buy-in to the aim of advancing coding for all their learners. Although they are a regular state school, they sourced their own external funding to resource and maintain an ICT lab for their learners. This is sadly not the status quo for many state schools in Durban, which have neither computing equipment nor the means to external funding.

Insights from Code Clubs in Cape Town

I also spent time with our Code Club partner in the Western Cape, Coder:LevelUp (**coderlevelup.org**), visiting two Code Clubs located in low-income communities. Their key difference from Code Clubs in Durban is that they are delivered as after-school clubs, and are attended by learners who live locally and do not need transport to and from school. While the learners in both schools were aware of our Code Club projects, they were creating their own projects on Scratch and ScratchJr when I visited.

At the Code Club at Ikwezi Lesizwe Primary School for sixth-grade learners (11- to 12-year-olds), I observed that the girls were more inclined to creating storytelling projects that revealed their interests and experiences, while the boys were more fascinated by soccer-themed projects. I also observed that their teacher, Mr Masibulele Njeza, went the extra mile for his learners, facilitating the Code Club during the school week and also on weekends for several hours. This was not only for students' educational benefit but also to create a safe space for them to thrive and learn amidst the challenges and pressures of living in a low-income community.

Teacher training and support

From meetings with both our Durban and Cape Town partners, the following teacher training and support needs were clear:

- Teachers preferred and valued in-person training
- Teachers needed to be trained in digital literacy before moving onto coding literacy
- Teachers communicated, accessed content, and shared ideas through WhatsApp

Student support

To better support students in South Africa and other communities lacking access to computing resources, we are working on:

- Code Club projects with minimal written instructions
- Code Club projects with themes that are easily adaptable to a wide range of cultural and local contexts
- Unplugged activities that develop computational thinking and basic coding concepts

We look forward to the positive impact of these resources on our Code Clubs globally, and especially in South Africa. (HW)

RUJEKO MOYO

Rujeko works at the Raspberry Pi Foundation as the Code Club community coordinator for England. She works with schools and libraries across England, supporting them to start and run their own Code Clubs.

ADVOCATING FOR NEURODIVERGENT STUDENTS IN COMPUTER SCIENCE EDUCATION

Dan Jones discusses recent initiatives to support neurodivergent students in computing

As a CSTA equity fellow and advocate for neurodivergent students in computer science, I am passionate about ensuring that all students, regardless of their neurological differences, have access to and can thrive in computer science education. Neurodivergent students, including those with ADHD, autism, dyslexia, and other conditions, bring unique perspectives and strengths to the field of computer science. It is crucial that we create inclusive environments that recognise and support these diverse talents.

In recent years, the push to support neurodiversity in computer science has gained significant momentum. This is crucial for creating an innovative future workforce, where varied perspectives and unique problem-solving approaches are valued.

Creating inclusive labs

In my role, I emphasise the importance of designing lab experiences that are accessible and engaging for neurodivergent students. This includes:

- **Flexible lab structures:** providing options for students to work at their own pace and offering multiple ways to demonstrate understanding
- **Clear instructions:** using straightforward, step-by-step instructions to minimise confusion and anxiety
- **Visual aids:** incorporating diagrams, flow charts, and other visual tools to support different learning styles

Supporting neurodivergent students

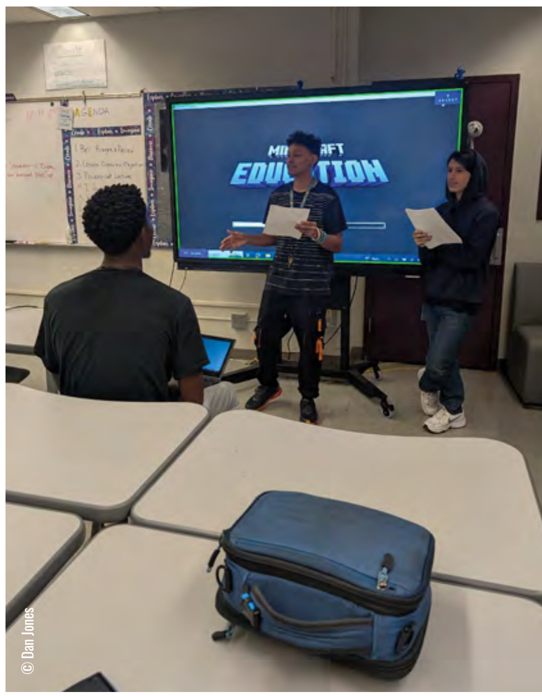
To support neurodivergent students in computer science, we must:

- **Foster a growth mindset:** encourage students to view challenges as opportunities for growth and learning
- **Provide accommodations:** offer accommodations such as extended time, quiet workspaces, and assistive technology
- **Promote collaboration:** create opportunities for peer collaboration, allowing students to learn from and support each other

Advocacy and awareness

Advocacy is a key component of my work. I always strive to raise awareness about the needs and strengths of neurodivergent students by:

- **Educating educators:** providing professional development for teachers on neurodiversity and inclusive teaching practices; participating in professional development cohorts and then sharing what is learnt with colleagues
- **Engaging families:** collaborating with families to understand their children's needs and to create supportive learning environments; opening lines of communication are a must between teacher and parents/guardians
- **Influencing policy:** working with policymakers to ensure that educational



© Dan Jones

Universal Design for Learning practices include allowing students to choose how to complete their assignments

Neurodivergent students bring unique strengths to the field of computer science

policies are inclusive and equitable; being active with my local Computer Science Teachers Association (CSTA) chapter and presenting at conferences.

Recent developments

The University of Florida's Department of Computer and Information Science and Engineering (**helloworld.cc/uf-cise**) is making significant strides in adapting computer science education to meet the needs of neurodivergent learners. Their INFACT project (**helloworld.cc/infact**) focuses on recognising and leveraging the strengths of neurodivergent students to prepare them for successful careers in computer science. Universal Design for Learning (UDL) strategies (**helloworld.cc/UDL4CS**), such as providing multiple means of representation and expression, are integral to this approach. High-Leverage Practices (HLPs), including collaborative teaching and scaffolded supports, further enhance the learning experience for all students (**helloworld.cc/HLP**). Here are some examples of each practice:

UDL practices

- **Flexible workspaces:** classrooms have workstations for individual work, group work, and teacher guidance
- **Multiple information modalities:** information is presented through print, audiobooks, videos, and verbal instructions
- **Assignment options:** students can complete assignments in various formats, such as essays, podcasts, videos, or comic strips
- **Digital and audio texts:** materials are accessible in print, digital, text-to-speech, and audiobooks, with options for text enlargement and screen-colour adjustments

HLP practices

- **Collaborative teaching:** teachers work together to plan and deliver instruction, ensuring consistency and support for all students
- **Data-driven planning:** instruction is guided by data on student performance, allowing for targeted interventions and supports
- **Instruction in behaviour and academics:** explicit teaching of social, emotional, and academic skills through structured formats
- **Intensify and intervene as needed:** providing additional supports and interventions for students who require them

Extracurricular activities

Beyond the classroom, extracurricular activities like esports and computer science clubs play a vital role in fostering a sense of belonging among students. Esports programmes offer a unique opportunity for students to connect and collaborate in a competitive yet inclusive environment. They can help foster a sense of connection to school and help students to see themselves as someone who can work in a team setting, as a leader and a supportive player. These programmes help students develop critical skills such as teamwork, problem-solving, and digital literacy, which are essential for success in both academic and professional settings.

Computer science clubs also contribute to creating an inclusive school culture. These clubs provide a space for students to explore their interests, share ideas, and work on projects collaboratively. My goal as the esports, robotics, and drones coach and club sponsor has been to create a safe space where creativity, collaboration, and individuality are not only supported, but are celebrated and honoured. By participating in these activities, students enhance their technical skills and build meaningful relationships with their peers, fostering a sense of belonging and community within their school.

Personal reflections

As someone deeply committed to equity in education, I have seen first-hand the incredible potential of neurodivergent students. Their creativity, problem-solving, and unique ways of thinking are invaluable assets to the field of computer science. By fostering inclusive environments, we can help these students succeed and contribute meaningfully to the world of technology.

Neurodivergent inclusion in computer science is not just about providing access; it is about creating a culture of acceptance and support where all students can thrive. Together, we can build a future where diversity in thought and experience is celebrated and leveraged for innovation and progress. (HW)

DAN JONES

Dan is an experienced Exceptional Student Education (ESE) teacher with 17 years in education, including 4 years dedicated to teaching computer science at Broward County Public Schools in Florida, USA. He is a cohort-1 member of the University of Florida CS Inclusion programme and was a CSTA equity fellow for 2023–24. Dan is passionate about fostering inclusive practices in computer science education to empower neurodivergent students and promote diversity in the tech industry (**daniel.jones@browardschools.com**).

FEATURE


© Sanele Gobinduku/peopleimages.com/stock.adobe.com

My female students have asked if working in tech means you work on your own

HERE COME THE GIRLS!

Janine Kirk shares tips for making computer science more appealing to girls

For the 16 years I have been teaching computing, I have continuously wondered what it would be like to teach a class of girls.

I was so tired of looking at my computer science GCSE class at the start of Year 10 (ages 14–15) with that sinking feeling. I am a girl in computer science, and yet girls don't think computer science is for them. What am I doing wrong? I have tried so many times telling girls that they are in demand in computer science, but without much success.

What's stopping girls?

I try to encourage any girls in my classes to throw themselves in, but there are often no girls at all in a class, or perhaps just one. This bugged me so much that I decided to study this for my dissertation for a PGDip in Psychology. Was it that girls didn't relate to me? Maybe a middle-aged female wasn't the best role model to make something seem good. Or was it that they just found the subject dull? How could I make my lessons more fun?

Then I noticed something. Three statements stuck in my mind: "We need more girls in computer science", "Computer science is the hardest GCSE", and "Computer science is maths-heavy." With that we literally had a recipe for no girls ever taking the subject.

Each statement was adding another reason to not choose computer science. First, there are no girls, so you will be the only girl. Second, it's really hard — not a great concept for a gender that stereotypically is risk-averse. And it's maths-heavy, and maths as a subject has its own issues with uptake by girls. Let's add that the gaming industry and high-value tech giants are majority-male and it's not really any wonder we struggle to get a good gender balance. Would you study a subject where you thought you would fail and would most likely feel quite isolated? It's not a great start.

Making computing appealing

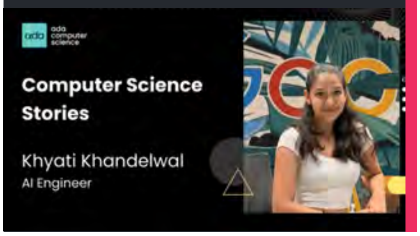
So, I thought about what we could actually do about this. For starters, I banned the word 'hard'. Computer science is not hard — it takes practice. Making mistakes in tech development is exactly that: development. Nobody ever gets the code right straight away.

Let's make sure we stop to remember that if we keep saying, 'We need more girls', maybe girls are hearing, 'You will be the minority.' Instead, let's focus on what computer science actually is: solving real-life problems with the speed of a computer. Let's concentrate on the human element. Where is computer science actually used? How can we create programs that are inclusive and help humanity? How do we spot bias? Let's remind our young people that if you want to create change, you have to fight for it.

Using practical examples really helps students see the real-world impact. I used an excellent example from Minecraft Education

NEED HELP?

I found that the 'I Belong' training with England's National Centre for Computing Education (NCCE) really helped (**helloworld.cc/i-belong**). Tech She Can (**helloworld.cc/role-models**) and Ada Computer Science (**helloworld.cc/ada-cs-stories**) also share role model stories. There are excellent resources for classroom displays to encourage the message that women in tech are here to stay!



