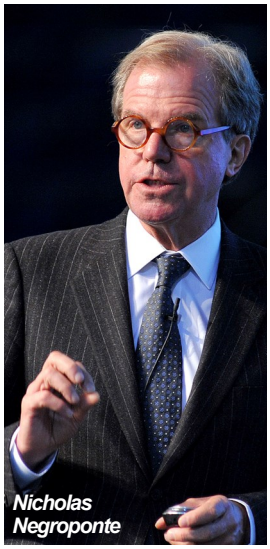
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

SWITCHED ON

COMPUTING AT SCHOOL NEWSLETTER
WINTER / SPRING 2014

LEARNING TO THINK LIKE A COMPUTER SCIENTIST

Our world is changing rapidly, posing challenges for education across the globe. “We are currently preparing students for jobs and technologies that don’t yet exist ... in order to solve problems we don’t even know are problems yet.” said Carl Fisch in his famous video, Shift Happens. The world needs children who are not just comfortable with new technologies, but understand some of the principles on which they work. Children who have the intellectual tools to cope with complexity. Children who understand that the systems that shape their work and leisure are not magic, but grounded in logic. Just as all children are introduced to laws of physics, so they need to grasp the concepts that drive our digital world.


Nicholas Negroponte

Last December, millions took part in an hour of code. That we need our kids to code has become a mantra repeated across continents. But we need more. We need our kids to ‘think’. There is more to Computer Science than coding. We need children to develop the ability to think computationally. We need flexible, independent problem solvers. “Children need to learn learning, which is primarily acquired through the passion that comes from access, the ability to make things, to communicate and to express” says Nicholas Negroponte, architect of the One Laptop Per Child initiative. “Writing a computer program, while seemingly esoteric, is in fact the closest a child can come to thinking about thinking. Likewise, debugging a program is the closest one can come to learning learning.” We need to teach all our kids to begin to think like computer scientists.

INSIDE THIS ISSUE

This issue takes a look at the idea of Computational Thinking. Computational thinking is something children do, not computers. Indeed, many activities that develop computational thought don’t need a computer at all. This influential term helps stress the educational processes we are engaged in. Developing learning and thinking skills lies behind our view that all children need exposure to such ideas.

Many teachers joined CAS last year. We hope there is something here for everyone, whether primary or secondary. The articles are written by people like you, in the hope that you will gain the inspiration and confidence to give it a go!

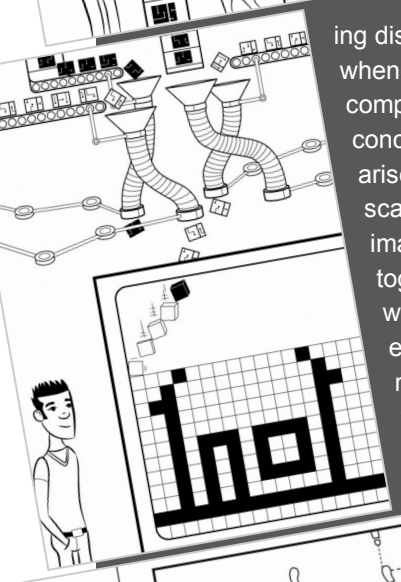
At A Glance p4 - 7
Developing the primary curriculum p8 - 13
Resources for secondary teachers p14 - 19
Explaining the Network Of Excellence p20 - 21
Highlights from trips and conferences p22 - 23
Developments in other countries

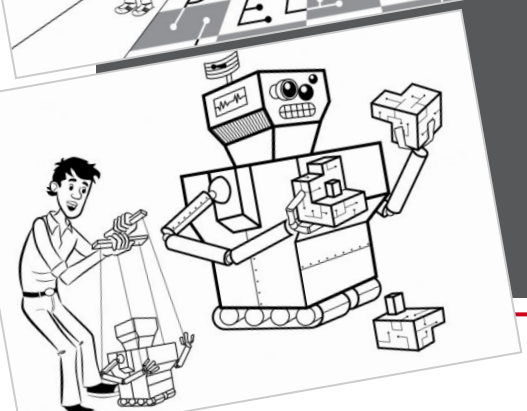
The “Computing At School” group (CAS) is a membership association in partnership with BCS, The Chartered Institute for IT and supported by Microsoft, Google and others. It aims to support and promote the teaching of computing in UK schools.

bcs Academy of Computing

ISSN: 2050 -1277 (online) 2050 -1269 (print)

Cartoons (www.ctillustrated.com) depicting aspects of computational thinking are the result of a project between computing teacher Ben Chun, from the Galileo Academy of Science and Technology, San Francisco and artist Tim Piotrowski. The pictures are ideal for displays and for prompting discussion whenever computational concepts arise. The scaleable images, together with good explanatory text are free to download.





COMPUTATIONAL THINKING: A SPECIAL WAY TO LOOK AT PROBLEMS

The term 'Computational Thinking' has gained widespread currency. Steve Hunt, from the School of Computer Science, University of Hertfordshire looks at what it means and why it is important.

According to Computing: A Curriculum For Schools (March 2012), computational thinking is "...a mode of thought that goes well beyond software and hardware, and that provides a framework within which to reason about systems and problems." But what does this mean in practice, and why is it so important?

First let's look at what it's not. It is not *thinking like* a computer. Computers *can't think*. But even if they could, where would be the benefit in learning to think like one? Computational Thinking (CT) is not *thinking about* computers either. It is not the study of computers or software, though it plays an essential part in the design of both and in the detection and correction of errors. The emphasis of CT is on *understanding how things work*, and why they might go wrong, rather than what they are supposed to do. CT is not *thinking about using* computers either. Computers are ubiquitous, and their uses many and varied, so it's important to have end-user skills. But an understanding of CT and the ability to apply it *informs* users in a way that sets them apart from those who are adrift on the sea of technology.

So what *is* Computational Thinking and why is it useful? It's thinking and reasoning about the world in computational terms. This is an instructive (and different) way of engaging with the world, and it is complementary to other modes of thought and reasoning. A grasp of CT enables people to see the world through a computational lens. Computational thinkers gain an understanding of the world about them that makes them able to model things from a variety of perspectives, each of which brings out some computational aspect of what they see. CT underlies much of computer science, and a grasp of how to 'think computationally' gives a valuable insight into how computers work, and why they behave the way they do. Computational thinkers gain the ability to model problems in a manner that makes them amenable to computational solutions. Where others merely see instructions, actions and things, computational thinkers are able to see algorithms, processes and data.

CT encourages separation of the 'what' from the 'how', and it helps us identify and address problems where the principles of computer science can be applied. It provides thinkers with a unique perspective on the world. Where 'ordinary' folk make little distinction between what needs to be achieved and how it should be done, the computational thinker is aware of the difference between results, and the processes by which results are obtained, and can see how to decompose problems, and how to use computing to solve them.

CT is useful *well beyond* computer science. It gives an insight into what can and cannot be computed, and encourages students to investigate the possibility of devising computational models of systems and problems that might otherwise be viewed as nothing to do with computing. Take an everyday household activity such as washing your hair. When computational thinking is applied to this kind of activity it becomes a process that is carried out by a human 'processor' interacting with other systems to achieve his or her goals. The process has a start state (dirty dry hair, shower

Jeannette Wing was Professor of Computer Science at Carnegie Mellon University when she first coined the term 'Computational Thinking'.

off, shampoo bottle closed, dry towel) and an end state (clean dry hair, shower off, shampoo bottle closed – but containing less shampoo, wet towel). The procedure for getting from start state to end state is described by an algorithm that is executed by the processor. There may be more than one algorithm that describes how to get from the given start state to the required end state, and the choice of algorithm may depend on a variety of factors, including the capabilities of the processor (what instructions the person understands), the availability of resources (is there a hair dryer?) and constraints such as cost and time. In order to get from start state to end state the processor may need to collect inputs (shampoo and water), and may provide outputs (soapy water). The system will pass through a series of intermediate states (dirty wet hair, soapy wet hair, and so on), and may cause the state of the environment to change (towel goes from dry to wet).

There are all sorts of activities that may be viewed as computational processes, such as cleaning your teeth, washing up, making a cup of tea, baking a cake, changing the headlamps on a car, or making a journey from London to Edinburgh. Each is instructive in different ways, and each is usable with students at different stages in their school lives.

CT involves concepts and skills that lie at the heart of computing, such as Abstraction, Decomposition, Pattern Matching, Generalization, Inference and Algorithm Design. To discuss these in depth is beyond the scope of this article, so we'll take a closer look at them, and how they might be taught, in future issues of SwitchedOn.

TEACHING CHILDREN COMPUTATIONAL CONCEPTS A LOOK AT WHAT SOME OF THE RESEARCH SAYS

There have been many research studies relating to how students at undergraduate level develop an understanding of computational concepts, with many tools and techniques developed to support this. But how much of this is relevant to schools? Here are a few studies, all publicly available, that teachers may find interesting.

An article relating to higher education that transfers well to school, written by Lauren Marguilieux and colleagues[1], discusses ways of reducing the cognitive load experienced by students when learning to program. This includes using sub goals, worked examples and scaffolding when teaching new concepts. Another short article by Leo Porter and colleagues[2] describes how pair programming and peer instruction can successfully engage students and support deeper understanding. Such approaches translate well to school. Both articles can be found at computinged.wordpress.com/guzdial-papers/. Much writing has focused on the use of Scratch. For example, research by Joel Adams and Andrew Webster[3] (www.calvin.edu/~adams/professional/publications/) shows that students creating games in Scratch used more loop constructs than those writing storytelling programs in Alice and more variables and if statements than those creating music videos in Scratch. Meerbaum-Salant and colleagues[4] (www.weizmann.ac.il/sci-tea/benari/izer.html) mapped a series of Scratch exercises to the revised Blooms Taxonomy focusing on the cognitive skills of Understanding, Applying and Creating. However, it is not always positive. In another paper by the same team, they suggest that using Scratch may actually embed *bad* programming habits. With reference to how students do learn concepts, a good analysis of constructivist approaches to learning and their relationship to Computing Education is given by Neil Brown in his blog post of 2nd Aug 2013.

And what about very young children? Roland Mittermeir[5] (www.scirp.org/journal/PaperInformation.aspx?PaperID=36667) describes experiments carried out with five and six year olds who were able to verbalise algorithms they devised to sort pencils. He concludes that "*… preschoolers are capable of designing algorithms, expressing them, and understanding them if presented properly. Thus, arguments such as "children below some already advanced age are incapable of abstracting to the extent of designing and working with algorithms" need to be revised by adding the suffix "if not properly introduced in a way conforming to their age and general non-algorithmic capabilities".* Food for thought as algorithms are introduced at KS1.

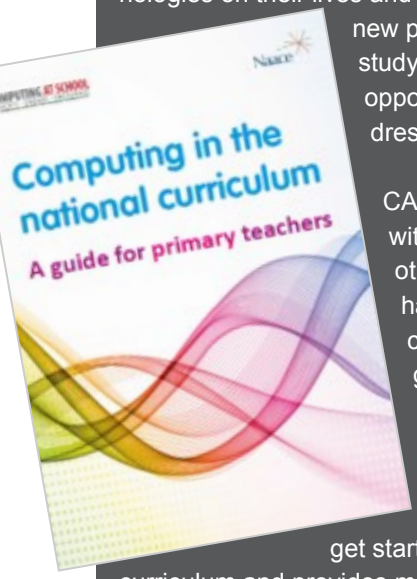
Neil Brown's (academiccomputing.wordpress.com) and Mark Guzdial's (computinged.wordpress.com) blogs are recommended. Both write insightful articles with an awareness of what it means to the teacher. Carrying out small-scale research in your own classroom is also a good way to progress pedagogical approaches and deepen your reflections on what makes a difference for your students.

Sue Sentance

1. Marguilieux, Guzdial and Catrambone (2012). Subgoal-Labeled Instructional Material Improves Performance and Transfer in Learning to Develop Mobile Applications.
2. Porter, Guzdial, Simon and McDowell (2013). Success in Introductory Programming.
3. Adams, J. and Webster, A. (2012). What do students learn about programming from game, music video and storytelling projects.
4. Meerbaum-Salant et al. Learning computer science concepts with Scratch and Habits of Programming in Scratch
5. Mittermeir, R. (2013). Algorithms for preschoolers: A contradiction?