(**helloworld.cc/minecraft-ai**). Students worked through programming activities to solve real-world problems where the code they developed had to be inclusive. The whole time they were programming, I could see great problem-solving and excellent discussions about bias, including gender and disability.

Another way to make computer science more appealing is to look at lesson structure. I have found that girls really enjoy coding together and working on solutions. The PRIMM (Predict–Run–Investigate–Modify–Make) model of teaching programming really plays into this (**helloworld.cc/primm**). It allows students to look at problem-solving elements and really delve into the program. That is much more interesting than getting a cat to do a little dance in Scratch; finding problems and solving mysteries is much more appealing. Using 'driver' and 'navigator' roles in pair programming has also helped, as it's a great opportunity for girls to flex their communication skills and enjoy social interaction and team-building opportunities (although we still have to remember this does not work for every girl). Having some strategies to replace a traditional coding lesson allows all students to improve their knowledge.

> IN ORDER TO CREATE CHANGE, YOU HAVE TO BE THERE FIGHTING FOR IT

The hard sell

Not to be distasteful, but in my teaching experience, girls seem to really care about the money! We have had some great speakers in school this year who have very kindly shared how much they earn. When I repeated these numbers in a school event, parents started to encourage computer science as a career pathway. In fact, I seriously considered a job switch! Having industry speakers has helped student and parent understanding. When computer science professionals were explaining about working as part of a team, or looking at solutions like a detective, students really seemed to take note. Interestingly, when I arranged for a mixed-gender group talk, the boys dominated with questions like, 'Can you tell me how to mine Bitcoin?' or, 'What's the biggest company you have hacked?' The girls sat silently, not asking their questions, and were lost amidst the boys' enthusiasm. So, I started to mix this up and created a separate female-only Q&A session with female speakers.

This does seem a little counter-intuitive after saying we need to stop highlighting gender imbalance. But I found that separating the students by gender allowed the girls the freedom to ask different questions, like 'How often do you work from home?', 'What's a day in your job actually like?', and 'Do you work on your own?' Their questions were more about the experience of working in the tech industry. Of course, the female students also asked, 'Have you ever been treated differently because you're a girl?'

A balanced future

We need to remember that we can make the curriculum more appealing to girls; it just takes a little work. We absolutely must be mindful that the reason we want more girls in computing is because it will assist in addressing coding bias and give us more inclusive technology. As I say to my students, "If we have one gender creating our codes, our codes will only work for one gender."

Our students choose if they want to continue studying computer science as one of their options for Years 10–11 (ages 14–16). I asked my line manager about the cohort signing up for next year, "How are we doing for girls?" Imagine when she turned to me and said, "It's mostly girls." She must have seen the shock on my face, because she had to repeat it a few times. I'm starting to believe my work to address gender balance is paying off ... maybe too well! (HW)

HEAR MORE FROM JANINE ON OUR PODCAST, WATCH THE EPISODE HERE
helloworld.cc/pod27-code

JANINE KIRK

Janine is the GCSE computer science lead, exam assessor, and teacher of both lower secondary ICT and computing at the King's CofE Academy, Kidsgrove, UK. She has 16 years of experience in leading and teaching an ever-changing curriculum.

FEATURE

BRIDGING THE DIVIDE: CONNECTING GLOBAL COMMUNITIES WITH EXPERIENCE AI

How education can serve as a tool to connect communities in understanding and engaging with AI

As AI technology evolves and plays an ever-greater part in our lives, it also plays an increasing role in the global vision for the future. As more and more governments around the world create AI strategies for the decade ahead, many are identifying an urgent need to address the gap between the creation of AI-related careers and the skills and knowledge young people need to develop. This gap is greater still when creating opportunities for educationally underserved communities.

The value of a global network

Experience AI was co-developed in 2022 by the Raspberry Pi Foundation and industry experts at Google DeepMind with a clear mission: to equip teachers with free, accessible, easy-to-use classroom resources that build AI literacy from the ground up (**helloworld.cc/exp-ai**). The programme offers a suite of materials to help students understand the real-world applications of AI, the basics of machine learning, and the ethical considerations that come with using these technologies.

In 2023, the Raspberry Pi Foundation expanded Experience AI internationally, launching a pilot programme through a group of existing partners. The pilot generated significant demand, received highly positive feedback, and reached an estimated 1 million young people, establishing a solid foundation for broader expansion. In late 2024, with support from Google.org, the programme grew significantly and we tripled the size of our partner network to 21, with new organisations joining from across Europe, the Middle East, and Africa.

Each partner in the Experience AI network is a unique organisation looking to create lasting social change. Through their local knowledge and the expertise of our network, we can present Experience AI to educators and students in a way that is engaging and understandable for local communities.

Partners not only train thousands of teachers on how to use the materials but also help us to adapt and translate our resources, all while making sure that the core pedagogy and design principles of Experience AI are preserved. Through our most recent expansion, we aim to reach an additional 2.3 million young people by December 2026, helping them to gain the knowledge and skills to engage with AI confidently in an ever-changing world.

Even more importantly, these organisations are providing educators with access to free training support, and work to reach a variety of communities that may have otherwise never had the opportunity to learn about AI.

Building communities

The Latvian Safer Internet Centre (LSIC), an initiative of our partner the Latvian Internet Association (LIA, **lia.lv**), is dedicated to helping young people protect themselves online and preparing young people for a fast-changing digital economy. As an Experience AI partner, they aim to train 850 teachers and support 43,000 students to build a strong foundation in AI literacy through the programme.

"We hope to spark a cultural shift in how AI is ... taught in Latvian schools. Our goal is for AI literacy to become a natural part of digital competence education, not an optional extra."

Based in Riga, the team is travelling to 18 different regions across Latvia to bring in-person professional development to teachers, including those in rural communities far from major cities. By meeting teachers where they are, the LIA is creating invaluable networks for learning and support between communities. Through hands-on training, they are also supporting teachers to bring Experience AI into their own classroom, creating examples which are relevant and engaging for their learners.

"We chose an in-person training model because it fosters a more collaborative and engaging environment — especially for teachers who are new to AI. Many educators, particularly those who are less confident with

The Experience AI Latvian launch on Safer Internet Day

digital tools, benefit from direct interaction, real-time discussions, and the chance to ask questions in a supportive setting."

As an Experience AI partner, the Latvian Internet Association is not just delivering content but working to strengthen digital competency across the country and ensure that no teacher or student is left behind in Latvia's AI journey.

One teacher shares: "The classroom training was truly valuable — it gave us the chance to exchange ideas and reflect on our diverse experiences. Hearing different perspectives was enriching, and I'm glad we're shaping the future of our schools together."

Embracing AI education

In Spain, it's clear that teachers no longer see AI literacy as optional — it's essential.

A recent survey conducted by our Spanish partner Fad Juventud (**fad.es**) revealed that six in ten students feel there is insufficient and inadequate training on AI in schools, despite over 80 percent of them already using AI tools. And with only 56 percent of Spain's population possessing basic digital skills, according to the European Commission, the urgency is clear.

To address this gap, Fad Juventud has committed to training 2000 teachers and reaching 82,500 young people across Spain by the end of 2026.

Their online teacher forum highlights how educators across disciplines are not only curious about AI but eager to embrace it.

For example, Yolanda, a mathematics teacher of 25 years, sees AI as a vital part of preparing students for the future:

"We live in a world of change, and with AI, the changes will be even greater and faster. I think that is what we have to teach our students: to adapt to the new era."

Azahara, a philosophy teacher, agrees: "AI is another tool for us, and it is the tool for our students. We must know it well to lose our fear of it, accept it, and get the best out of it."

For Mª. Teresa, a secondary-school music teacher, AI's impact on her field has already been profound: "In just three and a half years, AI has completely changed the processes of musical creation ... I hope that in this course I can adopt an appropriate perspective and learn to adapt each new resource to my classroom."

Even teachers from less traditionally tech-focused subjects are joining the movement. Jesus, a physical education and youth football coach, is excited about AI's potential to enhance sport training and classroom learning:

"I have always been interested in the application of technology in the educational and sports fields. I believe I can bring a practical perspective on how AI can be applied in physical education and sports development, as well as contribute insights on the responsible use of technology in the classroom."

These educators in Spain are part of a growing global network supported by Experience AI, showing that the future of AI literacy doesn't belong to computer science teachers alone but to all teachers, in all subjects, who are preparing students to navigate and shape a world increasingly powered by AI.

AI is for everyone

The mission of EdCamp Ukraine (**edcamp.ua**) is to unite educators and help them grow. Operating from their main base in Kharkiv, near the Eastern border and the frontline of the ongoing war in Ukraine, they see AI both as a tool for new technological breakthroughs and as something that can help build a fairer, more efficient and resilient society.

Educators at EdCamp Ukraine's Experience AI training event

"We firmly believe AI should not only be an object of study — it must become a tool for amplifying human potential. AI should also not be a privilege, but a resource for everyone. We believe the Experience AI [programme] can truly transform education from the bottom up."

Within their community of 50,000 teachers, EdCamp Ukraine ensures that every educator, regardless of their living conditions or where they work, can access high-quality, relevant, and accessible support. With the ongoing situation in Ukraine, this means being flexible with their planning, preparing for a range of different outcomes, and being ready to pivot their delivery to different locations or to an online setting when needed. These same considerations apply to their teacher community, who need to be ready to adapt their lessons for any scenario. EdCamp Ukraine says, "Recognising these war-related challenges helps us see the bigger picture and always have contingency plans in place. We think ahead and develop flexible scenarios."

This year, the team piloted Experience AI through their community of trainers who, when they're not training, are busy teaching in the classroom. One teacher, Yuliia, shared how her students valued the opportunity to be creators, rather than just users of technology: "One student, who is an active AI user, kept silent during the lesson. I thought he wasn't interested, but during the reflection he shared a lot of positive feedback and expressed his gratitude. Other students said it was important that they weren't just told about AI — they were using it, creating images, and working with apps."

EdCamp Ukraine plans to roll out training for Ukrainian teachers this autumn, reaching 2000 teachers and 40,000 young people by the end of next year.

More countries, more classrooms

In May 2025, two new partners — Mind the Gap (**mindthegap.ng**) and NerdzFactory (**nerdzfactory.org**) — joined our network as the first Experience AI partners in Nigeria, and many more countries and organisations are coming soon. As our partner network continues to grow, we are excited to reach more global communities and give classrooms around the world the chance to experience AI.

You can find out more about Experience AI on our website (**helloworld.cc/exp-ai**). If your organisation is interested in partnering with us to deliver Experience AI, register your interest (**helloworld.cc/ai-partner**) and we will let you know about future opportunities. (HW)

CONTINUE THE CONVERSATION ON OUR PODCAST, WATCH THE EPISODE HERE
helloworld.cc/pod27-ai

KAT LEADBETTER

Kat works as a Global Partnerships Manager for the Experience AI programme, supporting partners around the world to bring AI literacy to their communities. She can also be found singing in a choir, playing piano, or trying to learn Japanese.

GET THE MESSAGE ACROSS

Explore the features of clear messaging in digital media

This is the first lesson in The Computing Curriculum's 'Clear messaging in digital media' unit. Learners are asked to consider the search terms needed to find specific content on the web. They will then search the web to find content they deem 'good', which they will capture and annotate digitally. This lesson is designed to get learners to work between applications and use a variety of applications and tools. (HW)

AGE RANGE

11–12 years

OBJECTIVES

- Choose search terms relating to a particular issue
- Use tools to copy an image into another application
- Identify key features of a good poster

STARTER ACTIVITY: GETTING THE MESSAGE ACROSS

10 minutes

Ask learners to discuss with a partner the road sign shown in **Figure 1**. They need to write down the message that the sign conveys, giving as much detail as possible. Once they have done that, they should list places where other public messaging is displayed. Learners should note down the location and the type of messaging, such as bus stop, adverts, bus timetables, etc.

Gather ideas from learners for both points. Encourage a discussion about the importance of clear messaging for different audiences in different locations, such as road signs for drivers of moving vehicles, compared with posters displayed to passengers waiting at bus stops.

Figure 1

The following lesson plan is taken from the 'Clear messaging in digital media' unit of The Computing Curriculum (TCC), written by the Raspberry Pi Foundation. It is aimed at learners aged 11 to 12, and builds on the experiences learners will have had in primary school with using search terms to find content on the web and work between a variety of applications and tools.

ABOUT THE COMPUTING CURRICULUM

Raspberry Pi Foundation

The Computing Curriculum is the Raspberry Pi Foundation's bank of free lesson plans and other resources that offer educators everything they need to teach learners aged five to sixteen. It covers the full breadth of computing, including computing systems, programming, creating media, data and information, and societal impacts of digital technology.

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

Introduce the learners to the lesson objectives and then display **Figure 2**. Explain that posters, like road signs, use pictures and text to communicate a message. Learners will first focus on what makes a good poster, before using those features to make their own poster.

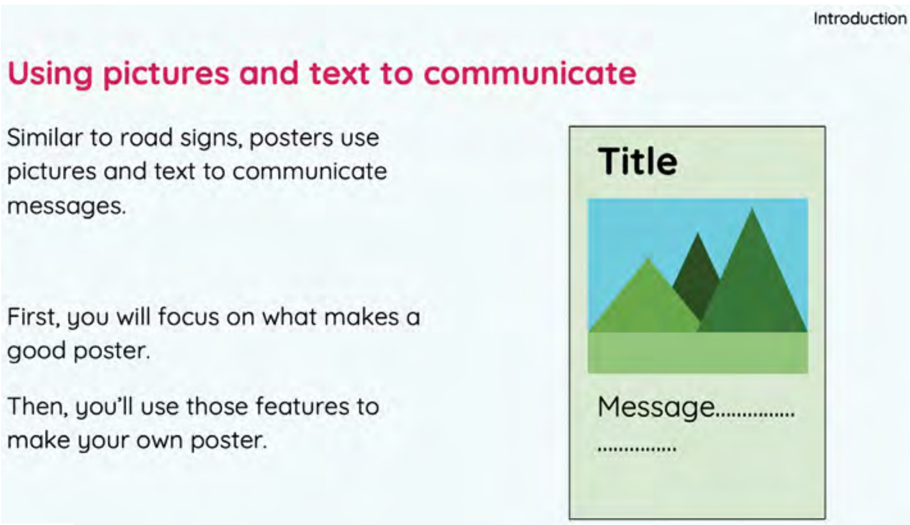


Figure 2

ACTIVITY 1: ENVIRONMENTAL ISSUES 10 minutes

Explain to learners that scientists tell us we need to do more to protect our planet, and that there are lots of charities working to protect and improve the air, land, and sea.

Outline to learners that they are going to search for posters made by environmental charities. Give them a few minutes to write down the keywords they would use to find these posters.

Now ask learners to work with a partner and review both of their lists of words. Between them, they need to choose their top three words and agree on why they have picked those three words.

After learners have had a few minutes to agree on their terms and reasoning, ask them to volunteer suggestions for search terms and note them down to create a class list. You may wish to selectively ask learners why they picked their terms.

ACTIVITY 2: WEB SEARCH

10 minutes

Tell learners that they are now going to use the search terms they have been considering. They need to search the web to find 'good' posters, meaning the poster clearly gets its message across. Once learners have found some posters, they need to choose one they think is good and take a screenshot of it.

Before learners do this, demonstrate how to complete an image search for a poster, and then how to take a screenshot and save or paste it into an application, so they can annotate it in the next activity.

Allow learners time to search for and screenshot a poster that they feel is 'good'.

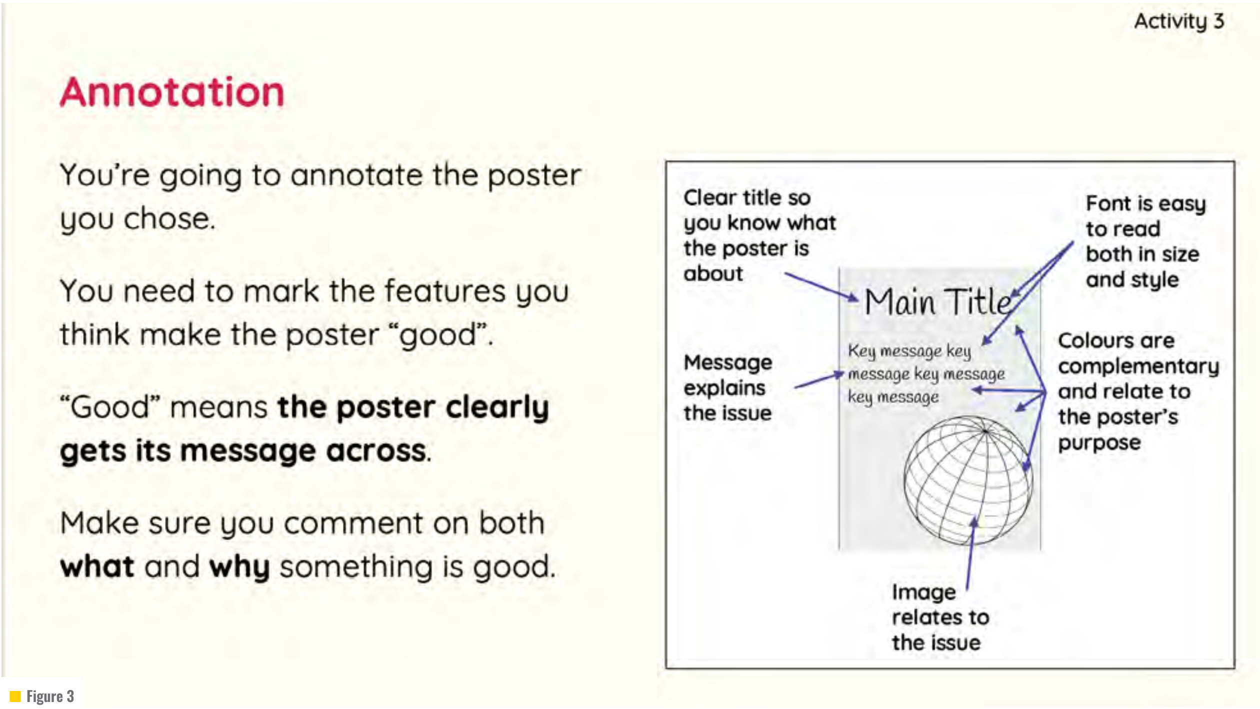


Figure 3

ACTIVITY 3: ANNOTATION

10 minutes

Show **Figure 3** and explain to learners that they are now going to annotate the poster they chose in the previous activity. You will need to show learners how to add arrows and comment on their image. Explain to learners that they need to comment on what they think is good and why they think it is good — remind them that 'good' means that the poster gets its message across clearly. Allow learners time to annotate their screenshots. Learners should save their work.

PLENARY: FEATURES OF 'GOOD'

10 minutes

Ask learners to share and discuss their annotated poster with a partner. While they are sharing, learners should write down key features that they believe make a poster 'good'.

After a few minutes, ask learners to share their ideas with the class. Write down a class list of features that make a 'good' poster. This list will be used in the next lesson of the TCC unit to guide students in making their own good poster. Among other things, the class list should include comments on colours, font size, style, and suitable images.

Briefly recap the lesson. Explain that in the next lesson, learners will make their own posters guided by the class's list of 'good' features.

RELEVANT LINKS

TCC 'Get the message across' lesson:
helloworld.cc/get-the-message-across

FEATURE

USING GENERATIVE ARTIFICIAL INTELLIGENCE TO PLAN AND RESOURCE PRIMARY SCIENCE

Sam Lovatt and **Alex Sinclair** discuss the findings of a small-scale research project that explored using ChatGPT to provide lesson ideas for the teaching and learning of primary science

Generative artificial intelligence (GenAI) has the potential to change the education landscape (helloworld.cc/parliament-research). In their policy, England's Department for Education (helloworld.cc/dfe-genai) suggests that teachers should be using GenAI to reduce their workload and use the time that has been saved to focus on teaching engaging lessons. However, to use a tool such as GenAI, a teacher needs to know how to use it, and how to use it effectively. Findings from recent research at St Mary's University (helloworld.cc/stmarys-genai-report) suggest that teachers want to use GenAI to support their planning and teaching, but do not know how.

One of the most frequent teacher uses for GenAI tools is to support the planning and resourcing of lessons (**helloworld.cc/call-for-evidence**). The Education Endowment Foundation goes as far as suggesting that effective use of GenAI can reduce teachers' planning time by up to 30 percent (**helloworld.cc/baxter**). These positives are not always recognised by teachers, especially if they have not had support or engaged with training on how best to use the tools, or have used them in a limited capacity and concluded that they are not useful (**helloworld.cc/stmarys-genai-report**).

Primary science is practical in nature and presents unique planning and resourcing challenges. Kehoe (**helloworld.cc/kehoe**) outlines some distinct benefits of using GenAI to support lesson planning: personalised learning, creativity-enhancing lesson ideas, and saving time. Recent reports on improving primary science practice cite the need to reduce teacher talk, plan for better investigations, and raise the importance of vocabulary (**helloworld.**

Keeping humans in the loop is paramount

Teachers want to use GenAI but don't know how

cc/bianchi). The effective use of GenAI tools could help meet these priorities, while providing teachers with alternative views and creative ideas. Examples of how to prompt the tools to yield useful solutions to these priorities are given below.

We gave teachers the opportunity to attend a one-off, virtual, science-specific training session in which they were guided on how these tools could be used, with an opportunity to practise using GenAI, as well as offering best practice advice.

How did we structure our session?

The St Mary's University study (**helloworld.cc/stmarys-genai-report**) more specifically showed that most teachers:

- Are unsure how these AI tools can help them
- See the tool as an addition to their practice
- Want further support with how these tools can be used by them and their students

Therefore, for the online support session we focused on providing concrete primary science-specific examples, followed by an opportunity for the teachers to explore how they could be used when planning a lesson. We used ChatGPT for the session, because the platform is easy to sign up for and is already commonly used.

The examples we provided covered six broad areas:

- Generating a list of jobs that children could do in the future based on a given science topic
- Generating a multiple-choice quiz to recap prior learning (including identifying misconceptions)
- Generating a story to support learning about a famous scientist
- Highlighting the necessary subject knowledge required before teaching a new topic
- Generating data from fictitious experiments for children to analyse and interpret
- Providing a starting point for planning and generating resources

Guidance for doing this was provided, and included the following 'rules' for engineering the prompt:

- **Give it a role**: tell the chatbot to adopt a role, such as a subject expert or a student
- **Task:** clearly define the task or goal for the prompt
- **Steps:** Outline the specific steps that the chatbot should take to complete the task
- **Context:** provide context for the task, such as the reading age of the audience
- **Goal:** specify the desired outcome for the chatbot prompt

An example of the prompt provided for generating a multiple-choice quiz can be found in **Figure 1**. For examples from all six areas, and the threads produced by ChatGPT, see the 'Useful links' box.