CURRICULUM GUIDANCE FOR PRIMARY SCHOOLS PUBLISHED

The 2014 national curriculum introduces a new subject, Computing, which replaces ICT. This represents both continuity and change, both challenge and opportunity. It gives schools the chance to review and enhance current approaches in order to provide an even more exciting and rigorous curriculum. It can address the challenges and opportunities offered by the technologically rich world in which we live. Many primary teachers currently equip pupils with high-level skills in using ICT, preparing them to apply these across the curriculum in secondary education. It's less clear that pupils leave primary school with much knowledge of how technologies such as computers, software, the internet, the web and search engines work, or a critical understanding of the impact of these technologies on their lives and on society. The new programme of study provides the opportunity to redress the balance.

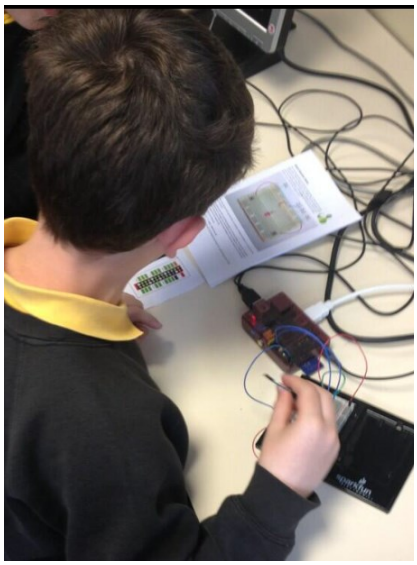


CAS, in association with Naace and other partners have published a comprehensive guide for primary school teachers. It explains how primary teachers can get started with the new curriculum and provides many pointers to excellent resources and ideas for building an innovative and exciting curriculum.

It's a really exciting time to be a primary school teacher. Don't be daunted by the changes in the move from ICT to computing. Rather, see this as an opportunity to develop your own knowledge about computing and to learn to program, if you've never had the chance before. Although this might sound like hard work, it's actually great fun. You'll find that you make better use of the technology you have at home and in school, and also that you start to think a bit differently, looking at systems and problems in the same way a computer scientist does.

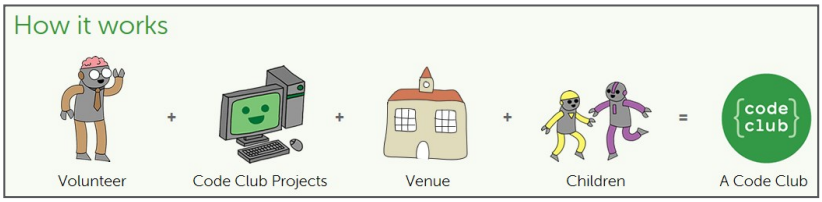
NEED HELP GETTING STARTED? CALL CODE CLUB VOLUNTEERS

Asking for local IT professionals to work with you in an after school club can be a great way to take your first steps in Computing. Sarah Zaman, from St Johns CE Primary School, Stockport extols the virtues of Code Club.



I first found out about Code Club on Twitter after another teacher tweeted about some of the projects their volunteer had been doing as part of their after school club. On Googling it I registered my school and waited for a response. Code Club is an after school programming club for 9-11 year old children run by volunteers. It was set up by two women whose aim is to have a Code Club in every primary school eventually, which obviously is a huge task. They write the materials and the volunteers run the clubs. I initially was contacted by an IT professional from IBM, Joe, who has been with us for over a year now. The children have learnt Scratch, Python including Minecraft Pi and even breadboarding with some material from a small company called Emergent Value, as well as material Joe has written himself. They have made good use of the five Raspberry Pi at our school and three children have now bought their own.

In the Spring term Joe was approached by a Professional from Siemens who asked if she could run a small Code Club alongside Joe with his support. She now has a large group of Year 4 and 5 children who are progressing well through the Scratch projects and we have requests all the time from other children asking to join as it is so popular. I cannot stress how beneficial to the children these clubs and their volunteers are, as their knowledge and commitment has been excellent. Some Fridays I struggle to get the children to go home as they are so focussed in what they are doing. You can sign up to try and find a volunteer of your own at www.codeclub.org.uk.



Code Club was co-founded by Clare Sutcliffe and Linda Sandvik in April 2012. In just 20 months it has grown from an idea to having over 1600 clubs up and running. Volunteers go to their local club for an hour a week. Code Club authors write a new set of projects for volunteers to use every term, so you don't need to worry about materials. They want to put a Code Club in every one of the 21,000 primary schools in the country. It's a big task but if you are an IT pro and think you can help support them, please sign up as a volunteer on their website.

A CONSISTENT LOCAL APPROACH HELPS TO LAY FIRM FOUNDATIONS

Developing a joint understanding with your local Primary schools makes a lot of sense. Emma Partridge, from Settle College, North Yorkshire has already made a start.

Any structure requires a solid foundation to build upon. We know that to enable students to understand and apply more advanced concepts, they need to cover the basics first. Without those building blocks, we often learn without deeper understanding and progression is slower. As highlighted in the previous issue of SwitchedOn, Primary education will play an important role in delivering the Computing curriculum. Without their skilled delivery, we find ourselves delivering or worse, correcting, computing basics before we can move on to deliver the KS3 curriculum. The focus on Computing is still very new, therefore it is no real surprise that there is a significant number of schools who are not yet au fait with the impending changes.

An offer of a Computer Science CPD was quickly snapped up by our local primary schools. The aim was to demystify the new curriculum, link it to some key areas of computational thought and provide suggestions and resources to help them meet the new requirements. The feedback I got from the group before we started, revealed that they had very little or no experience with computing and they were not confident about the new curriculum. None of them knew about CAS, which poses a question. How many primary schools are out there with no support and don't know where or how to get it? Have you contacted your local schools?

After playing Tic Tac Toe against CS4FN's intelligent piece of paper. some time was spent explaining the reasons for the change and working through the KS1 and 2 programmes of study, an activity that really helped to give teachers a sense of direction and understanding about what we are trying to achieve. An introduction to KODU followed. Making a basic world before programming a player to collect hearts to win led to excited discussion about how it could be used in lessons, from exploring Mars, to telling a story. Rude Ralph, a cheeky boy who blows raspberries, introduced Scratch as a KS1 project. It was a very simple program that allowed the group to get used to the environment before we looked at building a more complex game using operators and variables.

By the end of the session, our primary teachers were enthusiastic and positive about Computing. They felt they had a much clearer vision of the requirements for KS1 and 2 as well as confidence in utilising KODU and Scratch to support their students learning. As one participant commented, "For someone who has no experience and is a complete novice at computing, at least now I understand the links to logic and problem-solving. In other words, I can now see there is a point to it all! "

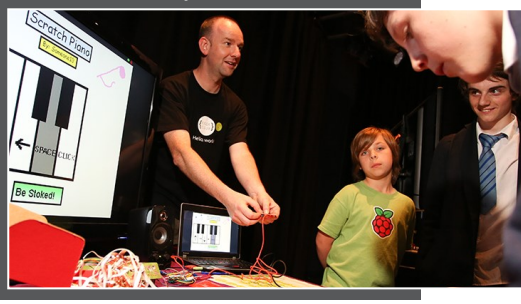
A suggestion of starting a hub generated keen feedback. Initial plans for the hub will be focusing on a practical session looking at input and output devices, further CPD on programming (using kinaesthetic methods as well as IDEs) and collaboratively developing resources. Working together can help develop a common sense of purpose, and strengthen links as well as laying firm foundations to build on at Key Stage 3.

FIND INSPIRATION WITH Animation14

If you are a teacher for whom much of the Computing curriculum is new, what better way to dip your toe into the water than encouraging your pupils to enter Animation14? The annual UK Schools Computer Animation Competition, now in its 7th year, is all about enthusing schoolchildren about Computer Science, and using computers creatively. It is aimed at students aged 7-19 in UK schools and colleges and is free to enter. For details and galleries of previous creations see animation14.cs.manchester.ac.uk/

Entrants create an original animation, up to one minute in length, using any of the eligible software which includes Alice, Flash, Scratch, Blender and others. Entries can be from individuals or small teams. There are prizes for each age category (KS2, KS3, KS4 and KS5) and include laptops, iPads, iPods, games consoles, vouchers and more! This year, alongside the normal categories, there is a special award, sponsored by BCS, The Chartered Institute for IT for animations on the theme of e-safety.

Last year saw over 1,100 submissions. In July 2013, 41 students from 24 schools collected nearly 50 prizes and the University played host to over 400 schoolchildren at the Animation Festival and Inspirational Computer Science Day. The closing date for this year's submissions is 28th March and the festival is scheduled for 11th July.



HOW CAS CAN HELP DEVELOP TEACHERS IN PRIMARY SCHOOLS

Andrew Shields is a primary school teacher working in Leicestershire. He is an SLT focussing on computing and is developing CPD sessions for staff across the county. We asked him why he joined CAS?

"Let's face it, there are lots of places where you can gain knowledge on the new curriculum and computing. You can join Twitter and build up your links to talented people. Follow a blog or two, or three, or you can join Naace and take some of their on-line ICT; cpd4free. Google+ has a thriving education community. All will move you, your knowledge and your skills forward. So why CAS? I stumbled upon CAS while looking for something to explain the proposals for a draft computing curriculum. CAS was part of the team helping to develop the content of the draft. It cost nothing, and through joining up I was able to engage with a group of people who wanting to move things forward. They produce this termly magazine, help signpost regional events and training, have a growing range of resources and a discussion area where very helpful people will always be willing to assist you.

At our area's inaugural CAS Hub meeting I was able to have my first hands on look at a Raspberry Pi and get to boot it up. I met people keen to share and explore computing issues from both education and private sectors. I came away with a new found confidence and insight into how computing could fit into the primary school.

My daily e-mail CAS Community digest keeps me up-to-date and is a good starting point for finding answers to questions. Over the past few months I have noticed a growing number of primary school teachers joining CAS. As more of us join, the more we are able to explore what is out there and what is available to us. The more insight we can get through collaboration with others, the quicker we will get there. CAS, for me, is a big part of my collaborative efforts to move my knowledge, skills and teaching forward. Could it do the same for you?"

INTRODUCING ALGORITHMS TO CHILDREN IN RECEPTION CLASS

Stuart Hadfield, from Saint William of York Catholic Primary School in Thornton, Merseyside shares his experience of teaching algorithms to Foundation Stage children.

As ICT/Computing co-ordinator, I decided to show staff how this new, fun and exciting curriculum can be taught to children throughout school. We looked at the new Computing Program of Study to look at fun ways of delivering the new content. Initially staff struggled with the actual wording and thought that it was beyond themselves to understand, let alone to teach. I decided to focus on looking at the first bullet point within Key Stage 1 and explaining how this could be taught to children as young as 3. I was then 'challenged' to run a 'Day Of Computing' in our Reception class to allow others to see how young children could be taught to 'Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.'

I displayed around the room pictures of everyday algorithms (getting ready for school, recipes, brushing teeth) alongside the appropriate vocabulary, including the word 'Algorithm'. I then set up a carousel of activities. I displayed colour coded 'I Can Statements' on each activity so staff could see the progression in skills being taught. **Red** denoted emerging skills whilst **Yellow** indicated things expected by the majority of children. You can see some examples in the column on the right.

Activity 1: Children used our kitchen role play area to make cups of tea for the visiting members of staff. They were encouraged to talk through the separate steps and use the actual term 'algorithm'.
Activity 2: This was set up in our outdoor area. One child was blindfolded and the other children asked to give them careful instructions to walk through a maze made of large building blocks. They absolutely loved this activity. Again I ensured that they used the correct vocabulary for control and the term algorithm.
Activity 3: This activity was the first to require Computing resources. I introduced our bank of 12 Beebots and a variety of Beebot mats. After initial exploration of the Beebots the children began to program a sequence of instructions to accomplish a variety of outcomes. One example was to navigate the Beebot around the Grand Prix course without hitting the walls.
Activity 4: The four class PCs were set up with each running a different, appropriate program to allow the children to experience them at as early an age as possible. I again asked the children to explain 'what happens when …' so they understand that the computer is following their instructions.

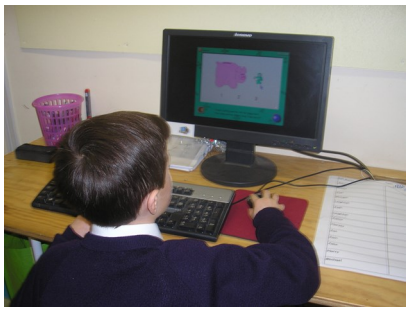
It is no exaggeration to say the children loved their Day of Computing and were happily using the vocabulary from the program of study after just one day of work on algorithms. Staff too stated how much they had enjoyed the experience of the day. More importantly, seeing Computing teaching in action removed some of their own reservations about teaching the new programme of study to their own year groups.