Following this, we asked teachers to explore how ChatGPT could be used for planning lessons based on a specific national curriculum objective. We provided prompts (like the one below) for planning a lesson about plants in Key Stage 1 (ages 5–7), lower Key Stage 2 (ages 7–9), and upper Key Stage 2 (ages 9–11):

"I am a Year 2 teacher. Plan a lesson idea

USEFUL LINKS

For practical advice on the uses of AI in education: **ai-in-education.co.uk**

Example threads produced by ChatGPT: **helloworld.cc/stmarys-chatgpt** to meet the curriculum point 'observe and describe how seeds and bulbs grow into mature plants.'"

To focus the teachers' reflections, we also suggested that they consider the output in relation to prior learning, age-appropriateness, science capital, skill development, health and safety, and engagement. The teachers were then encouraged to provide a further prompt, to refine or improve the output produced.

What did we find?

Most groups agreed that their initial searches had produced a "serviceable lesson". Three of the groups had focused on creating a lesson for Year 3 (ages 7–8) children, and all noted that there was too much content. The other two groups generated lessons for Year 6 (ages 10–11) children. Interestingly, one of these commented that the suggested ideas went well beyond aspects of the curriculum for that age group, while the other believed that the initial prompt did not provide the necessary detail for Year 6 children. As the prompt was so generic, it was understandable that most groups highlighted that there was little or no reference to the skills of working scientifically, health and safety, how to adapt the lesson, or science capital. As a consequence, groups gave further prompts.

To address the issue of too much content, one group asked, "Can you change this so that the focus is just on pollination?", which produced a "manageable lesson". Another wrote, "That's quite a lot of content for one lesson. Please can you split this into four separate lessons." This provided a choice of two sets of lessons, one of which was deemed appropriate. Specifically requesting, "Can you include a working scientifically objective from the English national curriculum, lower Key Stage 2" resulted in it choosing an appropriate objective. Because of time constraints, groups agreed that prompts about incorporating science capital and how to adapt the lesson for specific learners would have been useful.

The conclusions from carrying out this process will come as no surprise. Groups summarised that using GenAI can be a real time-saver, and that it can find and suggest resources that otherwise may have been missed or not known about. Specificity and clear input are paramount. As with all planning, including the use of schemes of work, the key is that the human in the loop is really important. Teachers need to have sufficient knowledge about science pedagogy and the curriculum to appreciate what is of value: you need to know enough to know whether it's good enough.

© Seventyfour/stock.adobe.com

Teachers need to have sufficient knowledge to appreciate what is of value

> IF PROMPTED CORRECTLY, THE GENAI TOOLS CAN SUGGEST AGE-APPROPRIATE AND ENGAGING LESSON IDEAS

Multiple-choice quiz	'I am a primary-school teacher in Year 5. We have just taught the children the following objectives in science 'describe the movement of the Moon relative to the Earth'. We discussed the phases of the moon in this lesson. Produce a five-question multiple-choice quiz that includes misconceptions. Ensure each question has four potential answers to choose from. Please provide answers with explanations.'	**Example thread from ChatGPT: helloworld.cc/chatgpt-y5-quiz** **Be mindful of:** ■ The quiz is about knowledge you have taught; you can be specific about what to include ■ Ask it to include specific misconceptions that came up in your class when teaching ■ Ensure the answers are accurate

Figure 1 Prompt for generating a multiple-choice quiz on ChatGPT

Suggestions for best practice

There are some cautions when using GenAI tools to plan and resource lessons. As noted, the role of expert knowledge should be considered when teachers are asking the tools to produce and resource lessons.

GenAI tools can 'hallucinate' (falsify) information, as lack of subject knowledge in non-expert users could lead to false information or poor lesson ideas being included (**helloworld.cc/imundo**). A better understanding of how to use the tools can counter this to some extent, but the teacher's professional judgement is of paramount importance.

Findings from this small-scale project mirrored some of those found in the literature, especially around the need for subject knowledge (**helloworld.cc/imundo**) and that teachers should view GenAI as another tool to support good practice. It also highlights that tools like ChatGPT can and should be used by teachers to support them in preparing their primary science lessons. If prompted correctly, the GenAI tools can suggest age-appropriate and engaging lesson ideas. The tools can also be used in more creative ways, such as producing multiple-choice quizzes and lists of future jobs relating to the learning, to help engage children with their primary science learning.

If you want to start using GenAI but are unsure where to start, here is a checklist for creating lesson ideas and resources in primary science:

- **Information is key:** ensure your prompt is well structured and you are specific about what you want created
- **Question, question, question:** the first response from the tool will probably not be the best; ask further questions to refine the tool's answer
- **Make connections:** you are the human in the loop, with the subject and pedagogical knowledge; ensure the content suggested is appropriate for your class
- **Think beyond lesson content:** as you can see from the list above, these tools can be used for more than lesson ideas; consider the more creative aspects of lesson planning that can come from using these tools
- **This is just another planning tool:** it should not replace your subject knowledge, or the knowledge you have of your class

Conclusion

GenAI can help to reduce the planning and resourcing workload for teachers; however, it is important that these tools do not become the only source of information and ideas for your lessons. Use them critically to provide you with creative ideas and alternative views. For further information about using GenAI in the classroom, see **helloworld.cc/dfe-genai** and **ai-in-education.co.uk**, and look out for further training and information from the Association for Science Education (**ase.org.uk**). (HW)

*This article first appeared in Primary Science, Issue 185 (**helloworld.cc/ase-primary-science**).*

FURTHER READING

Education Endowment Foundation (2024, December 12) *Promising findings from first major trial of teachers' use of ChatGPT in schools in England* [Press release]. **(helloworld.cc/baxter)**

Bianchi, L., Whittaker, C., & Poole, A. (2021) *The 10 key issues with children's learning in primary science in England.* The University of Manchester & The Ogden Trust. **(helloworld.cc/bianchi)**

Department for Education. (2023). *Generative artificial intelligence in education call for evidence.* **(helloworld.cc/call-for-evidence)**

Department for Education. (2025). *Policy paper: Generative artificial intelligence (AI) in education.* **(helloworld.cc/dfe-genai)**

Education Endowment Foundation (2023) *Improving Primary Science Guidance Report.* **(helloworld.cc/primary-science)**

Felix, J. & Webb, L. (2024) *Use of artificial intelligence in education delivery and assessment.* UK Parliament POST. **(helloworld.cc/parliament-research)**

Imundo, M. N., Watanabe, M., Potter, A. H., Gong, J., Arner, T. & McNamara, D. S. (2024). Expert thinking with generative chatbots. *Journal of Applied Research in Memory and Cognition. 13*(4), 465–484. **(helloworld.cc/imundo)**

Kehoe, F. (2023). Leveraging generative AI tools for enhanced lesson planning in initial teacher education at post primary. *Irish Journal of Technology Enhanced Learning, 7*(2), 172–182. **(helloworld.cc/kehoe)**

Lovatt, S., Martin, C., Bhorkar, S., Veale, V., Murray, J., Upfield, H. & Price, M. (2024) *Perceptions of Generative Artificial Intelligence from the classroom.* St Mary's University. **(helloworld.cc/stmarys-genai-report)**

ALEX SINCLAIR

Alex is associate professor and primary science education lead at St Mary's University, Twickenham, London.

SAM LOVATT

Sam is a lecturer in primary education (initial teacher education), computing, at St Mary's University, Twickenham, London.

SMARTER IRRIGATION FOR A THIRSTY WORLD

Graham Hastings shares a classroom-ready, project-based learning activity for building an automated irrigation system

We hear a great deal about project-based learning and STE(A)M education. These two approaches are designed to equip pupils with skills they will need in the real world. Rather than passively absorbing information, learners tackle meaningful, authentic problems that cut across subject boundaries. The result? Greater engagement, deeper understanding, and longer-lasting learning.

With this in mind, I adopted a project-based approach when introducing sensing and control in computing. I wanted my pupils to see physical computing as more than just abstract logic. I wanted them to see it as a tool for solving real-world challenges.

CURRICULUM LINKS

Computing
Coding for sensing and control, robotics

Design and technology
Designing and making the sensor and pump

Engineering
Servo, lever mechanism

Geography
Water scarcity, recycling, frugal innovation

Maths
Calibration, < operator, angles, trial and improvement

Science
Plant growth, circuit electricity, experimentation

The problem: smarter irrigation for a thirsty world

In geography, my pupils had been learning about global water scarcity and its impact on food production. They quickly identified a key issue: crops need regular irrigation when rainfall is low, but in many parts of the world, farmers lack the time, energy, or financial resources to manage this irrigation manually.

Furthermore, irrigating when the soil is already moist wastes precious water. What's needed is an automated system: a low-cost, energy-efficient setup that waters crops only when the soil is becoming too dry.

Figure 1 Soil moisture sensor

A computational thinking approach

We broke the problem down into three simpler problems:

1. Sensing soil moisture
2. Building a water pump
3. Processing data and making decisions

Solving problem 1: moisture detection

The class began by researching how soil moisture can be measured electronically. They discovered that wet soil conducts electricity better than dry soil — a useful property for building a sensor.

We used two 10 cm iron nails as electrodes, supported in parallel by rolled plastic from a milk carton (**Figure 1**). The pupils experimented with different distances between the nails, using a micro:bit to read analogue values from soils of varying moisture. After trials, they determined that

Figure 2 Calibration readings

MAKECODE TUTORIALS

The MakeCode website has a wealth of outstanding tutorials for science and STE(A)M projects. This tutorial is particularly relevant to this project: **helloworld.cc/microbit-soil**.

a 2 cm gap gave the most useful range of readings.

The final experiment was to calibrate the sensor using a very dry soil and a wet soil. The calibration readings are shown in **Figure 2**. The class decided that if the reading was less than 600, the soil was too dry and the pump should be turned on.

Wiring the system

Set the project up as follows:

- **Moisture sensor:** connected between pin 1 and 3V on a micro:bit edge strip
- **Servo (pump control)**
- **VCC:** positive terminal of 6V battery pack
- **GND:** negative terminal of 6V battery pack and GND on the micro:bit
- **OUT:** pin 0 on the micro:bit

Note: a micro:bit's 3V output is insufficient to power a servo, so we added a separate 6V battery pack (**Figure 3**). The micro:bit will still require its independent 3V supply.

Solving problem 2: pump design

The pump needed to lift water from a reservoir and distribute it over the field. There are a number of ways of doing this, but they will all require an actuator of some kind to produce movement, typically a motor or a servo. Given the emphasis on frugal innovation, pupils opted for a tipping mechanism operated by a servo, a simple but effective solution.

They cut a lever from a milk carton handle and attached it to a servo arm (**Figure 4**). The servo was glued to an old food container next to a notch cut to allow the pump's arm to move.