EXAMPLES OF 'I CAN' ALGORITHM STATEMENTS

Activity 1: I can use role play toys and do things in the right order e.g. make a cup of tea and say this is an algorithm.

Activity 2: I can tell my friend to do something in the right order eg get through a maze with a blindfold on and I can say this is an algorithm.

Activity 3: I can control a toy by programming a sequence of instructions and understand that this is called an algorithm.



Activity 4: I can say programs are algorithms working on computers.

HOW 'KIDS CAN CODE' HELPED KICKSTART COMPUTING

Janu Shan

We call ourselves 'Kids Can Code'. We believe children are the future and require an education that actively involves them. They need to know how to code. Today's children need the fundamental skills taught by Computer Science. Computer Science teaches children how to think. Our project first came into existence when sixth form student, Janushan Shanmugavadivel (Janu) had the realisation that his transition into A-level computing would have been made infinitely easier if he had learnt basic computer skills as a child. So why wasn't computer science part of the primary school curriculum?

The curriculum needed change. With that question as our foundation and helping children toward a brighter future as our mission, Janu founded the 'Kids Can Code' outreach programme in October 2012. Since then, the project has blossomed. Whilst in year 12, Janu, with help from his friend Rohan Warrier, started an after-school computing club at Newbury Park Primary School, Redbridge with 17 children. By the end of the academic year there were 82 involved! Today, over 80 learners in Redbridge and around 40 in Waltham Forest are part of the programme. By the end of January there will be another 40 more children that 'Kids Can Code' will teach.

Our organisation offers both 8-week and 21-week course options to teach children to become avid problem solvers literate in Scratch and, if they complete the longer course, Python. Currently, the 'Kids Can Code' Curriculum Developers are working to integrate Computing and basic Electronics using the Raspberry Pi computer so children are better prepared for an opportunity to employ their creativity and logic in innovative projects.

Pupils took to this project like ducks to water. Programming and lateral thinking was a perfect opportunity for them to fully exploit their love of creativity; it was almost natural for them to understand and develop code. It's not a rare event for a child to talk to us about the many projects they've programmed at home or even taught their class. The children's enthusiasm, fuelled by the beauty of computer science motivates us to continue supporting them. Moreover, the kids love it!

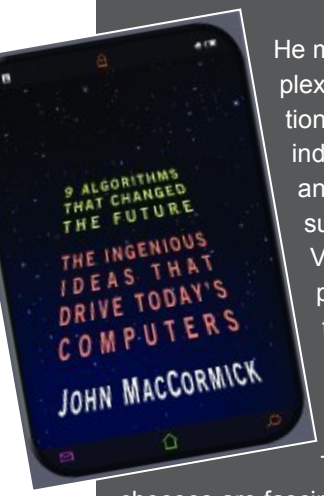
Inspiring youngsters at Ilford Jewish Primary School

The whole project was developed, set up and is run by students from years 10 to13 from two different schools: Forest School and Ilford County High School. Currently 26 volunteers teach in 11 different primary schools. The entire project is student-led and student-run. Ilford County High School's Head of Computing and CAS Hub Leader, Mr Jogia and the Liaison Officer, Mrs Hitchcock have greatly supported our initiative. Mrs Pearce, Head of Computing from Forest School, has also aided our efforts but there are still a lot of high schools with strong Computing departments that 'Kids Can Code' would love to work with. We hope you've enjoyed reading about our drive to ensure our passionate beliefs become reality. Kids truly can code, we just need to give them a chance.

Janu Shan & Guy Coop

INGENIOUS ALGORITHMS THAT HAVE CHANGED THE FUTURE

Written by John MacCormick in 2012, this excellent book aims to explain in layman's terms how our computers can perform some highly complex tasks with relative ease. MacCormick takes nine algorithms that underpin our daily use of computers, concepts like pattern recognition, data compression and error correction, and then explains in a clear and accessible way how the algorithms behind the magic are constructed.



He makes sense of the complexity of public key encryption and of search engine indexing; he gives context and background to clarify, such as his tour of Silicon Valley garages, then explains the complexity of the algorithm in a disarmingly straight forward way.

The algorithms he chooses are fascinating yet familiar, and his analogies are engaging. In a closing chapter MacCormick explores 'What is Computable?', skipping through bugs and crashes, then undecidability, again laying complex ideas open to the less technically-minded. I enjoyed this book immensely; however it appeals to my students too and succeeds in giving them a richer context for their learning. *Lyndsay Hope*

NEW CODECADEMY RESOURCES

Codecademy has recently released a beta site full of free interactive courses, slides and quizzes for teachers to use. Materials for Python, JavaScript, HTML/CSS and jQuery are available. Sharon Angland, from William Howard School in Cumbria, commented: "We've been using these as part of an internal pilot of the new curriculum, and the presentations have been a valuable 'leap off' point for teaching a wide variety of abilities. They are clear and easy to understand even for subject non-specialists. The quizzes are great. Speedy feedback with a check mark is a nice touch." For more details, email Leng Lee leng@codecademy.com

MAKING MUSIC WITH APPLES, PEARS AND A LITTLE AUDACITY

Last summer Christine Swan, from The Stourport High School and Sixth Form, webcast a lesson with Maggie Philbin, founder of TeenTech. She outlines her eclectic approach to investigating and capturing music.

We talked about how to enthuse students and provide an opportunity to learn through having fun. The lesson looked at how analogue sound can be input and stored as a digital data and how it can be output by being triggered from a program. The students were from Year 9 who had just started their GCSE Computing course. A key objective was to give the students real hands on experience. I used apples and pears to connect to a Makey-Makey to act as an input device for playing music. The students were familiar with both conventional input devices and musical instruments so the chance to create something very new was a great experience. Students came up with their own ideas too, using pencil pictures and even themselves as input switches.

The next challenge was to create sound, by any means, which was recorded and edited using Audacity. Key concepts about analogue sound waves needing to be sampled and how the sample rate affects playback quality were introduced. Audacity is great for playing around with recorded sound and applying a variety of effects.

Once their digital sounds were created, we needed to use a program to trigger playing them. Some did this using Scratch but I wrote a basic Python and Pygame program to get them started. The Pygame mixer module contains everything needed to load and play sounds and you can create a GUI to boot. The students were then able to edit the code to change the GUI and alter the sounds being played. The original idea came from watching the amazing work of the digital musician and multi-instrumentalist, Jack Conte, but my intention was to make a low-tech music player. All that was required was creativity and a little coding skill.

The students had a huge amount of fun and Maggie encouraged them to think about the wider issues relating to Computer Science also. "It was great to see students so engaged and enthusiastic" commented Maggie. "They were clearly enjoying experimenting and having some freedom to create. This is what the new curriculum should be about." If you haven't had a chance to look at the Live Lesson, you can still view it here: www.tes.co.uk/article.aspx?storyCode=6361228. A set of resources will appear on the OCR website shortly.

UNDERSTANDING ABSTRACTION WITH ROCK, PAPER AND SCISSORS

Do you wish there was a project which is fun, has a lot of possible extensions, keeps pupils of all abilities engaged and generates lots of interest? Ilia Avroutine, Key Stage 5 subject leader at Royal Grammar School, High Wycombe gives one suggestion.

Rock Paper Scissors (RPS) is a time honoured exercise for teaching computing with unlimited differentiation potential. Kids can relate at any level and it is a game – win! For the teacher, the goodies to tick off the scheme of work include an interface, multiple assignments, data types (strings vs. integers), random numbers, iteration and a lot of selection. Do I hear validation mentioned by readers? At the top end, this could be a good introduction to modelling and statistics and a time to introduce object-oriented programming. In this article though, I want to focus on a particular concept linking numeracy and computational thinking.

The heart of RPS is the selection routine which can be transformed with the use of abstraction. Having previously learned IF ELSE, your pupils would, no doubt, enthusiastically start typing away at all possible combinations of the two choices (written in a generic pseudocode syntax):

```
IF Player1="Rock" and Player2="Paper"
THEN Winner=Player2
ELSE IF Player1="Rock" and Player2="Scissors"
THEN Winner=Player1
...
ELSE IF Player1="Paper" and Player2="Rock"
THEN Winner=Player1
```

An experienced coder should feel uncomfortable typing repetitive lines but let the kids do that first – it will set them up for your next revelation: **Abstraction**. Abstraction looks for patterns from which to generalise and extrapolate. It is at the heart of computation and scientific thinking. First let's state the non-abstract summary:
Rock smashes Scissors: Rock wins.
Scissors cuts Paper: Scissors wins.
Paper wraps Rock: Paper wins.
If one player wins, they get a point. If the choice is the same, it is a draw.
Now, let's "abstract it". Do I see any patterns? Rock is bigger than Scissors. Scissors are bigger than Paper. Paper is bigger than Rock. The key with abstraction is often converting word concepts to numbers. Let's imagine that RPS is actually a dice game with a twist: Rock is 0, Paper is 1, Scissors is 2. Instead of comparing them, let's find the difference between players' choices. Take one combination: Rock = 0, Paper = 1. Rock - Paper = -1 which is < 0 so Rock loses. Paper minus Rock is1. As this is > 0 Paper wins. Let's test for another combination: Paper = 1, Scissors = 2. Paper minus Scissors is -1, less than zero so Paper loses and so on.

The pattern that emerges shows that ***if the difference between Player 1's choice and Player 2's choice is -1, Player 2 wins***. This is much easier and compact than the original comparisons. At this point you might say, "but Scissors (2) loses to Rock (0)…" Yes, it kind of wraps around, so we have to deal with it like so:

```
Scissors - Rock = 2.
IF (Player1 - Player2) = -1 OR (Player1 -
Player2) = 2 THEN  Winner = Player2
```

A proof of principle in Python is shown. Notice the Pythonic use of lists instead of a typical SELECT CASE statement to convert numbers back to Rock, Paper and Scissors .

Who says such problems can't be solved with good old ICT? Below it is the same solution in Excel, using nested IF formulas, random number generators and VLOOKUP. There you have it. A good prequel to RPS is a coin/dice toss game, which has most of RPS features but easier selection. An extension is Rock-Paper-Scissors-Lizard-Spock ("hit the googles" for this one) or modelling a slot machine which, if you think about it, is just RPS with 3 hands (question to the class: why?). With the slot machine you are again moving to a different level of curriculum reach – from text on a line to a gorgeous App-style GUI with animation and flashing lights.

```
import random
#Set up our variables
Options=["Rock","Paper","Scissors","Draw"]
Player1=0
Player2=0

#Abstract pattern selection function, two parameter inputs and one output
def whoWins(pl1,pl2):
    Diff=pl1-pl2
    if Diff==0:
        return 3 # 3 represents Draw in our list
    elif Diff==-1 or Diff==2:
        return pl2
    else:
        return pl1

#main program: testing our selection by running 30 Computer vs Computer games
for counter in range(0,30):
    Player1=random.randint(0,2)
    Player2=random.randint(0,2)
    print(Options[Player1],"vs",Options[Player2],
          "winner:",Options[whoWins(Player1,Player2)]) #one line split in two
```

	A	B	C
1			0 Rock
2			1 Paper
3			2 Scissors
4	Player 1	Player 2	Difference
5	0	2	-2
6	Rock	Scissors	
7	Who wins?		
8	Player 1		

	A	B	C
1		0	Rock
2		1	Paper
3		2	Scissors
4	Player 1	Player 2	Difference
5	=RANDBETWEEN(0,2)	=RANDBETWEEN(0,2)	=A5-B5
6	=VLOOKUP(A5,$B1:$C3,2)	=VLOOKUP(B5,$B1:$C3,2)	
7	Who wins?		
8	=IF(C5=0,"Draw",IF(OR(C5=-1,C5=2),B4,A4))		

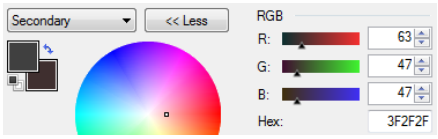
GETTING TO GRIPS WITH DATA REPRESENTATION AND GRAPHICS

Most ICT teachers will have taught a graphics module of some sort. SWITCHEDON editor, Roger Davies, who teaches at Queen Elizabeth School in Cumbria, looks at several ways to introduce computing concepts through manipulating graphics.

There is a lot more to Computing than just programming and no need for ICT teachers to abandon all previous ideas. 'Traditional' areas of the ICT curriculum, such as graphics or sound editing provide fertile areas for children to explore the basic ideas of how data is represented in digital form. Instead of binning previous ICT work, why not seek to deepen the content instead? Links to all the materials introduced here can be found at http://community.computingatschool.org.uk/resources/1627.

Given the prevalence of digital cameras, most secondary age children have some appreciation that bitmap graphics are built from a grid of pixels. That said, it can be worthwhile completing the excellent paper and pencil exercise developed by CS Inside (Painting By Numbers). In a paired activity, each child digitises a black and white image, whilst their partner attempts to render the image at two different resolutions. CS Unplugged have produced a similar type of exercise (Picture This), that uses run-length encoding to describe images which the pupils attempt to render. That there are different ways to describe images with numbers is well explained in a feature on the cs4fn website, Pixels and Puzzles, which also provides a challenging logical puzzle which could be used as an extension or homework. Restricting introductory exercises to black/white (pixel values of 1 or 0) provides a good introduction to the binary exercises developed by Neil Kendall (left).

What if we want to encode colour images? We can now introduce the idea of each pixel being a number that represents a colour. Investigating colour depth helps cement an understanding of binary. 4 bits per pixel will allow only 16 separate colours to be represented, 8 bits will allow 256 variants



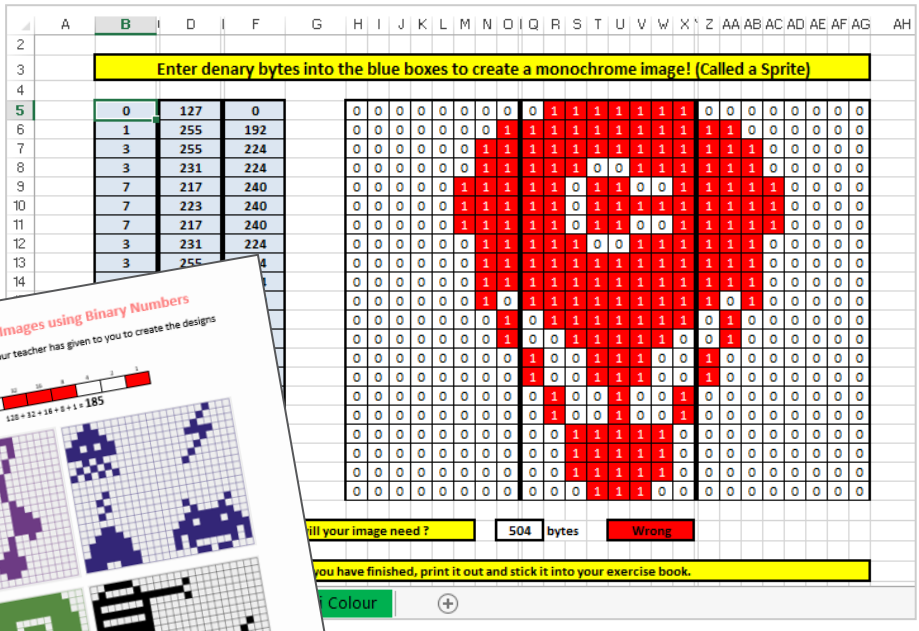
and so on. A simple investigation of the colour palette in Paint (above), will illustrate how 24 bit colour allows for the combination of three channels of Red, Green and Blue values between 0 and 255; 1 byte for each channel. Conveniently, Paint also describes the colour in hexadecimal format too, so there is scope to look at how hex can be used as a shorthand representation for 4 binary digits. If your pupils have experience of HTML, you might want to get them to investigate HTML colour codes, which use the same hexadecimal representation.

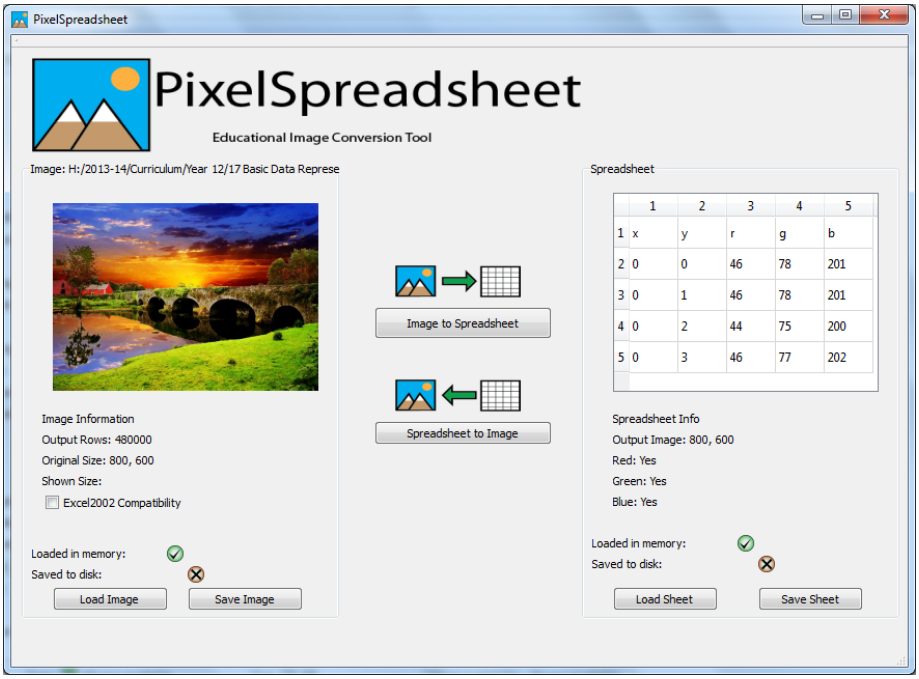
If we can represent both colours and text with bit patterns, we should be able to represent a picture as text. This is often a revelation to pupils. GIMP is a free, open source, fully fledged image editor on a par with Photoshop. It will render the netpbm

CREATING SIMPLE SPRITES WITH BINARY NUMBERS

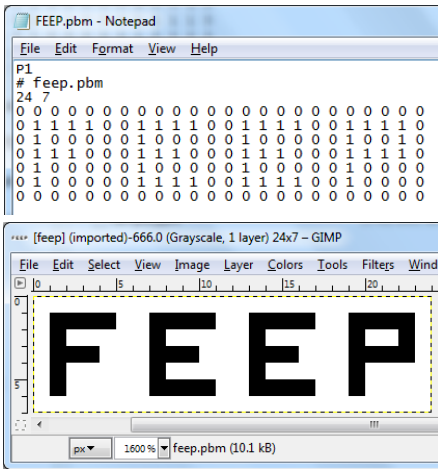
Once pupils can grasp that pictures can be described by numbers, it is a logical step to investigate how computers handle numbers. Again, there are excellent activities by CS Unplugged to introduce binary numbers and also simple text encoding. If pupils understand that both numbers and words can be described by the same binary bit patterns, it makes it easier to see that images are simply another interpretation of a set of bit patterns.

Neil Kendall, a Computing teacher at Beckfoot School in Bingley, shared the exercise below with the CAS Community. The spreadsheet allows pupils to enter three denary bytes per row, which are converted to produce a monochrome image. A second spreadsheet investigates how multiple colours can be represented. A sample worksheet is also available. A former 'bedroom coder', Neil previously wrote the game 'Lethal' for the Commodore 64. Retro gaming fans will recognise the Commodore balloon sprite in the example!

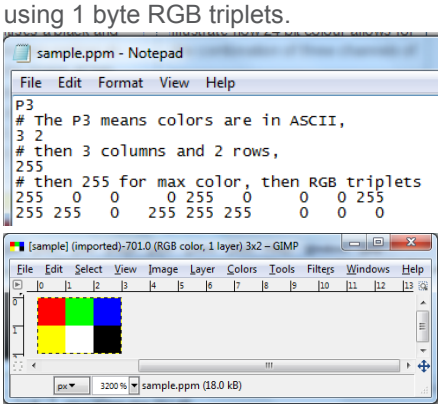




family of image formats, namely PBM (monochrome), PPM (colour) and PGM (greyscale). These can be created in Notepad. The simplest example is pbm, where the ascii text code for 1 and 0 render as black and white, as shown below.



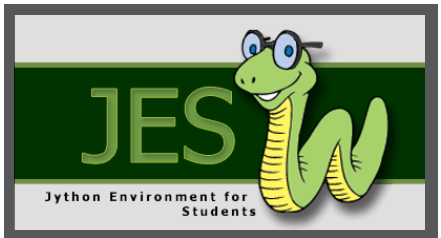
Note the file header: the information before the image data, which is required to render it correctly. Once familiar with monochrome, pupils can investigate and create colour images. The example below is a 6 pixel image using 1 byte RGB triplets.



Pupils can appreciate the sheer scale of data contained in image files by using Pixel Spreadsheet, an awesome utility developed by Mark Guzdial (right) and colleagues, who have pioneered the use of 'media computation' at Georgia Tech in the US for over ten years. One of the powerful ideas in media computation is data abstraction: that visible manipulations of pictures are made by changing low-level, almost unseen objects - pixels. Pixel Spreadsheet allows students to load images, extract the pixel data and save it to a spreadsheet. From here, they can apply simple formula to manipulate the values. Setting one colour's values to 0 will tint the picture, for example. Subtracting each value from 255 creates a negative image and averaging the 3 colours will render greyscale. Once reloaded, the new values can be rendered as a new image and compared with the original.

Using this utility, pupils will immediately see how simple Photoshop effects can be achieved. They can be given a series of challenges to test their ability to express data manipulations as formulae within a spreadsheet. Try rotating or flipping an image for example.

I have found activities in Pixel Spreadsheet to be a real eye opener for kids. However, it's hard to do everything by manipulating a spreadsheet. Let Mark show you how to do it quickly in three lines of Python code…

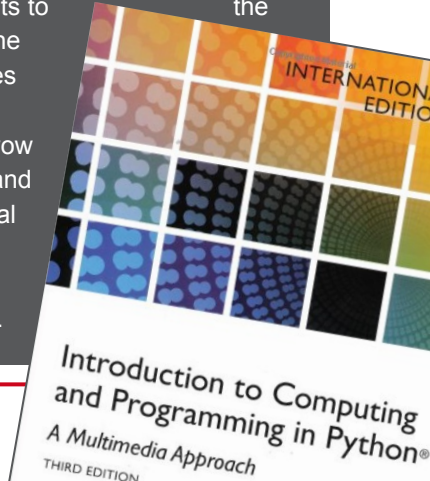


Welcome to JES, a wonderful Python implementation developed by Mark Guzdial and colleagues with libraries to manipulate not just pixels but sound samples too.

Much media manipulation involves simple calculations iterated over a large data series, be it pixels or sound samples. There is something deeply satisfying for pupils in coding simple loops which have immediate visual or auditory impact. Good coding needs firm foundations. Firm foundations are built through repeated exposure to simple ideas. Multimedia computation allows pupils to develop procedures to alter images that help reinforce basic concepts like selection and iteration.

JES is supported by an excellent book with activities that go well beyond the basic manipulations possible in Pixel Spreadsheet. Students can write code to blend, blur or posterize images. Techniques such as using chromakey are well explained, but most importantly, the computer science ideas behind them are constantly drawn out and reinforced.

Further sections on sound and video editing are equally deep and the final chapters bring the CS concepts to the fore. The activities have a huge wow factor and are ideal for use at KS4 or KS5.



ESSENTIAL IDEAS CAPTURED IN COMPUTER SCIENCE FIELD GUIDE

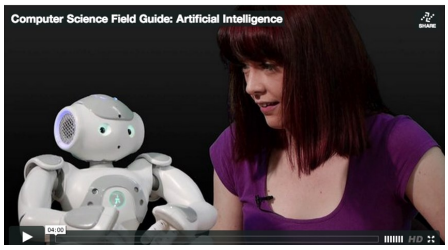
Tim Bell, best known to readers of SWITCHEDON as co-author of the Computer Science Unplugged activities, outlines an exciting new initiative that will become a central reference source for those new to the ideas of computer science.

The "Computer Science Field Guide" (csfieldguide.org.nz) is an online, open source "textbook" that is being developed to support the topics recently introduced in New Zealand high schools. The guide is being developed in a way that it could eventually be adapted for use in other countries.

At the time of writing it covers 6 topics: Artificial Intelligence, Complexity and tractability, Formal languages, Computer graphics, Computer vision and Software engineering. More topics are being added, but these were the ones for which there was the least material currently available at the right level. Topics in preparation include Data representation, Algorithms, Human-computer interaction and usability, Programming languages, Compression, Encryption and Error control.

The field guide project is still in its early stages, but is being made available in "beta" form. Thanks to sponsorship from Google Inc. we have been able to make it an open resource. The source material uses "Restructured Text" markup language, so it can easily be repurposed - for example, both the student and teacher versions come from the same source.