On the down stroke, the scoop is immersed and filled with water. On the up stroke, the water runs through the handle and is spread over the field. The pump's lever has to tilt through the correct angle to move the water from reservoir to field. The servo angle settings were calibrated

>

CROC CLIPS PRACTICAL TIPS

- Hand out croc clip leads in bunches of five — red, black, yellow, green, and white
- Cable-tie the bunches together so they are easier to hand out and collect
- Glue the clip's insulator to the wire, which stops it sliding off the clip
- Connect the clips at 90°, one jaw through the hole, to stop it sliding off the micro:bit

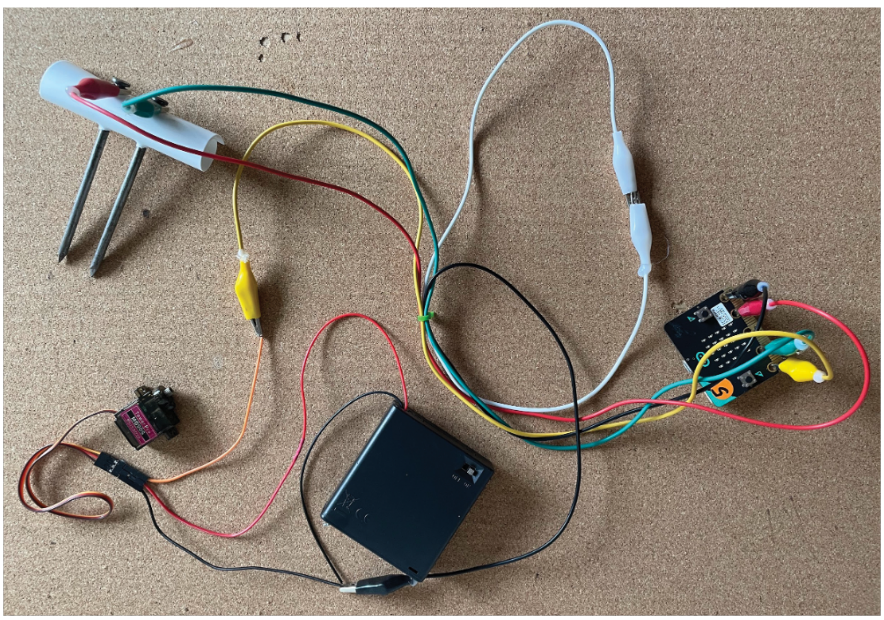


Figure 3 The complete circuit showing how the moisture sensor and the servo are connected to the micro:bit and 6V battery pack

Figure 4 A milk carton was cut for the lever scoop

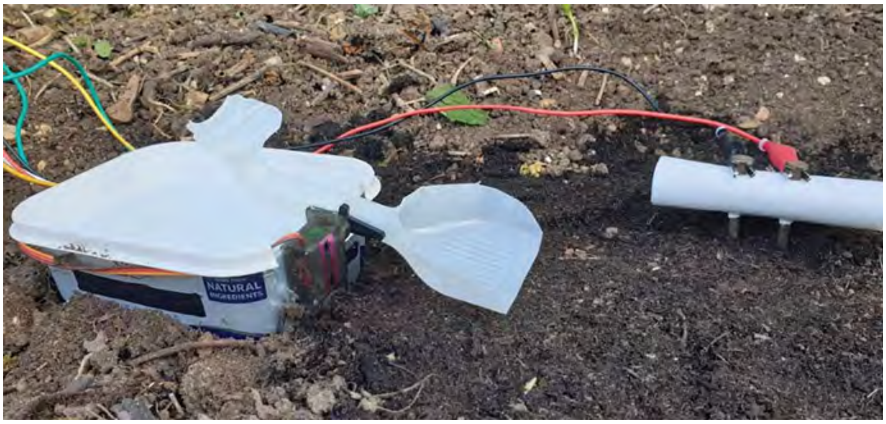


Figure 5 The system in situ: reservoir and water pump on the left and soil moisture sensor on the right

SERVO TESTER

When working with servos, it is well worth investing a little money in a CCPM servo consistency tester.

As well as testing for a faulty servo, it can be used to trial the operation of mechanical systems.

by experimentation using the micro:bit simulator, and then tested on the pump with some alterations to improve the performance (**Figure 5**).

Solving problem 3: coding the system

The class chose to use a micro:bit as the processor because it is cheap, has low energy needs, and the learners were familiar with the online MakeCode editor. They devised an algorithm using a flow chart to plan their program to control the automated irrigation system (**Figure 6**).

A common, simplified approach when teaching sensing and control involves connecting a sensor and a servo to a micro:bit, then writing a script that adjusts the servo based on the sensor's input. While technically correct, this method often limits learning to surface-level understanding and can be rather dull.

By contrast, giving learners an open-ended, real-world problem invites them to engage more deeply with the entire design cycle, from planning and making to testing, evaluating, and refining. It challenges them to think critically, solve problems creatively, and collaborate effectively.

Most importantly, a project like this provides a rich opportunity to connect multiple areas of the curriculum, in this case, computing, science, geography, engineering, and design, in an exciting, meaningful, and engaging way. Instead of learning in isolation, pupils experience how knowledge and skills combine to solve genuine global challenges.

This is the true power of project-based learning: it equips pupils not just with technical skills, but with the mindset to apply them where it really matters. And they had great fun solving a real-world problem that they could all relate to. (HW)

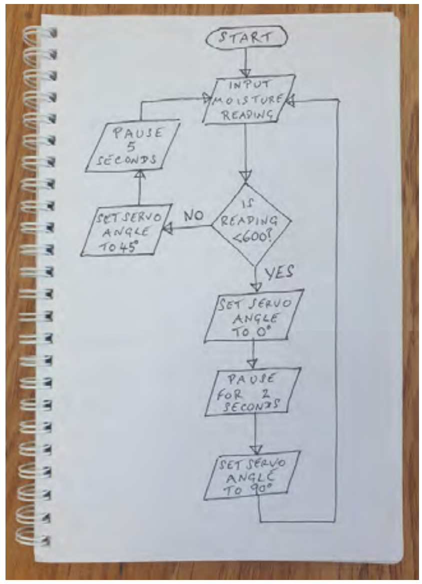
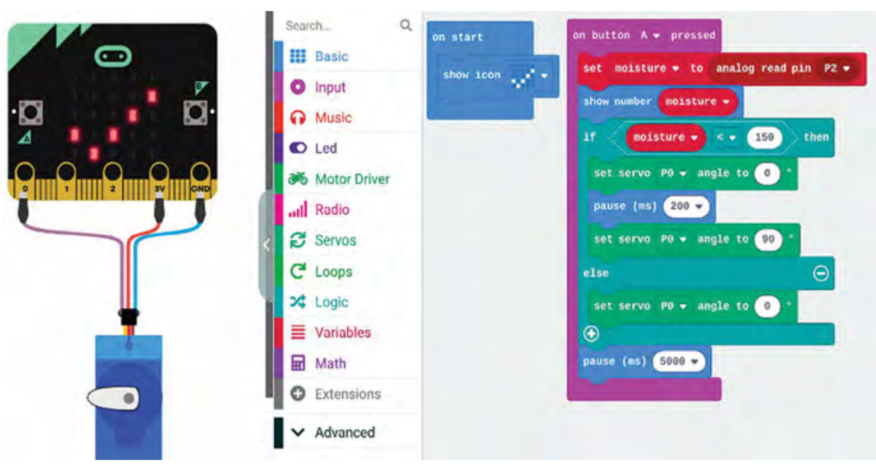


Figure 6 The control algorithm



The servo simulator and control system's program script (note the servo extension blocks)

GRAHAM HASTINGS

Graham is head of computing at St John's College School, Cambridge, UK.

AI BEYOND THE CURRICULUM: INVOLVING PARENTS, TRUSTEES, AND THE WHOLE SCHOOL TEAM

Teresa Harris Boag explores how schools can engage parents, trustees, and school staff in developing an effective AI strategy and culture

As schools begin to engage with the opportunities and challenges posed by artificial intelligence, much of the attention has understandably focused on the curriculum. How can we teach about AI? How does it fit into subjects like computing, or design and technology (D&T) and personal, social, and health education (PSHE)? What tools might enhance classroom practice? These are all important questions. But as someone who worked in IT before moving into education, and who now combines teaching computer science with a senior leadership role which includes safeguarding responsibilities, I'm convinced that real progress lies not just in what we teach about AI but in how we prepare our entire school community to navigate it confidently and ethically. This curriculum focus, while vital, is not enough.

In particular, if we want to move beyond the media headlines and create a thoughtful AI strategy, we must actively involve parents, trustees, and both teaching and non-teaching staff. AI is not just another tool; it presents a shift in how we interact with information, how we make decisions, and how we safeguard our young people. We must respond in kind, not with fear, but with clarity and collaboration.

To prepare our young people for a future (and present!) shaped by artificial intelligence, we need a whole-school, whole-community strategy that reaches beyond the classroom. That means engaging not just students and subject teachers but also parents, trustees, teaching assistants, office staff, and site teams — everyone who plays a part in school life.

Why this matters

There are three main reasons why this matters.

1. Media hype vs informed understanding

Much of the public conversation about AI is polarised: it is either utopian or dystopian. In schools, we need to cut through that noise. If the adults in a school community are going to support students effectively, whether that's through governance decisions, safeguarding conversations, or just helping with homework, they need access to calm, clear, and credible information about what AI is and isn't.

Running informal information sessions for parents, or inviting trustees to CPD briefings, helps bring them into the conversation. Even a simple newsletter section can go a long way towards demystifying terms like 'machine learning' or 'generative AI'.

2. Shared ethical ground

AI raises big questions about ethics, equity, and bias. These aren't just academic issues; they touch on safeguarding, recruitment, accessibility, and the way we model digital citizenship to students.

A whole-school approach allows everyone to share the responsibility for those conversations. When trustees understand algorithmic bias, they're better equipped to scrutinise procurement decisions. When office staff understand data protection in the context of AI tools, they can handle personal data more confidently. When parents know how AI might influence social media feeds, they're more prepared to talk to their children about online behaviour.

3. Empowering confident, values-led use

Perhaps most importantly, a joined-up strategy allows schools to model thoughtful, values-led AI use. That starts with teachers feeling confident in using AI to support their own workload, but it extends to

Partnering with parents is crucial for supporting students in their understanding of AI

other staff too, using AI for timetabling, improving communications, or streamlining administrative processes.

Many schools are being bombarded with the vast array of new software packages coming to market and claiming measurable impact in monetary terms. It's important to recognise that the whole industry is in a period of rapid growth. You can calmly appreciate that the big school management information system (MIS) you've already invested in will likely come good with new AI functionalities as it becomes clearer what schools actually need versus the dazzling but perhaps impractical things AI can do.

And let's ensure that the right people are flagged in discussions about AI too. Yes, AI can do your timetabling, but that staff member who's been doing timetabling for years has a wealth of experience that AI can't replicate. That staff member, along with your heads of years, knows that putting one class on the same narrow corridor as another class at the same time is likely to cause some impact on behaviour management staff. So while AI is great, saves time, and, it has to be said, is sometimes better at achieving what's needed than we busy school staff can achieve in the time we have, the human element must be the decision maker. It's important for senior leaders and trustees to recognise this.

When everyone feels empowered to explore and question AI, the school becomes a space where AI isn't feared or fetishised. Instead, it's understood and harnessed for the good of the students. That's the goal.

Why trustees matter

Trustees play a vital role in guiding the long-term vision and accountability structures of a school. AI touches on many areas within their remit, from strategic planning and procurement to safeguarding and data protection.

What can you do? Consider inviting trustees to staff development sessions exploring AI's potential uses across teaching, data analysis, and well-being monitoring. This will open up rich conversations about workload, ethical use, and infrastructure planning. For example, you can showcase, in a whole-school forum, how it could help with safeguarding alerts. It's not science fiction; it's a practical leadership tool.

Beyond admin efficiency, AI also poses serious questions about student safety. Increasingly, our most vulnerable young people are turning to AI chatbots for support, companionship, and guidance. While there is some benefit in accessibility, these tools are not designed for emotional well-being, and often lack the boundaries, nuance, and accountability required to support a child in distress. Trustees must understand this emerging landscape to shape responsible safeguarding policies. And once we touch on this, we need to bring parents into the approach.