The urgent need for a field guide became apparent as the new computer science standards for high school were rolled out in New Zealand in 2011 (see page 23). We had identified thousands of resources for teaching the new topics, but they were a disparate mixture of web pages, articles and book sections, many of which were relevant but not directly targeted to students' needs, with the subject material not matching exactly, or the style of writing being unsuitable.



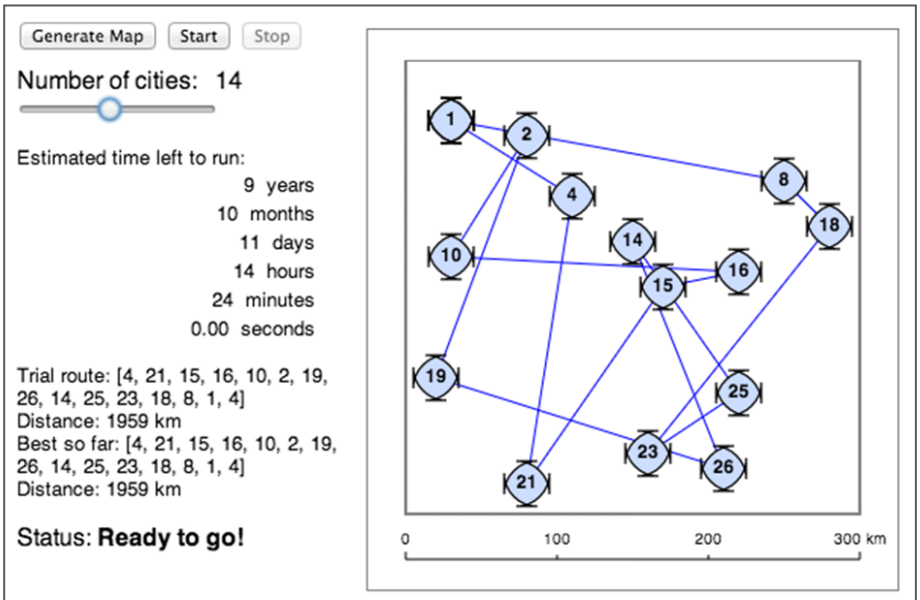
To address this we have used the following strategies:

- quirky introductory videos to capture students' interest
- engaging interactive activities to exercise a concept
- down to earth explanations that relate the topic to students' experience
- material that covers the requirements of the curriculum closely
- extension material is provided for keen students, but it is clearly marked
- no requirement to do programming
- available online and able to be used on most modern browsers and tablets
- open source, so that others can adapt it, and make local copies or versions without restriction (including all videos and interactives)
- a version for teachers with lots of hints, alternative ideas, and solutions

The lack of programming may seem odd, but this is one area that has a lot of coverage online already, and because students are only starting to learn programming in parallel with learning CS concepts, we didn't want a lack of programming experience to be a barrier. For students who are confident programmers, there are options for implementing the CS ideas on a computer, but the emphasis is on ways to grasp a concept and demonstrate it in a project even with very limited programming skills.

A key point is that we want students to understand the breadth of computer science; they don't need to become experts, but they have a chance to find out the range of areas that are covered and the kind of thinking required to work in computer science. We want them to have a meaningful engagement with those areas, which will help them to decide if it is something they have a passion for.

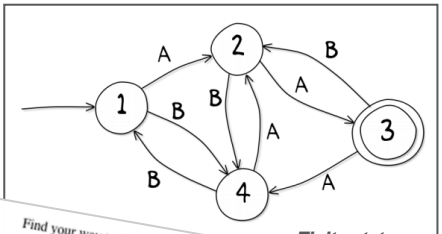
Although the guide is initially being developed for NZ schools, it has its genesis in a project by Peter Denning, which involves out-of-school clubs with a mentoring and badge system. The club is being developed in parallel

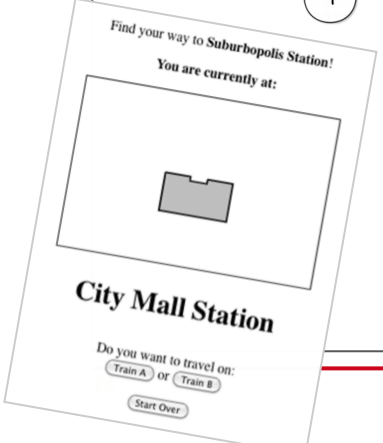


Intractability is introduced by students trying a brute force solver for the travelling salesman problem . The exponential explosion in running time becomes obvious; in the above example, if you press the start button for 14 cities it will take over 9 years, and it gets a lot worse!

(computerscienceclub.org), and the badges enable students to demonstrate competence in the topics even if they aren't taught in their school. This in turn has drawn on the CS Unplugged project (csunplugged.org), which offers ways to teach many topics in computer science without using a computer at all, and related projects such as CS4FN (cs4fn.org) and Computing Science Inside (found at csi.dcs.gla.ac.uk/). The field guide makes heavy use of digital devices for student interaction, but many of the interactive activities build on the Unplugged approach, and in a classroom situation can be introduced using alternative kinaesthetic activities away from the computer.

Currently the focus is on building up content so teachers have access to suitable material, but we will also gradually work on the form so that it is more approachable for high school students. The interactive activities use HTML5, which runs on most browsers and tablets (even the 3D graphics and computer vision activities run in a web browser!) Eventually it will be available in PDF and eBook format and be printed or downloaded if necessary.

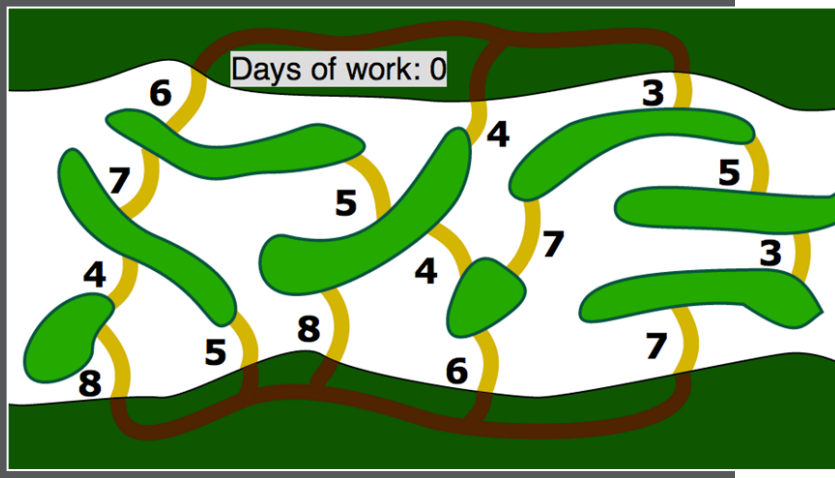




Finite state automata are introduced by taking students on a mysterious train trip choosing train A or B at each station, based on the CS Unplugged "Treasure Hunt" activity.

FIRST UK PARTICIPATION IN BEBRAS CONTEST ON INFORMATICS AND COMPUTER FLUENCY

In November over 21,000 students from about 150 schools in England, Scotland and Wales joined more than 500,000 students world wide in this year's Beaver Competition, organised by bebras.org. Students were entered in four age categories Junior, Intermediate, Senior and Elite covering ages 10-18 in two year bands. It was open to anyone regardless of ability or programming experience as this was about problem solving and no coding was required. The students had to work out how long it would take a network of beacons to be lit across Japan by Samurai beavers; how to send messages with flashlights that can only be 'on' or 'off'; study a printing factory's algorithm to work out which paper patterns were possible outputs. And the question that caused the most problems of all? The bridge building exercise! Here Bob the Beaver had to link both banks of a river and all islands with bridges, all in the fewest possible days. You'll find the answer at the bottom of the page.



The challenge for the organisers was to provide questions that stretch the most able but retain the fun for everyone. Many of the questions were interactive using Javascript such as the Bridge one above: Clicking bridges highlighted them and tracked the number of days of work as students experimented. This helped achieve the fun aspect but meant that thorough testing by schools was a must before the actual week took place.

A post competition analysis showed that in UK schools there was no gender bias whatsoever: It seems that boys and girls of all ages are equally suited to solving computational problems. I hope no one is surprised by this.

This was a trial year. The aim was to find out how to run the competition with a reasonable number of entrants and to get feedback as to whether teachers found it worthwhile. It was only advertised on CAS and was free to enter. The number of entrants was fantastic and the responses were overwhelmingly positive from teachers and students. I am so grateful to all the School Coordinators who took a chance on this year's competition and helped me run the event. I must also thank Eljakim who hosted the UK competition on his servers and provided me with huge amounts of friendly support. It was a very worthwhile project. Unfortunately running this competition for free is not sustainable and will have to be self-funding to continue. Potentially, schools could use their budgets, charge their students or find a local sponsor but none of these options are easy. Moreover, it would mean few schools could enter all their students as many did this year. A large annual UK entry could generate valuable long term predictive data and become a regular feature in our calendar. Can anyone suggest a solution to the funding dilemma?

Chris Roffey

The answer to the Building Bridges problem is 39 days.

NETWORK OF EXCELLENCE

COMPUTER SCIENCE TEACHING

Although only six years old, over 8,000 people have joined CAS - the majority in the last year! The last two years have seen CAS shift from being a small group making the case for Computing in schools, to an organisation seeking to ensure it happens. One focus is on building the 'Network Of Excellence'. But what does it mean? CAS members outline the different pieces of the jigsaw.

EXPLAINING THE NEW NAMES

The main information about how and why schools should join the Network of Excellence (NoE) is at http://goo.gl/990pek. Every school intent on responding to the curriculum changes should be part of the NoE. Any school that is already teaching some Computing and is prepared to share informal advice (perhaps a secondary school being the point of contact for feeder primary schools, or a primary school already embracing computing) should consider being a Lead School. Being in the NoE means being 'in the loop' as an institution, rather than as an individual who has joined CAS. Being a Lead School means being willing to help others in whatever small ways are appropriate in your area.

Likewise, if you are a confident teacher who is prepared to share some of your ideas through organising local cpd, do consider applying to be a Master Teacher (MT). As is often said, there is no them, only us …. and if we can grow the numbers in each locality willing to put on a few cpd sessions and offer support, the pool and breadth of expertise will widen. If you feel your experience isn't yet broad enough you can still apply to become a Level 1 MT which will allow you to receive training via Universities in the network. As a model for professional development, locally based, cost effective (minimised travelling), face to face support by fellow teachers is hard to beat. Backed by access to expertise in CS Departments in universities (who are very welcoming) and the more informal 'glue' of the local hubs to strengthen a sense of 'togetherness' and it becomes a very powerful force for professional development.

Most members' first experience of CAS is via the CAS Community website. When launched in August 2012, around 1,200 existing CAS members joined. In the year and a half since then, the membership has grown rapidly. We are growing at a rate of 10% per month. This growth (and the corresponding growth in the number of resources and discussions) presents new challenges. We need to ensure that the rich content of the site is well organised and easily accessible. Thus, we are currently making various alterations.

A major feature of the website is the sharing of teaching material. With over 1,300 shared resources, CAS Online is already well-stocked, but with the impending change to the English National Curriculum (and similar changes in other regions) there will be new teachers arriving looking for resources to use. No one can view them all, so we must make sure they can be browsed by a teacher who is short on time. We aim to organise the resources by curriculum. So, for example, if a teacher arrives at the website needing to find a resource for English National Curriculum Key Stage 3 algorithms, they can find a suitable set of resources. Similarly for Scottish National 4 and so on. We hope teachers will easily be able to find content relevant to their situation, whether they are Welsh primary teachers or delivering Scottish Highers.

We are also taking the website local. Many members have said that conversations on a national scale can be too overwhelming for those taking their first steps in computing. One of our core strengths are the local hubs, and we want the website to reflect this. We are rearranging the forums to promote local hubs with smaller scale conversations, and members recognisable from local meetings. We are also trying to encourage increased feedback in discussions, by adding anonymous like/unhelpful buttons to give small-scale feedback without a full-blown reply.

We are also adding new features related to the Network of Excellence. These will let you see which local schools are participating in the NoE, and allow you to easily find your nearest Lead Schools or Master Teachers. All these sit alongside other usability improvements. We want the website to support teachers in their work and be part of the glue that can cement a vibrant, living community of practice. *Neil Brown*

The CAS Community website is designed and maintained pro bono by a small team at the University of Kent, alongside their regular work. The website's success has been noted around the world, and used as a benchmark to inform efforts to build a similar site in the USA. It's one exemplar of the synergy developing in CAS between schools, teachers and their colleagues in higher education.