Parents as partners

Parents are another crucial piece of the puzzle, and unfortunately, many are absorbing unhelpful narratives. The myth that AI will 'replace thinking' in young people is both overstated and misdirected. I've seen this before. I remember the panic that surrounded the arrival of search engines! Many feared it meant the end of books, libraries, and deep learning. That fear proved unfounded, and so too will much of the current hysteria.

That said, we do need parents to be aware of the genuine concerns, particularly around safeguarding. AI tools can sometimes bypass content filters, generate persuasive misinformation, and feed unrealistic expectations of instant answers.

> REAL PROGRESS LIES NOT JUST IN WHAT WE TEACH ABOUT AI BUT IN HOW WE PREPARE OUR SCHOOL COMMUNITY TO NAVIGATE IT

What can you do? Consider including a 'Demystifying AI' column in your school newsletter, helping parents understand what tools their children may be using and what questions to ask at home. As teachers, it's useful to work with parents if our students are relying heavily on AI for homework. Parents might worry that AI is making their child lazy. However, you might be able to look more closely and see that the use of AI can be an indicator that a student is using it because they don't understand their homework or are too embarrassed to ask for help, perhaps masking deeper struggles in learning.

Whole-school questions we must tackle

A truly inclusive AI strategy means grappling with complex questions across every layer of school life:

- Where does AI as a learning support end and plagiarism begin?
- What counts as original work in a world of machine-generated text and images?
- Who owns the intellectual property when a student or staff member co-creates with AI?
- Where are the General Data Protection Regulation (GDPR) risks?
- Do we have the infrastructure and devices needed to even start this journey?

AI can be a wonderful support for learners with special educational needs and disabilities (SEND), levelling the playing field and engaging learners in ways we've not been able to before. An example of this is a SEND learner using voice-activated AI to help draft written work. AI supported the structure, but the ideas were theirs. It opened up a new pathway for success. However, the same tool in another context might blur the lines between support and substitution. These decisions require consistent policy, training, and dialogue. I can hear exams officers and SEND coordinators all around the land declaring, "Usual way of working!" — that these AI tools are swiftly becoming the usual way of working for some students, and should therefore be allowed in the students' exam concessions. That needs looking at, but it's a discussion to be had in conjunction with departments of education and exam boards. (Then we need to look at summative assessment methods, but that is beyond the scope of this article!)

Reframing how we use AI for learning

In many classrooms, I've noticed that students treat AI like a fancier search engine, simply asking it for answers. That's a missed opportunity. With the right questioning techniques, AI can prompt deeper thinking.

For example, if you're up for some experimentation and your policy allows it, encourage students to ask their AI engine of choice:

- Which approach might work best to debug this algorithm?
- What's the most efficient method to solve this problem and why?

This shifts the conversation from answer-hunting to strategic thinking. It's not about getting the answer; it's about learning how to think through the problem better. And yes, as teachers this is our job, but it is also our job to develop independence in our learners and encourage a lifelong passion for learning. That's the kind of enriched learning experience AI can offer, but only when guided well. Perhaps this is the kind of homework task that can bring parents into the fold, and reiterate how AI should or could be used correctly. If AI is used in homework to support independent learning, rather than in class, your IT manager, finance manager, and designated safeguarding lead might also be appreciative!

A calm, values-led approach

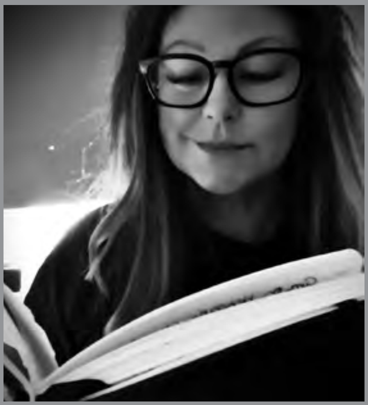
It's easy to feel overwhelmed by the pace of change, but schools don't need to solve everything at once. Instead, we need to lead calmly, collaborate widely, and stay grounded in our values.

AI brings challenges, especially in safeguarding, ethics, and infrastructure, but it also opens up opportunities for creativity, accessibility, and innovation. The key is making sure that everyone, from trustees and teaching assistants to office staff and parents, feels part of the conversation, because only together can we help our students thrive in an AI-shaped world.

Moving forward

If your school is starting to think about AI, I encourage you to begin with a simple question: who needs to feel involved? From there, it's easier to spot the conversations that are missing, and to bring more people into the fold.

AI in education isn't just about what we teach. It's about how we lead, how we collaborate, and how we prepare our entire community for a rapidly changing world. (HW)



TERESA HARRIS BOAG

Teresa is assistant head teacher and computer science teacher at Hillview School for Girls, Tonbridge, UK. She has experience of computing leadership and whole-school strategy, and a passion for inclusion (**linkedin.com/in/teresa-harris-boag-30b4345**).

FEATURE

STREAM MAGIC: ENGAGING STUDENTS ON A BUDGET WITH MICRO:BITS, MAGNETIC TILES, AND MORE

Zahra Amin Shelke shares how embedding STREAM into the curriculum is transforming learning, boosting creativity, and preparing students for a future of innovation

At Arcadia Global School in Dubai, United Arab Emirates, innovation is not an add-on — it is the very heartbeat of our school. As one of the few schools in the UAE to embed STREAM (Science, Technology, Robotics, Engineering, Arts, and Maths) directly into the curriculum from the early years, we are reimagining what classroom learning can be.

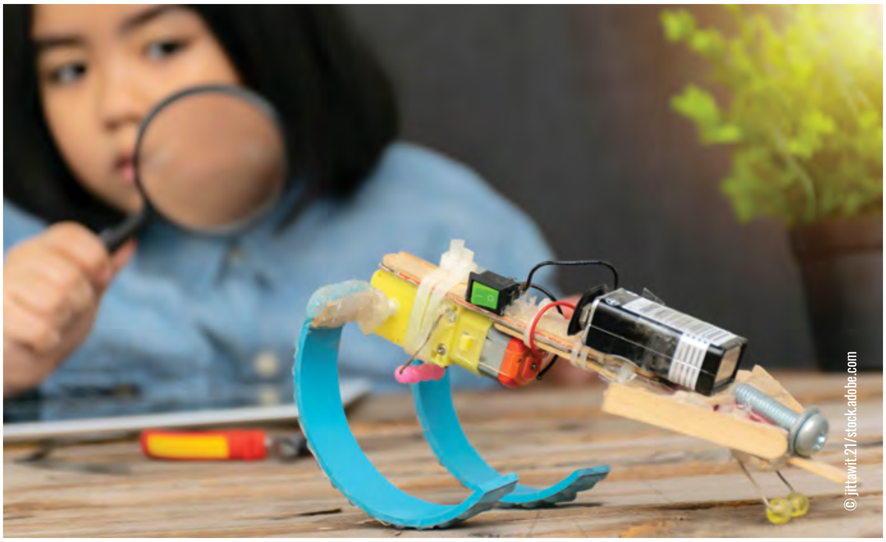
Innovation at the core

STREAM at Arcadia is not an after-school club or a termly initiative — it is fully timetabled and deeply cross-curricular. And the results speak for themselves. Students are thriving, creativity is flourishing, and families are more engaged than ever. Many parents say that STREAM was a deciding factor in choosing our school.

While we do have powerful tools like virtual reality headsets, Lego robotics kits, and 3D printers, this article is about something equally important: the approachable, joyful, low-cost tools that any school can use to ignite STREAM learning — no tech lab required.

This is your invitation to get started with launching that STREAM or STEM club, to embed STREAM within subjects you already teach, and to spark something extraordinary.

You can use magnetic tiles to explore balance, symmetry, and sustainability



Start with what you have to bring STREAM to life in your classroom

Small tools, big thinking

Let's begin with micro:bits. These tiny programmable devices are true game changers. Whether measuring temperature, coding a game, or powering a prototype, they make the abstract tangible. Our students have built pedometers, reaction-time games, and environmental sensors, learning logic, sequencing, and digital skills along the way.

Then there's the surprise superstar of our STREAM programme: magnetic tiles. Used from early years upwards, these colourful tiles support exploration of balance, symmetry, and sustainability. Students build structures and cities, testing ideas and solving problems collaboratively.

FIVE TOOLS TO TRY

- Micro:bits: ideal for coding, electronics, and real-world projects
- Magnetic tiles: build shapes, structures, and spatial awareness
- Straws and connectors: excellent for engineering challenges and design thinking
- Art supplies: bring creativity and visual thinking into every project
- Recycled materials: budget-friendly, sustainable, and brilliant for prototyping

Even humble straws and connectors have a powerful role. They encourage genuine engineering thinking. Pupils plan, prototype, test, and redesign towers, bridges, and even skyscrapers. These simple tools foster resilience, teamwork, and creative problem-solving in ways that sometimes surpass more structured kits.

Five simple rules to get started

So, how can you bring STREAM to life in your school, whether you have a high-tech makerspace or a quiet corner of a classroom? These five guiding principles, which I also shared at the inaugural British Schools in the Middle East (BSME) STEAM Conference in Abu Dhabi, UAE (April 2024), can help:

1. **Start with real-world problems.** Ask students to design for their community, improve sustainability, or rethink a playground. Make learning purposeful.
2. **Foster enquiry-based learning.** Let curiosity drive discovery. Encourage questioning, experimentation, and open-ended exploration.
3. **Integrate across subjects.** STREAM shouldn't sit in a silo. Connect it with history, geography, literacy, or well-being. It belongs everywhere.
4. **Champion collaboration and communication.** Have students work in teams, share their thinking, and present their ideas. The process is as powerful as the product.
5. **Connect to the real world.** STREAM isn't just academic — it's preparation for life, for careers, and for solving global challenges.

Whether it's a child in Key Stage 1 (ages 5–7) discovering magnetic shapes for the first time, or a student in Year 6 (ages 10–11) coding with micro:bits, the impact is clear: STREAM has energised our learners and captivated our families.

Make it happen

The most important message I want to leave you with? You can do this too!

You don't need a robotics lab or a big budget. Start with what you have: magnetic tiles, micro:bits, straws, art supplies, or recycled materials. Integrate hands-on challenges into your timetable every week. Let students take the lead.

You'll be amazed by what they create and how your learning environment transforms. When STREAM is meaningful, practical, and connected, magic happens.

So here's to your own STREAM journey. It doesn't have to be perfect — just purposeful. Begin small, think big, and inspire wonder.

I'd love to know what your journey looks like. I hope I've inspired you to get that STREAM club started. (HW)

ZAHRA AMIN SHELKE

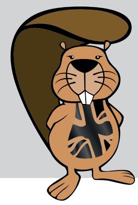
Zahra is a dynamic senior leader and head of innovation at Arcadia Global School, UAE, leading the strategic integration of STREAM, AI, computing, and digital learning. She is passionate about making future-ready education accessible, engaging, and impactful for all learners (**linkedin.com/in/zahra-shelke, @ags_innovationlead_zahrashelke**).

THE BEBRAS PUZZLE PAGE

Each issue, **Andrew Csizmadia** shares a computational thinking problem for your students based on the work produced by the International Bebras Community

ABOUT BEBRAS

Bebras is organised in over 90 countries and aims to get students excited about computing and computational thinking. Last November, 467,000 students participated in the UK annual challenge. Our archived questions let you create your own automarking quizzes at any time during the year. To find out more and to register your school, head to **bebras.uk**.