HUBS: MEETING LIKE MINDS AND BUILDING A REAL COMMUNITY

Your CAS Profile tells you where your nearest hub is, but have you ever attended? Whether you are a practising teacher, a helpful IT pro, a University academic or a parent of school aged children, a warm welcome awaits you assures Claire Davenport.

As the CAS Hub Support Manager, I see the immediate ripple effect - membership applications rise following Hub meetings, where people have attended for the first time. Hubs offer the personal face of CAS, real 'live' support from keen volunteers who understand the pressures that fellow teachers can face. Hubs promote face to face networking over a cup of coffee and can open up discussion about shared issues, such as how to introduce the new curriculum. They usually include a taster CPD session and they encourage the sharing of resources to help in day to day work. They can be organised by one or more Hub leaders. Sharing the load can make organisation easier for time pressed teachers. Every hub has a working teacher as leader to ensure that every meeting fulfils the brief of providing practical help. Hub meetings are not just talking shops, each provides a takeaway idea or resource that can be used in the classroom the following day.

If there is no hub near you, maybe you could organise one? Passion and interest rather than experience are key. We suggest hosting at least one meeting per term to establish a routine. There is no need to present topics yourself, unless keen to do so. You can gain maximum attendance by ensuring the programme is balanced, targeted and the date set at least 3 weeks in advance. Obtaining feedback can help shape future events. It's worth offering Primary-targeted events for feeder schools. CAS can offer reimbursement for small-scale refreshments and give you periodical updates on national initiatives to share. We can help source speakers, if there's a particular topic you're looking for and offer telephone and email support for new leaders to get the first meeting off the ground. There is a dedicated hub leaders mailing list for sharing, buddying and mentoring and an annual Away Day weekend (Fri/Sat) where hub leaders can get together for hands-on practice of new tools and networking with fellow Hub leaders. We've also written a guide to help with running a hub: http://goo.gl/JpU7ok

The network of Hubs is growing rapidly, with 83 hubs now meeting. Fifteen new groups started last term. Our aim is that members are no more than 40 minutes from their nearest hub.

SHARING EXPERTISE VIA CAS MASTER TEACHERS

CAS Master Teachers have been receiving training to maximise the impact of their CPD. I attended a session in October at the University of Manchester, expertly run by Ali Redmore of Science Learning Centres and overseen by Mark Dorling. The key to success is ensuring attendees have ownership of their learning, and develop skills and resources they can directly apply to their classrooms.

After lunch we worked in teams of three to develop a 10 minute CPD session for the other course attendees. The day was so engaging I didn't check my phone all afternoon. As you're probably aware Ofsted usually phone the afternoon before they arrive. You can guess where this is going… So I must say a special thank you to others who helped develop the idea for introducing binary numbers which worked so well a day later.

Overall, it was a great opportunity to share ideas and work together. Personally, I enjoyed the chance to work with Primary Colleagues who have so many ideas to offer for making learning active and engaging. It's clear that there are many people doing many things and when we come together we are much more than the sum of our parts. Master Teachers can help develop cpd that is responsive to local needs. We need more! If you'd like to get involved there's never been a better time to come on board. *Tom Attfield*

CAS HUBS SPREAD TO THE FAR EAST!

An enthusiastic team of teachers gathered for the first meeting of CAS Bangkok last term led by Simon Aves. James Abela hosted and brought everybody up to speed on developments in the new Computing curriculum in the UK. We then had a wide ranging discussion on computer languages to teach, importing robotics and linking up with universities at home and abroad. To ensure as many students as possible can participate in a rich program of computing, the international schools agreed that they would work together to bring workshops to Bangkok and discussed ways they could share the costs of bringing international computing superstars to the city.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

THE NETWORK OF EXCELLENCE UNIVERSITY SUPPORT TEAM

Sue Sentance and Nick Cook are both working to support universities directly to develop the Network of Excellence. Nick, a lecturer from the University of Newcastle, speaks from his own experience when he says: "You may be like me. Before last Easter I had never delivered CPD and was unsure of how to start. This is where CAS comes in and for which there is now additional support available. First, we are not expecting CS academics to be experts in classroom pedagogy. That expertise is with the teachers. Indeed, I have found that their expertise can inform our practice. The new school computing curriculum offers many opportunities to forge links. It is also a significant challenge for many schools. Many experienced teachers require training to deliver the new curriculum. They already meet the most important requirement - they know how to teach. What they can lack is confidence in their subject knowledge. Addressing this is a focus of the NoE. It will rely on the efforts of CS departments up and down the country. If every CS department runs a CPD course this year, together we can deliver training to over 2000 teachers!"

TRAINING MASTER TEACHERS

Six universities are the first to run specific training for Level 1 (Trainee) CAS Master Teachers. L1 Master Teachers receive subject knowledge training before becoming a fully fledged L2 CAS Master Teacher. The universities involved are Anglia Ruskin University, Birmingham City University, Edge Hill University, Newman University, Roehampton University and Sheffield Hallam University. More will be offering subject knowledge training for CAS Master Teachers in September and January 2015.

UNIVERSITIES HELP DEVELOP A UNIQUE MODEL OF SUPPORT

The Network of Excellence aims to provide professional development for teachers, largely through the expansion of Master Teachers. This wouldn't be possible without the help and support of Universities. Sue Sentance explains.

When we originally invited schools to be part of the Network of Excellence we hoped to have 250 but over 500 signed up in the first phase! We hoped to have 10 universities on board who would support these schools and now, in England alone, we have 73. Apologies for only giving English statistics but DfE funding for the Network of Excellence only extends to England. Scotland has a similar model called Plan C. All these universities want to join with CAS in supporting teachers as they prepare to deliver the new Computing curriculum. This is a unique feature of this model of professional development. Teachers teach teachers, but they can be supported by a network of universities who have the depth of understanding surrounding the subject. This support can be given in many ways, as David Rydeheard outlines in his article on the facing page. Fostering face to face links between teachers and academics is mutually beneficial, helping break down barriers and scotch myths.

Over the last year many CPD events have been run by universities for teachers. A glance at the Events page of CAS Online at any time will show how many events are being run, at different levels, at different times and in different formats. The response from the academic community to the challenge faced by teachers by aspects of the new curriculum has been overwhelming. The Network of Excellence team are working to support universities who wish to offer professional development for teachers on behalf of CAS. This includes help with booking, promotion of courses and provision of certificates and we encourage all universities to take advantage of this. Courses that are run directly with the support of the Network of Excellence team have a red rosette next to the title of the course on CAS Online. This hopefully makes it much easier for universities to offer courses and enable teachers to see which courses have been provided directly through the Network of Excellence.

Academics in Computer Science departments may be concerned that they do not have sufficient knowledge about the school setting and curriculum and not able to pitch the content to teachers appropriately. Working together with an Education department can help here, as universities involved in teacher training have a good understanding of the needs of teachers and issues relating to classroom pedagogy. In addition, universities and Master Teachers can work together to deliver courses; a model which we have found works very well for teachers. Master Teachers can advise on how to pitch sessions, whilst teachers value the depth of subject knowledge available to them. There are also some off-the-peg tried and tested CPD sessions which universities can take and adapt for their own purposes.

If you are a teacher who would like support or training, make sure you know who to contact at your local university. There is a map on the website Network of Excellence page. Your nearest Master Teacher is shown when you log on to CAS Online – do contact them too as they are there to help, with more Master Teachers coming on board all the time.

COLLABORATION IS ABOUT MUTUAL BENEFIT, NOT JUST ASSISTANCE

Assisting schools to improve Computing teaching is not just an act of goodwill, but vital to university-level Computer Science. David Rydeheard, from the University of Manchester, offers ideas based on his experience.

We have been involved in supporting schools for quite a few years. Activities range from those that help the delivery of the curriculum through to activities for pupils to engage with the excitement of computing. So what can universities do? The issues around up-skilling teachers in computing are widely recognised. Universities can provide a wide range of CPD opportunities from single events, through to courses, unaccredited and accredited, face-to-face or via distance learning. It should be an ideal partnership! Some universities have reported low interest but others run a range of successful courses. There is a number of issues to share to ensure good take-up of such courses.

Universities may act as a 'hub' for local schools keen to collaborate. This can include access to research, industrial and business contacts and the expertise of university staff. Universities may also set up academic-teacher partnering or provide access to advice and guidance for teachers.

The largest resource that any Computer Science department has lies not in its staff but its students. They can be mobilised to support schools and, as a by-product, this may well encourage them to consider teaching as a career. One scheme is "student ambassadors" who support teaching in schools. Incorporating into the university curriculum an opportunity to help can be done by creating a course unit option, for which there is a national scheme, or providing final-year projects which involve developing learning tools for schools. By liaising with local or national teacher training organisations, Computer Science departments can also become involved in initial teacher training. A key aspect of this for universities is to encourage Computer Science students to consider teaching as a career.

Schools gain considerable benefit from good links with companies. Universities are in a good position to broker relationships with their industrial partners and local schools. Computer Science departments may also consider supporting regional or nationwide competitions for school students (such as our own Animation14). Universities can provide activities for schoolchildren, at various levels and with various aims. Examples: (a) open days for computing, (b) specialist workshops for local schools, (c) inviting school students in as ambassadors, (d) summer camps, (e) work experience opportunities, (f) providing 'after-school' clubs, (g) student conferences and (h) offering projects for schoolchildren. Universities can co-operate with national organisations, such as Apps for Good, providing a base for activities and expertise.

This is a personal list. Many universities are already involved and there are other activities which may be undertaken. Computer Science departments are often under considerable pressure in terms of staff and workloads. Some of the ideas above require little staff input but others are clearly additional work. It may be possible to `think creatively' about this, e.g. involving recently retired staff, or former students, or others associated with the university department. If you wish to discuss your involvement with schools and or any of the issues or activities above, please don't hesitate to contact me via david.rydeheard@manchester.ac.uk.

GUIDING PRINCIPLES FOR UNIVERSITY SUPPORT

Here are a few principles guiding how universities may support schools and colleges:

- University support for schools is best viewed as co-operation for mutual benefit. Teachers need to advise universities how they can most productively support them and their schools.
- Research-led universities should consider how to bridge the gap between research and school-level teaching. Bringing the excitement and achievements of world-class research to schoolchildren can enthuse them and increase awareness of science.
- Support is needed from universities at all levels of education from primary, through secondary, to college level.
- Universities need co-operation among themselves to co-ordinate a network of support, and with others such as Master Teachers and schools and colleges in the Network of Excellence.
- Women are poorly represented amongst CS UG students. Universities are recruiting from only half the population! Many types of activities can help to address this.

ADMISSIONS CRITERIA

A final important point which many universities are already considering. CS departments need to encourage, in their admissions process, CS qualifications in schools. Moreover, science departments in general need to accept CS qualifications. The latter is a particular problem: why should pupils take computing qualifications if they are not recognized in university admissions? The next Russell Group Informed Choices guidance will list A Level Computing as useful for a whole range of degree programmes. This will go some way to addressing assertions that University admissions officers do not value Computing A level.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

WARWICK UNIVERSITY MOOC GETS LARGE TEACHER TAKE-UP

For the last few years the Department of Computer Science at the University of Warwick has been running CPD sessions for local teachers. Early last year we ran a 6-session course which combined lectures on computing concepts, some Python labs and time for teachers to share their experiences of teaching computing. The course was a great success but we were unsure how to run it again. We wanted to reach beyond local teachers and also avoid the perennial problem of teachers getting released. A successful application to Google's Computer Science for High School fund (www.cs4hs.com/) allowed us to adapt the material for delivery online.

The course launched in November in two modes – a free mode with over 500 teachers signed up to support each other through 8 fortnightly sessions, and a tutor supported mode with over 30 teachers enrolled with additional support from Warwick staff and PhD students.

The course is aimed at teachers with little, or no programming experience who may be required to teach the Computing GCSE qualification in the near future. Like its face-to-face predecessor, it covers 3 areas:

- Computing concepts
- Programming in Python
- How to teach the concepts

The course content is mapped to the subject knowledge requirements for entry into computer science teacher training.

It's not just the teachers who are learning, we are too! It is the first MOOC we have delivered and we're very open to feedback. We hope to respond and run the course again. The materials provided are just a start. We'd be very happy to include further resources so do let us know of any fantastic resources that you think we should be sharing on the course. Contact us via computing-cpd@warwick.ac.uk For more information about our teacher cpd visit http://goo.gl/q0I1SL *Claire Rocks*

OPEN UNIVERSITY SUBSIDISE UNDERGRADUATE MODULE

An OU introductory computing module is being offered at a subsidised rate to CAS teachers. SWITCHEDON talked to lecturer Neil Smith about 'My Digital Life' which gives a good grounding in a range of modern computing topics.

Why did you decide to offer the resource pack to teachers?

The new computing curriculum is a challenge for a lot of teachers. A common refrain is they feel unprepared to teach the new curriculum, but are short on release time (and CPD budget). When the new curriculum was being drafted, it was obvious that *My Digital Life* covered just about everything needed. That made us think how we could get this material in the hands of teachers. Signing up teachers on the module wouldn't work, as they aren't eligible for a loan to cover the £2500 fees. We hit on the idea of using the materials as a resource pack. We approached CAS, who were very keen to work with us to make the material available.

How does it differ from the taught undergraduate module?

For a start, it's not taught! There's no credit and it doesn't count towards accreditation. A reasonable chunk of taught time is spent developing study skills. None of that is relevant to teachers, who just want subject knowledge. So the content is all there and in a form we think is useful for teachers. The idea is that when you have to teach a new area, you can go to the relevant part of *My Digital Life* and bone up on it before designing your lessons. Making a resource pack keeps the price down because the teaching takes a lot of specialists' time. By stripping out all of that, we can offer the materials for pretty much the cost of printing, posting and admin time. It also covers the cost of the SenseBoard kit.

What's the programming like?

My Digital Life is an open-access module so anyone can study it, regardless of prior experience. This is very similar to what's happening in schools: both need to learn everything from first principles, and all get turned off when they're stopped by silly errors. All programming languages have the same problems of tricky syntax so we ended up settling on Scratch. As an environment for learning programming, it's great. The colour-coded blocks help people learn the differences between procedural steps, pure functions, and predicates (among others). The limited ways that the blocks can fit together means that it's impossible to produce syntax errors, and the fault-tolerant approach of Scratch means that logically wrong programs will do *something* rather than throw an obscure error and just stop. Adult OU students took the playful look of Scratch as an invitation to play with it, being very keen to dive in and start trying things out, which is exactly the response we wanted. There's not much research in this, but anecdotally people make the transition to other languages very easily. Scratch does a good job of showing people the basic principles, including how to break down problems into a form that a computer can solve.

Isn't Scratch too limited?

Yes and no. There's a surprising amount you can do in Scratch. Just look at the Scratch website to see the complexity of some of the projects. However, for very good reasons, Scratch is missing a few features that could allow more projects. Because we wanted that flexibility for our

HOW CAN WE TRAIN ALL THE TRAINERS?

One size doesn't fit all. As schools prepare to implement the new curriculum teachers will find they have a variety of needs. Through local hubs, Lead Schools and a growing number of Master Teachers, CAS can address some immediate requirements. But many teachers are wanting to engage in longer term training programmes. Many Universities are looking at what they can do to help and here we outline three early initiatives which have all been well received and cater for different requirements.

adult learners, we ended up making our own version of Scratch, called Sense. Scratch can't read or write files for example. I wrote a Sense program that breaks (simple) ciphers; it needs ciphertext to work on, so reading it from a file helps. Writing files is useful for recording what you've done in all sorts of ways. We can use it for logging data, for example. If you write the file as a set of comma-separated values, it's easy to open it in a spreadsheet for more processing and analysis. Another thing we did was make Sense network-aware. It can read data from any website you want and has special support for reading and parsing RSS feeds. It can write data to servers allowing programs to interact over the internet.

You mentioned the "SenseBoard kit." Why is that included?

The module puts post-PC computing centre stage. The trend is towards ubiquitous (or physical) computing, where we're surrounded by smart devices. Every student gets a SenseBoard kit. The tag line we use is "a ubiquitous computing lab in a box" and it's pretty accurate. It is a small interface board that plugs into a PC via USB. It's got some sensors and a bank of LEDs as output. Other sensors and a small stepper motor can plug in to the board so you can easily build devices that interact with the world. Pedagogically, it's great. There's something immediate and engaging about building a physical device. People really enjoy the making and immediate feedback of a working device. In a school context, there are lots of possible activities. Data loggers monitoring experiments, or the environment and writing the data to a file is an obvious application. The SenseBoard is dead easy to set up: just plug it into the USB port and away you go. Sensors just plug in the 3.5mm sockets. Sense works really well with it, and we're putting the finishing touches to a Python library that also connects to the board which will give teachers further flexibility in how it is used.

How can teachers get a copy of the resource pack?

We're limiting the resource pack to teachers only. That way we keep the university happy by not cannibalising our existing market. If you want the module, you apply to CAS who can check that you're really a teacher. Assuming you check out, CAS ask the Open University to contact you and they ask you all the questions about addresses, money, and the like. Once that's all sorted, the boxes will arrive in the post in a few weeks. More information is available at http://goo.gl/BAiIKC.

BCS/CAS PILOT CERTIFICATE IN COMPUTER SCIENCE TEACHING

A proposed Certificate in Computer Science Teaching, initiated by Sue Sentance has generated a lot of interest from teachers. CAS received 122 applications for the 50 places available on the initial pilot which starts in January. The intention is to give teachers recognition of their teaching competence in the Computer Science elements of the new Computing curriculum. In the next few years we anticipate that many teachers across the UK will become skilled in teaching Computing although this may not be the subject for which they have trained. This requires a good understanding of the domain of Computer Science and the development of appropriate pedagogical skills to teach the subject. The certificate, accredited by the BCS Academy of Computing aims to recognise this.

The certificate (post-pilot) will be launched in September. It is to be an evidence-based certificate rather than a course. The idea will be that you register, then within a certain time-frame need to submit three pieces of evidence to show:

- a commitment to updating your subject knowledge
- your competence in programming
- your understanding and application of pedagogical strategies suitable for teaching Computer Science

Teachers will apply to CAS to register. Once accepted, evidence needs to be submitted within 12 months. Further information can be found at http://goo.gl/IiLf4q. Detail may change given feedback from the pilot, due to complete by June so keep an eye on the CAS forum. The certificate will give recognition to competent Computer Science teachers willing to engage in significant professional development.

WOMEN IN ENGINEERING @ GOOGLE SERIES: CS ON AIR

One of the difficulties in encouraging more girls to study computer science is the lack of female role models. To help redress this in a scalable way, Google have produced a series of short (~15 min) "Hangouts on Air" with individual women engineers across Google. They are informal and conversational in style, talking not just about their work but also how they got into computer science and advice for others studying it. The interviews were broadcast live at plus.**google**.com/+**GoogleinEducation** and recordings are available to watch via Google's YouTube playlist. For a full schedule of interviews and other Hangouts on the subject of CS see the Education on Air website. The series is also available via a CAS resource.

Google supports a variety of initiatives and competitions. Now 11 years old, Code Jam 2014 will start on 11 March; code.google.com/codejam/. The Google Science Fair may also be of interest. An online competition open to students aged 13-18 around the globe, it covers a broad range of STEM subjects, including Computer Science. Last year there were three UK regional finalists, though sadly none made the final fifteen. The competition reopens in January. More details at www.googlesciencefair.com.

GIRLS INSPIRED BY A DREAM TRIP TO SILICON VALLEY

Joanne Devlin, Head of IT/Computing and Business at Lancaster Girls' Grammar School, wanted her pupils to see that women can be successful in this field. Last summer she arranged the experience of a lifetime.

At the end of July, 30 GCSE and A Level IT and Computing students returned from an inspirational trip to San Francisco and Silicon Valley, having followed a 10 day action packed itinerary. It started as an idea, and after embracing Twitter, I made many connections and managed to arrange visits to Qualcomm, Google, Salesforce and Stanford University. Pupils boarded their very own cable car on the Powell and Hyde line to experience the great city of San Francisco, cycled the Golden Gate Bridge and visited Alcatraz. Then it was on to Silicon Valley for an action packed 2 day tour of some top IT companies. Qualcomm showed pupils the Clean Rooms and how chips were made. Workshops allowed the girls to work with prototypes to see how accelerometers and cameras were being developed to work together in mobile telephones. All of the hosts were women, which was a key aspect of the trip, as I wanted the pupils to see and hear the experiences of real women living and breathing a career in the IT sector. One pupil, Naomi recalled, "All the women were enthusiastic and eager to get us involved in technology. They offered inspirational speeches which reduced some of us to tears. We couldn't have asked for more."

The Computer History Museum at Mountain View was fascinating and pupils had opportunities to see the evolution of the computer, bringing home to them how far this technology had come. One pupil exclaimed, "I can't invent the future if I don't know where we have already been". It was so satisfying to see ambitions developing. After a lunch with successful women in the IT field, arranged and sponsored by the British Computer Society the girls had a tour of Stanford University and then came the highlight of the trip, a lecture and workshop at Stanford University's Robotics Lab. As they entered the department the first robot they laid eyes on was Asimo the Honda robot. The excitement was fever pitch. Following a lecture by Stanford Professor Oussama Khatib, who explained the evolution of robotics, the girls were able to take part in a workshop to experience haptic technology and see the robots Romeo and Juliet in action. That evening Salesforce hosted us for a truly motivating evening in which pupils heard about the range of career options on offer from software development to User Experience. The girls had a truly amazing and unique opportunity; we hope it inspires them to consider the technology industry when selecting career options, a route of study where women can be very successful!

SHARING IDEAS AND CONNECTING WORLDS AT SCRATCH CONFERENCE

Sue Gray, Fakenham High School, joined CAS in 2011 because she wanted to change how ICT was taught. She had no idea there was an international dimension to her vague feeling of discontent. Two years on, she was sharing her experiences in Barcelona.

After joining the new Norfolk CAS Hub I was heartened to find there were others like me. The hub organised a Scratch competition. Great! I thought; this is what I wanted. It was really the first time I'd explored Scratch and I found it fun, engaging, stimulating and easy to work with. We didn't win, but pupils were immediately asking about the next event! Hub Chair, Neil Collins suggested I put together a proposal for a talk at the *Scratch Connecting Worlds* event, to be held in Barcelona. When I received an email confirming acceptance of my proposal my first thought was …panic! What was I thinking? Me, present at a conference!

From the opening Keynote with Mitch Resnick and Karen Brennan to the closing speech from Joek Van Montfort it was a roller-coaster of learning. So many talented people sharing their experiences; inspiring and delighting with their talks. Luckily, my 'slot' was in the morning of the first day, introduced by Joek Van Monfort. The presentation ranged from how we began with Scratch and the competition to my frustration at not being able to attract girls to Computer Club. I also covered data I discovered about the levels of educational achievement and income in North Norfolk which impact on our pupils. At the end Joek asked if there were questions and…they poured in! I was amazed! The *girl thing* had struck a chord with many in the room. I spoke briefly about the runaway success of our girls-only day and how wonderfully responsive the girls were without the boys around.

The three days flew past. Each morning there were workshops to sign up for and hard choices to be made! I learned about using the Picoboard with Margaret Low and her team of Tech Volunteers and had huge fun with Drew Buddie and the Makey Makey board. There were so many highlights but most of all I was struck by the number of women there. It was inspirational and I longed to be able to show the girls at school that being a geek is cool. On the second day we formed a human chain of 156 people to complete a circuit to a Makey Makey board and make the Scratch cat '*meow*'! I'm already plotting how we will change, improve, build on our experiences at work. I have learned so much, made some great new friends and been inspired to use Scratch in new ways to improve our learning.

Of course, it would not have been possible without the help of Neil Collins, Margaret Low at Warwick University and Clare Davenport at the British Computer Society (BCS) who helped organise a bursary from CAS, my Head teacher Matthew Parr-Burman and my husband who dug deep and helped out with expenses. It really was a brilliant experience. For more information see *Connecting Worlds* at www.scratch2013bcn.org

education innovation CONFERENCE AND EXHIBITION

CAS is running an advice clinic at the Education Innovation Conference & Exhibition (EICE) on the 27/28th February in Manchester. The free event is designed to offer advice, training and guidance when it comes to integrating innovation and technology into learning. This will come in the form of CPD training, inspirational keynote speakers, practical workshops, and an exhibition of over 100 solution providers including the CAS Clinic. EICE is thrilled to be working with CAS as an official supporter of the event. Simon Humphreys and Mark Dorling will be speaking to visitors about "change management" and "CPD training advice". The CAS Clinic, which will sit inside the main exhibition, will be a fantastic place for education professionals at any level to drop in and ask any questions about how to improve the way they adapt to the challenges of effectively teaching computing.