THE PROBLEM: ART SCRATCH PAPER

DOMAIN

Data, data structures, and representations

SKILLS

Decomposition and evaluation

AGE

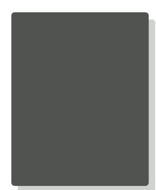
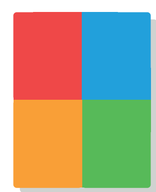
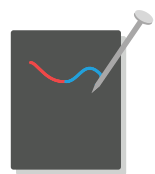
6–12 years

DIFFICULTY RATING

Ages 6–8 medium
Ages 8–12 easy

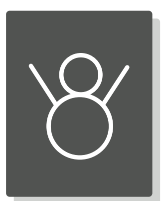
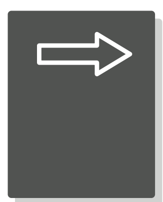
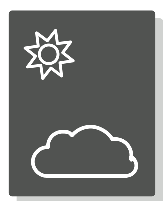
Art scratch paper is special. It is dark grey, but has four colours hidden underneath.

You can draw a pretty picture by scratching it with a sharp object.

		
New art scratch paper	The hidden colours	When scratched, the colour behind appears

Task

If using art scratch paper, which of these drawings will show exactly three colours?

A	B	C	D
			

Explanation

In order to find out the number of colours that appear on the paper by scratching, you must think carefully about where the real colours are located behind the black paper.

When you work with computer software to paint images, you often have to deal with layers that look like sheets of paper in a stack. The images in each layer can mix with each other. Paint applications have even more possibilities. They allow you to change the order of the layers, and they also allow you to take parts of the layer away. This is called transparency.

If you look at any website you will see many things on top of each other: background image, colours, text, and so on. In order to display these correctly, the computer has to know in which order things are put on top of each other. Computers call this the z-index: a number that says how high up in the stack a layer is located.

This puzzle was developed by the Bebras team in South Korea, and reviewed and modified by members of the Bebras community. The solution is on page 81. (HW)

Collaboration is key in coding

MEET THE CODE CLUB

Code Club leader **Kay Wojnarski** and student **Seung Woo (Tony) Han** tell us more about Code Club Collège Jeanne-Sauvé in Winnipeg, Canada

Code Clubs provide young people with an opportunity to explore coding — encouraging creativity, critical thinking, and teamwork. With coding increasingly influencing the future, these clubs are vital in equipping children with the skills, confidence, and opportunities they need to make a difference. We meet a teacher and a student who run their school's Code Club.

Kay Wojnarski
High-school technology teacher

Why did you start a Code Club?
We started a Code Club this year by the request of some of the students in our computer science class, so that they would be able to work on projects in coding languages different to those that we use in class.

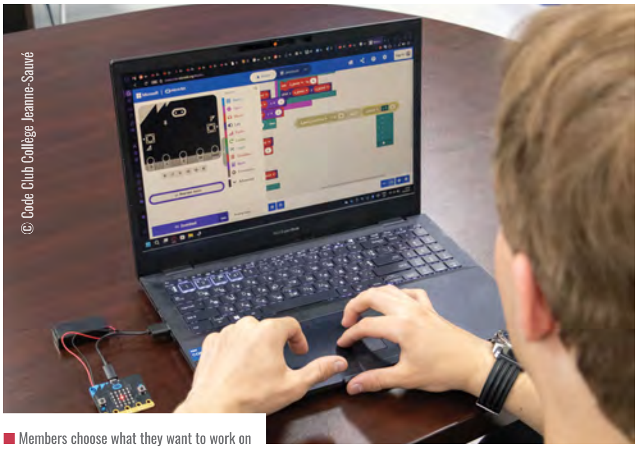
What Code Club projects are creators working on?
micro:bit projects in MakeCode and Python.

What makes your club special?
CJS is a French immersion school with a diverse population that encourages respect and appreciation for others. Our club encourages students to bring in and share their work on personal projects. We are proud of the growth of our technology department at CJS, which has undergone many changes in recent years to meet the learning needs of our students. Our computer science programme teaches students how to code in Unreal Engine while learning the principles of video game design.

What does a Code Club session look like?
We get together once a week at lunch to work on coding projects in the computer science classroom.

Describe your club in three words.
Diverse, ambitious, kind.

KAY WOJNARSKI
Kay (she/they) is a high-school technology teacher at Collège Jeanne-Sauvé (CJS), Winnipeg, Canada, who specialises in media, drafting, and video game design, and can often be found running Pathfinder games at the CJS Tabletop Club. She and her wife Sarah run Sunny House Rescue, a foster home for neglected and abused cats.



Members choose what they want to work on

Seung Woo (Tony) Han (grade 11)

Co-founder of Code Club of Collège Jeanne-Sauvé

Why did you want to start a Code Club?

I had the idea to start a coding club for a very long time. I have always been passionate about computers and all their inner workings. For starters, I wanted to create a space where everyone would be accepted and encouraged to learn more about coding and not be ashamed to 'nerd out' about this subject, like I do very frequently. I've dreamt of creating a space that embraces this beautiful passion.

Like many others, I was independently taught, and during this time of independent learning I'd find many different roadblocks and challenges that I had to overcome alone. A big setback for me was finding the right resources in order to learn how to code. Sure there are many fantastic utilities out there for learning. However, it is very challenging to find these resources without emptying your wallet.

Another big obstacle for me was motivation. I would find myself losing interest on a project. I didn't have the exterior motivation to help me push through the inevitable hardships that come with coding. Now, with the coding club in place, I was able to solve all of the numerous problems I had with coding. Finding resources is easier than ever with a teacher and all of the amazing members. Furthermore, motivation will never be an issue as members share ideas and work together, learn together, and find solutions as a collective. I created the coding club to help everyone on their journey with computer science, no matter their skill level, and that is exactly what it is turning out to be.

What makes your club special?

Collaboration. Our coding club is big on communicating and helping each other. Collaborating and working in a tight-knit team is a key element in the coding process. Without it, everything would be a mess. Therefore, in our club, talking with other members and building friendships is highly encouraged. We take pride in knowing that our members feel safe and connected with others in the room. Independent hard work and dedication are no doubt very important for success, but we believe that teamwork makes those elements a whole lot easier. Besides, what's the point of coding if you're not having fun?

What does a Code Club session look like?

Our coding club is held during the lunch hour every Thursday. Members choose what they want to work on for the day, for example working with their designated micro:bit, and they start coding away! Some sessions have groups of people that work in collaboration to create something together and pick up where they left off last week. In our club, you have the freedom to choose what you'd like to learn and how you want to do it. Of course, we are willing to guide them if they ever feel lost in their subjects at any time. There is so much laughter in the coding room every time — it is so amazing to see people engaging in their passion of coding and having fun.

Describe your club in three words.

The first word would definitely be 'fun'. Coding should be something that is fun — not stressful like others may make it seem. Celebrating achievements, making short-term goals, and problem-solving with friends are all great ways that we make coding fun in our club.

A second word would be 'teamwork'. Without teamwork, our club wouldn't really be a club, it would simply be an ensemble of people coding in their own little cubicles, much like a stereotypical office job. To me, that does not sound too enjoyable. Teamwork is our little secret ingredient in problem-solving and building motivation. We embrace it by creating a safe space where everyone can speak their minds without judgement.

Finally, the third word I would use to describe our coding club would be 'new'. I would choose this word because it is used to describe so many different aspects of our coding club; like how it is fairly new since it was founded in December 2024, how you can and will make so many new friends, how this is a place where you can learn new things no matter your skill level, and finally how you can make so many new memories here in the coding club with many others who are equally as passionate as the next. (HW)

SEUNG WOO HAN

Seung Woo (Tony) is the co-founder of the Code Club at Collège Jeanne-Sauvé. He is a grade 11 student who would like to pursue a career in computer science. In his free time, he loves to code, play games, practise his instruments, and talk with friends. For his whole life, he has been curious about the inner workings of computers and his inspiration for coding is Tony Stark or Iron man from the Marvel Cinematic Universe!

Matti Tedre speaking at the symposium

LEARNINGS FROM THE 'TEACHING ABOUT AI' SYMPOSIUM

Navigating the world of AI without a map

Inspiring Learners is a multi-academy trust in Manchester, and Tyntesfield Primary School is part of that trust as a two-form primary school (ages 4 to 11) in Trafford, England. Children are at the centre of everything we do, and our values reflect that education is about more than just scores and grades.

As the computer science specialist, I am responsible for teaching computer science to children in Years 1 to 6 during the class teacher's planning, preparation, and assessment (PPA) time. For the first half of the year, I work with Years 2, 3, and 5; and after the February half term, I switch to teach Years 1, 4, and 6. This format allows me to focus on a progression of skills and deepening understanding. It gives me a clear picture of knowledge and gaps across the school, so I can tailor my teaching to meet the needs of the pupils.

The computing person

The children have always known me as the computing person — even when I had my own class — and they often assume that I studied computer science at university. However, my background is in psychology. Computer science wasn't offered at GCSE at my high school and so I wasn't even introduced to it until it became a compulsory part of primary education in 2014.

I vividly remember attending a staff meeting led by our computing lead in my previous school and being introduced to Scratch for the first time. I was instantly hooked, and was incredibly pleased with my animation! Since then, I have dedicated time to developing my knowledge and pedagogy by reading blogs, articles, books, attending CPD, and networking on LinkedIn. I'm also undertaking a Level 3 apprenticeship around digital transformation in education, which has provided a range of CPD opportunities.

My AI preconceptions

Outside of teaching, I am also a children's author, and I must admit that AI frightened me at first. The thought of someone being able to produce in 30 seconds something similar to what I pour my heart into was, quite honestly, soul-destroying. Then there was the added ethical question of my work being used to train AI models; it's not hard to see why the author and illustrator world is worried about the implications of AI. However, the more CPD I attended as part of my apprenticeship, the more I saw the potential of AI. And the more I learnt, the more I realised how important it was to teach the children about AI — not just use it. Once you've opened Pandora's box, there is no going back. AI isn't going anywhere, and it will be a vital skill for future jobs. Despite skirting on the edge of AI and drafting an AI policy for our school, I still hadn't tackled teaching the children about AI. I suppose I was waiting for a map or directions to make sure I was doing the right thing, so when I saw the 'Teaching about AI' symposium (**helloworld.cc/26**, pages 6–9) advertised, I knew I had to apply.

Attending the 'Teaching about AI' symposium

Arriving at the symposium, I was a little nervous. Someone on LinkedIn said it was a real 'Avengers Assemble' moment, and I had a flurry of imposter syndrome as everyone introduced themselves. Who was I but a teacher from Manchester? What was I even doing there? I had no reason to worry, though. Everyone was incredibly friendly and we were all there for the same reason — to share ideas and drive teaching forward. One thing that was clear from the start was that no one had a map, and that was strangely reassuring,

> " LEARNING ABOUT THE MECHANISMS BEHIND THE MAGIC TRICKS HAS DEEPENED MY UNDERSTANDING OF AI

The whole day was incredible from start to finish. I had briefly read the UNESCO framework (**helloworld.cc/ai-unesco-framework**), but hearing first-hand how it came about really deepened my understanding. I like that it has such a heavy focus on ethics and the human element — something that is very important and close to my heart. Hearing Matti Tedre speak of how schools in Finland tackle AI was also inspiring, and I was starting to see hints of a way forward. But mostly, the chance to network with other primary educators was the highlight of the day. I learnt so much just by listening and asking questions. The whole day gave me the confidence to go back to school and start the journey, even if I wasn't 100 percent sure of the directions.

Research to practice

On returning to school, I took out everything from the symposium and sat down with a blank piece of paper. I started to mind map everything I wanted the children to learn, mapping this onto the UNESCO framework. Before long, I had some 'I can' statements for each year group, and I started to expand these into lessons. One of the most useful parts of the day was being exposed to resources such as Machine Learning for Kids (**machinelearningforkids.co.uk**) and Quick, Draw! (**quickdraw.withgoogle.com**). I incorporated these into lessons to create rich, meaningful experiences for the children.

So far, I have taught the AI topics in Years 1, 4, and 6. I have been blown away by the discussions, ideas, and enthusiasm from the children. In Year 4 (ages 8–9), we covered bias, how AI models are trained, and the ethics of the data used. You can find the resources for these lessons here: (**helloworld.cc/Walmsley-Y4-lessons**). I think one of my favourite moments was when a child pointed out (in as many words) that AI isn't unethical — it's the people using it, and they should be taught to use it kindly. The children often leave the lessons with more questions than when we started, but that, for me, was a sign that they were starting to think critically about AI.

Looking ahead

I'm looking forward to trialling the Year 2, 3, and 5 lessons next year and building on this year. AI for staff is something that the trust is keen to develop, and I think it's also important that staff understand what AI is and isn't, to better appreciate its capabilities and limitations. From personal experience, learning a little about the mechanisms behind the magic tricks has deepened my critical thinking and use of AI.

I am lucky to work in a forward-thinking school and academy trust that recognises the importance of AI and preparing our pupils for the future — the world our young people will work in will require a whole new set of skills. While we can't predict exactly what those are, we can help create critical thinkers who are capable of evaluation and of using AI ethically. It can be scary to try something new, especially if there is limited guidance. However, finding your people — either in person or on LinkedIn or similar — can give you the confidence to try. Without the symposium, I am not sure I would have been brave enough to give it a go.

Embracing the AI journey

I plan to continue reading, learning, and absorbing all I can about AI in education, to help empower the staff and pupils at my school and trust. I look forward to networking with other educators and sharing ideas and resources. I'd like to thank the Raspberry Pi Foundation for hosting the springboard into this exciting adventure I find myself on. I don't have a complete map, but I'm heading out anyway. (HW)

CLAIRE WALMSLEY

Claire is a computing, EdTech, and online safety specialist based in Manchester, UK, working across KS1 and KS2 (ages 5–11). You can find her on LinkedIn at **linkedin.com/in/claire-walmsley-3889a0249**.

LANGUAGE, ABSTRACTION, AND AI

Michael Conterio and **Tracy Gardner** compare language acquisition in children and large language models

A child's description of a grandma will be influenced by their own personal and sensory experiences

Children will often apply a term too widely, for example calling lots of different older people 'Grandma'. They refine their linguistic model based on new input, whether that's being corrected ("That's not Grandma — remember, we saw her yesterday?") or just hearing different terms being applied to members of that wider group ("Did you enjoy playing with Grandad?"). Later on, we tend to acquire linguistic knowledge by having the meaning described to us and thinking about it logically ("A player is offside when ..." — so is that player offside?). But you can still pick up on the meaning of unfamiliar words based on the context they are used in. Even though the collection of sounds or letters that make up a word usually bears no relation to its meaning (onomatopoeic words being an interesting exception, along with humans across the world linking similar sounds with 'sharp' or 'soft' shapes — see helloworld.cc/sound-shape), the frequency with which that word appears with other, related words can help you to understand it.

Relationships between words

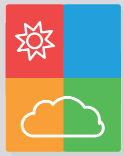
So how does this link to how large language models (LLMs) 'learn' language? LLMs also associate words with the contexts they are used in. To get some insight into how this works, try the online game Semantle (**semantle.com**), which is based on the idea of word similarity. You make guesses at the word of the day and it tells you how similar your guess is to today's word. For example, 'key' and 'lock' are closely related, while 'key' and 'cloud' are not closely related. This is based on a database of similarities between words. Semantle gives us a glimpse into how LLMs work, though they are more sophisticated.

LLMs are so good at identifying relationships between words, including the probability that a word would appear next in a particular context, that they can produce a coherent answer to a human-language question. This is because they have been trained on vast amounts of text and, in doing so, have captured the statistical regularities and arbitrary mappings between words. While humans also have access to sensory input (including internal feelings) and the ability to abstract meaning from our experiences (including being able to share ideas about things we've never experienced, such as subatomic particles), the current generation of LLMs have access only to these statistical associations.

BEBRAS PUZZLE

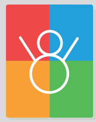
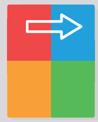
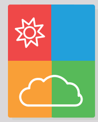
BEBRAS PUZZLE SOLUTION: ART SCRATCH PAPER (PAGE 75)

Solution



Explanation

Each picture touches the following coloured areas:

A	B	C	D
			
4 colours	2 colours	3 colours	4 colours

As you can see, only answer C will show three colours.

COMPUTING KEYWORD SPOTLIGHT: EVALUATION

Defining everyday words and phrases in computer science

Evaluation is the process of ensuring that a solution, whether an algorithm, system, or process, is a good one — that it is fit for purpose. Various properties of solutions need to be evaluated. Are they correct? Are they fast enough? Do they use resources economically? Are they easy for people to use? Do they promote an appropriate experience? Trade-offs need to be made, as there is rarely a single ideal solution for all situations. There is a specific and often extreme focus on attention to detail in evaluation based on computational thinking. Computer interfaces are being continually developed to meet the needs of different users. For example, if a medical device is needed to deliver drugs to a patient automatically, it needs to be programmable in an error-free, quick, simple, and safe way. The solution must ensure that nurses will be able to set the dose easily without making mistakes, and that it won't be frustrating for patients and nurses to use. In the proposed design there would be a trade-off to be made between speed of entering numbers (efficiency) and error avoidance (effectiveness and usability). The design would be judged on the specification proposed by clinicians, regulators, and medical device design experts (criteria) and the general rules relating to good design (heuristics). Criteria, heuristics, and user needs enable judgements to be made systematically and rigorously. (HW)

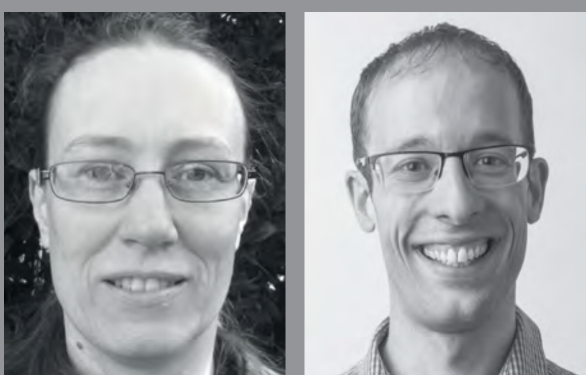
Statistical patterns

Even without an understanding of what a sentence is, or what a discussion is about, LLMs can still use the statistical patterns in the text data to generate responses in conversations. When you ask a child to explain what 'Grandma' means to them, their response will be influenced by their personal and sensory experiences. If you asked an LLM the same question, its response would include some similar words and sentence structures, but would come from a totally different method. While it would be able to use the statistical patterns of sentences in the data it was trained on to construct a grammatical and relevant response, which would draw on what other people had said in response to similar questions, and even provide some specific details, these would not come from its experience. The LLM would not have an understanding of the actual experience of having a grandma.

The big breakthrough for AI

This association-making approach to conversation, along with enough computing power to achieve it, was the big breakthrough that enabled a huge leap in the practical application of AI. Previous attempts at AI had attempted to provide the structure and semantics for the machine, but it turned out to be more practical to allow the machine to extract the connections from huge amounts of examples. Many experts think that we will need to bring semantics, the meaning in language, back into machine learning to make the next major breakthrough (**helloworld.cc/chomsky**).

When discussing GPTs (generative pre-trained transformers) with your students, encourage them to maintain an awareness of how the GPT is using language to communicate in a human-like way, but without the meaning that accompanies human thought and reasoning. If appropriate for the age you teach, you could ask an LLM a question and consider how it came up with its response and how it is different to the approach a human would take. (HW)



TRACY GARDNER & MICHAEL CONTERIO

Tracy is a computer scientist, tech industry professional, technology educator, and a co-founder of Flip Computing. Michael is a former physicist and now works as an online course production manager at the Raspberry Pi Foundation.

BELONGING IN TECH

Connecting culture, creativity, and career in the CS classroom

As a cybersecurity and computer science teacher at a STEM magnet high school, I've made it my mission to ensure every student sees themselves in tech. Representation matters — in who teaches, what we teach, and how.

I design learning experiences that allow my students — particularly Black and Brown students — to explore identity, culture, and social impact through technology. I've learnt that students thrive when they can connect computer science to their lives, values, and aspirations. They code with purpose. They explore CS and cybersecurity not just as a career path, but as a tool for equity and empowerment.

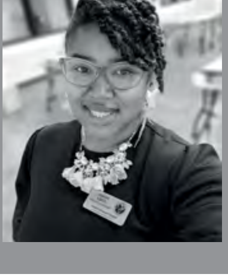
From hardware to empowerment

As someone who transitioned from engineering into education, I know that students don't just need to understand how technology works — they need to experience it with their hands, eyes, and minds. Sometimes it means ditching the traditional lesson plan and diving into real-world IT scenarios that simulate industry tasks and encourage deep exploration.

In my IT hardware classes, students don't just memorise components — they act as technicians, solving mock help-desk tickets, simulating hardware installations, and documenting their work as they go. It's about building fluency and confidence — especially for kinaesthetic learners who shine when they get to touch, tinker, and troubleshoot.

Real-world projects with purpose

I make it a point to align national competitions with course standards and embed them directly into our curriculum. Some examples are:



TIFFANY N. JONES

Tiffany is a high-school computer science and cybersecurity teacher in Georgia, USA. A Department of Defense STEM Ambassador and a former engineer, she specialises in culturally responsive, hands-on STEM learning (**@thesocialblenerd, linkedin.com/in/tjonesatl**).

TRY IT IN YOUR CLASSROOM

Here are three ways to start blending culture, creativity, and career prep into your CS or IT classroom:

- Reframe your projects around identity and impact: ask students, 'How can this technology tell your story or solve a problem in your community?'
- Turn national competitions into class projects: align with your standards and use the prompts as springboards for group work and cross-curricular innovation
- Use hands-on challenges to build tech confidence: try a help desk day where students troubleshoot common hardware/software issues in teams using mock service tickets

- **eCYBERMISSION:** an online virtual STEM competition where students use the engineering design or scientific inquiry process to solve local problems that they identify (**ecybermission.com**)
- **Amazon's Your Voice is Power:** where students remix beats with code using EarSketch, embedding messages of social justice and equity (**helloworld.cc/voice-power**)
- **Samsung Solve for Tomorrow:** where students blend design thinking with innovation to address issues in their community (**helloworld.cc/solve-for-tomorrow**)

These projects give students a platform to tell their stories, solve real problems, and gain recognition beyond the classroom. We've celebrated national finalists and winners, but more importantly, every student walks away with pride in what they have designed.

Seeing yourself in tech

When students see themselves in what they're learning, they rise to meet the challenge. They retain more. They ask better questions. They start to see tech as a space where they belong — not just a space they can access. Whether they go into cybersecurity, engineering, or IT, they leave with a strong foundation of skills, creativity, and confidence. (HW)

(Hello World)

helloworld.cc

Raspberry Pi Foundation