Confirmed speakers for the event include the head of ICT inspection at OFSTED, David Brown, Tim Rylands, Toby Young, Russell Prue and Vic Goddard as well as representatives from Jisc, DfE, Naace, ALT, EduGeek, Raspberry Pi Foundation and many more.

Event manager David Ventris-Field said: "Building on the success of last year's Education Innovation, we're focusing on how innovation can be used to raise achievement with lots of practical, hands-on workshops for teachers and lecturers and more strategic advice areas run by experts such as CAS". Education Innovation is proud to be co-located with the Raspberry Jamboree 2014, run by CAS member Alan O'Donohoe. The show is also co-located with the "iThink Therefore iLearn" conference. More details at www.educationinnovation.co.uk

TECHNOCAMPS INTRODUCES PLAYGROUND COMPUTING

Primary schools throughout South Wales are queuing up to host the Technocamps team for a day of fun, hands on computer science workshops, all carried out without the use of a computer. Playground Computing (playgroundcomputing.com) is a new primary school outreach project borne out of the successful Technocamps programme based at Swansea University. It is supported by a grant from the Digital Makers Fund, a scheme backed by Nesta, the Nominet Trust and Mozilla. The project focusses on three different topics, Hardware, Data and Programming and teaches young pupils conceptual computer science principles through a variety of activities, tasks and games. These lively workshops – run in the school's gym or (weather permitting) playground – explore how computers work, but mainly drive home the message that computer science is predominantly about problem solving, not boxes of electronic components.. They provide a platform for primary pupils to develop computational thinking skills, thus easing the transition into the emerging secondary school computing curriculum. Our aim is to visit a different school almost every day of the school year. Though intensive, this goal is proving easy to attain due to the popularity of the Technocamps initiative amongst Welsh schools. *Faron Moller*

Over 75 youngsters developed their own tourism app this past Summer as part of an innovative programme targeted at youngsters not in education, employment or training (NEET). Technocamps teamed up with Swansea City Council's Young People Services to deliver "*Push the Boat Out",* a five-week programme to boost confidence, communication and problem-solving. 15 youngsters were transported each week to a remote part of the Gower Peninsula, to compete in various challenges including app development, where they were trained in the use of the MIT App Inventor software. The groups came up with some fantastic, original ideas, and many went away with a new hobby and potentially a new lease of life to consider further learning and possible job prospects. Overall the summer residential workshops were a great success.

CAS IN NORTHERN IRELAND WELCOME THEIR NEW CHAIR

June saw the election of a new Chair, Dr. Irene Bell, to the NI hub. Irene is the head of STEM at Stranmillis University College, Belfast, a leading teacher training college and hopes to build on all the work done by Clarke Rice.

"Taking over from Clarke Rice is a privilege. We say a huge thank-you to Clarke and wish him all the best for the future". Monday 25th Nov saw the launch of the report "Teaching Coding to Pupils in KS2 and KS3 " and the CAS re-launch at Stranmillis University College. Every school has since been mailed information on joining CAS. The authors of the report, Dr Irene Bell and Ian Simons, were later awarded the 'Teaching Tech Award' at the recent NI Blackboard Awards, sponsored by Kainos, University of Ulster and Belfast Media (right). The report outlined the findings of a project designed to assess the way forward for developing computer science skills. The project saw students tackle html scripting. Results indicated that all participating staff found the work to be effective, giving real impetus to moving Computer Science forward.

CAS member Andrew Mulholland ran successful Raspberry Pi training and jam sessions with students and teachers. In the New Year Ann O'Neill will deliver further Pi sessions on the Stranmillis Campus. Ann's enthusiasm for Computer Science education is infectious and we look forward to her training programme in the next term. Another event is planned for March targeted specifically at KS2/KS3 pupils who may think a career in Computer Science is out of their reach. Details of these will be highlighted through the CAS community forum.

Pupils in KS4 and post-16 have not been forgotten. In N.Ireland the Department of Education have given Sentinus a contract to run a series of 30 one day workshops over the next 8 months. The aim is to encourage children to consider a career in Software Development and encourage the uptake of Computing 'A' Level. David Gault, a lecturer in Computer Science at Queens University Belfast, has designed the programme. The day starts with a problem solving ice-breaker, goes on to use Makey Makey boards as input devices, introduces Computer Programming using Processing, the final program being a version of 'Pong' which can be controlled by either keyboard or Makey Makey. A 'training the trainers' session ran in November and we are ready for future work with this age group. Mark Dorling, the National CPD Coordinator at CAS, will be visiting Northern Ireland later in the academic year to work with Irene constructing some workshops. If there is something specific that you would like to see addressed please contact Irene.

CAS SCOTLAND ANNUAL TEACHER CONFERENCE DOUBLES IN SIZE

Glasgow University hosted the second annual CAS Scotland teacher conference on 26th October, with over 200 teachers attending. Mark Tennant reports on a packed day of workshops, discussions and keynotes.

The first of two morning keynotes was delivered by Professor Sally Fincher from the University of Kent. She talked about pedagogical content knowledge being situated and not easily transferable. She discussed the setting of farms, where farmers come together to bare all and get honest feedback on their crops and farm management. Sally then talked about kitchens and how difficult it is to transfer your knowledge about how to cook a particular meal. Different recipe formats have been tried, including Jamie Oliver's 30 minute meals and an engineers recipe, expressed as a table. Lastly she related this to difficulties in explaining how you teach a particular Computing lesson or topic stressing the importance of having a dialogue with other teachers about pedagogy.

Next up was Quintin Cutts from Glasgow University. Now seconded as one of the project officers for the PLAN-C CPD programme (Professional Learning and Networking for Computing), he gave delegates an overview of the programme and the progress made so far. The link to Sally Fincher's talk was certainly obvious: professional learning communities are a big part of the programme.

A series of practical workshops and seminars followed the keynotes with a wide variety of topics: Raspberry Pi; Touch Develop; Lego Mindstorms; Programming with LiveCode, Kodu & Python; there was a choice of 25 sessions in total. For the first time there was also a focus on Computing Science education in the Primary sector, and we were delighted to be joined by speakers focussing on this: Primary teachers Bianca Ní Grógain flew in from Ireland to discuss flipped learning in the primary setting and Karl Barrs from Fife demonstrated his use of Scratch in the primary classroom.

The afternoon sessions finished with two further keynotes. In the first, Alistair O'Brien from the Amor Group gave us his view of 'what industry really wants'. The answer was clear: students who have computational thinking skills! In the second keynote, Yen Yau and Hugh MacDonald demonstrated the need for Computing Science in the creative industry. Hugh's demonstration of 3D rendering techniques in blockbuster movies might just have contributed to the CS teacher shortage in Scotland, judging by the comments on twitter!

The day was wrapped up with a wine reception and a chance to chat, with thanks to NESTA for sponsoring the wet stuff. Thanks are also due to the CAS Scotland committee who organised such a large event on top of their regular jobs. Here's looking forward to next year!

A special 8 page conference issue of SwitchedOn Scotland was rushed out for the conference, featuring the launch of the PlanC cpd project. Download a copy from www.casscotland.org.uk/

CURRICULUM CHANGES AFOOT IN NEW ZEALAND

Computing has undergone a complete metamorphosis in the last five years. A radical overhaul has occurred, coinciding with the development of a robust Teachers Association, NZACDITT. We have seen the blending and development of teacher capability, programme delivery, and ability to use different aspects fluidly across the curriculum. Web Development, for example now includes aspects of graphic design and manipulation, server side scripting and database management, rather than putting pictures in FrontPage. The new content is still being rolled out but there have been many early adopters. Thousands of students have taken up the new standards. Numbers taking specialist CS topics indicate the pipeline is actually expanding! The first wave of students will leave school at the end of 2013, and tertiary institutions are preparing for this new group. The universities have been getting on board to develop content and provide professional development to upskill teachers, much of which has been made possible through CS4HS grants from Google.

The rapid change has been a challenge, as there has been a lot of new content and pedagogy to grapple with. Teachers would have liked a lot more government led Teacher Professional development. Changing the course models has also been a challenge, as traditionally computing classes have been regarded as an "easy" option. The mode of assessment was new, largely done by projects and not exams, and this requires considerable thought to make it work well. The saying "Good things take time" still holds true, but many highly capable students are now studying computing at school instead of shunning it.

Gerard MacManus , John Creighton
President & Past Pres, NZACDITT

A PAUSE FOR THOUGHT

Do you know how online stores can tell you what book you'd like, what film to watch and what track to download? What about online dating services or automated stock trading? Algorithms have the power to change our world; their accuracy and reliability make them superior to the flawed inconsistency of humans. As our world becomes increasingly automated, and we inspire our children to think computationally and apply logic to solve real-world problems, do we see a role for humans in the process? Twitter's 'trending' algorithm shapes our social interaction, while search engines' auto-complete and search algorithms shape our browsing (who ever goes to Page 2 of search results). But how do these algorithms determine their results, and would we agree with their judgments?

Understandably the companies using those algorithms to identify 'trends' and 'relevance' keep them under wraps. This helps avoid manipulation, but puts us in a position where we rely on algorithms to select and determine what is most relevant to us with such materials becoming almost canonical to end-users. As algorithms emerge to predict the likelihood of a prisoner reoffending and myriad other fascinating tasks, Evgeny Morozov, author of 'To Save Everything, Click Here', explores whether there is a need for 'algorithmic auditors'; such trusted representatives could delve into the code to spot ethical, political, and objectivity issues, to evaluate the moral judgments of algorithms. As our entire corpus of data becomes digitized, and the minutiae of our lives is transferred online, before we devolve responsibility entirely to algorithms should we consider who guards the guards? *Lyndsay Hope*

ARE YOU REGISTERED TO KEEP RECEIVING CS4FN?

Issue 16 of Computer Science For Fun (cs4fn) is out! It explores almost 20 different stories about language and computing. You can read all the stories at www.cs4fn.org/magazine. The core funding for cs4fn has changed – they are now funded by the Mayor of London and the DfE. They still hope to find ways to continue to send class sets of magazines and other teacher resources to school subscribers throughout the country. To receive any future cs4fn mailings you'll need to subscribe – for free! It's really easy to do this. Just visit www.cs4fn.org/teachers, fill in your details and you'll get any future mailings of magazines and resources addressed directly to you. On the website you will always find everything cs4fn has produced since it begin in 2005. That includes 16 cs4fn magazines, two magic books, a special on women and computing, another on biology, classroom activities and over 350 articles on the fun of computing. And there's a lot more to come over the next couple of years, as they help teachers navigate the new computing curriculum. Tell them what you'd like to see at cs4fn@eecs.qmul.ac.uk.

CAS will be holding a series of Training Days (in association with Rising Stars), aimed at primary teachers. Regional Conferences are also planned (dates as shown) but keep a lookout for further announcements on CAS Online.

CAS PRIMARY TRAINING DAYS

University of Exeter	28th Feb
Nottingham Trent University	12th March
Oxford Brookes University	26th March
Manchester Met University	28th March
University College London	31st March
University of Cambridge	2nd April

CAS REGIONAL PRIMARY CONFERENCES

Southampton University	2nd April
Newcastle University	2nd April

COMPUTING AT SCHOOL

EDUCATE · ENGAGE · ENCOURAGE

Computing At School was born out of our excitement with the discipline, combined with a serious concern that students are being turned off computing by a combination of factors. **SWITCHEDON** is published each term. We welcome comments, suggestions and items for inclusion in future issues. Our goal is to put the fun back into computing at school. Will you help us? Send contributions to **newsletter@computingatschool.org.uk**

Many thanks to the following for help and information in this issue: *James Abela, Tom Attfield, Ilia Avroutine, Tim Bell, Irene Bell, Jonathan Black, Neil Brown, Guy Coop, Nick Cook, John Creighton, Claire Davenport, Roger Davies, Joanne Devlin, Peter Dickman, Sue Gray, Mark Gudzial, Stuart Hadfield, Lyndsay Hope, Toby Howard, Steve Hunt, Bharat Jogia, Neil Kendall, Catriona Lambeth, Leng Lee, Gerard MacManus, Faron Moller, Emma Partridge, Claire Rocks, Chris Roffey, David Rydeheard, Sue Sentance, Janu Shan, Andrew Shields, Neil Smith, John Stout, Chris Swan, Mark Tennant, David Ventris-Field, Sarah Zaman.*

www.computingatschool.org.uk

Computing At School are supported and endorsed by:

bcs The Chartered Institute for IT
Enabling the information society

Microsoft Research

Google

CPHC The Council of Professors and Heads of Computing