

Introduction: Computing unplugged

Teaching ‘Computing’ is now a fact of life for all schools in England, and soon to be the same across the UK and elsewhere. It is a subject with lofty goals: *a high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world*. (Programme of Study, 2013, <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>).

This book is about a way of teaching computing without using computers. Each of the chapters shows how central concepts in computing and computational thinking can be explored and developed away from technology. Each chapter is based around a different theme and shows how thinking like a computer scientist can be illustrated by everyday activities, and how it can help us solve problems in many different situations. This approach should give children and teachers a deeper understanding of computing, and how and why it can be applied to more than just programming. The lesson ideas model how computational thinking can be embedded across the curriculum.

Computing

Computing is a subject with three strands: digital literacy, information technology and computer science (Shut Down or Restart, Royal Society, 2012, <https://royalsociety.org/topics-policy/projects/computing-in-schools/report/>). Digital literacy is about creating, evaluating and using digital artefacts. Information technology is about how all these computers work, work together, and can be made to work. Computer science is about the essence of what computing *is*: what can be computed, and how to think about problems and solutions with a view to applying them to computers. In this sense, it’s not really *about* computers: rather it’s about how to think. Therefore, one key aim of teaching computing is to about getting children to think about the world in a different way.

Viewed this way, we can see that ‘computing’ isn’t about computers in the same way that ‘science’ isn’t about test tubes and ‘art’ isn’t about paintbrushes. Computers, test tubes and paintbrushes are all vitally important tools we use in these different disciplines, and we have to understand them and how to use them effectively. But there’s much more to art and science than just the tools, and there’s much more to computing than just computers and programming.

This book is an approach to the discipline of computing that does away with computers. By removing the distraction of computers, we can instead concentrate on the essence of computing and learn how to apply computational thinking techniques to a range of different problems.

Computational thinking

When the current Computing curriculum was introduced, everyone was very exercised by ‘programming’ and ‘coding’. This, people thought, was the key skill required by the new curriculum. However, just understanding the particular syntax rules of a particular programming language is neither interesting nor particularly useful on its own.

Programming is much like bricklaying: you need to know a little bit about it to be able to build something, and master artisans can have long and detailed conversations about pointing and the relative advantages of variations on herringbone layouts, but fundamentally a wall is a wall. Bricklaying isn’t that interesting: *architecture* is interesting. Just as architecture is about understanding people’s requirements and seeing how a particularly shaped pile of bricks could address them, *computational thinking* is about understanding a problem and seeing how a particularly shaped pile of program statements could address it.

Computational thinking has many definitions. Jeanette Wing (2006, Computational thinking *Communications of the ACM* 49 (3): 33. doi:10.1145/1118178.1118215)

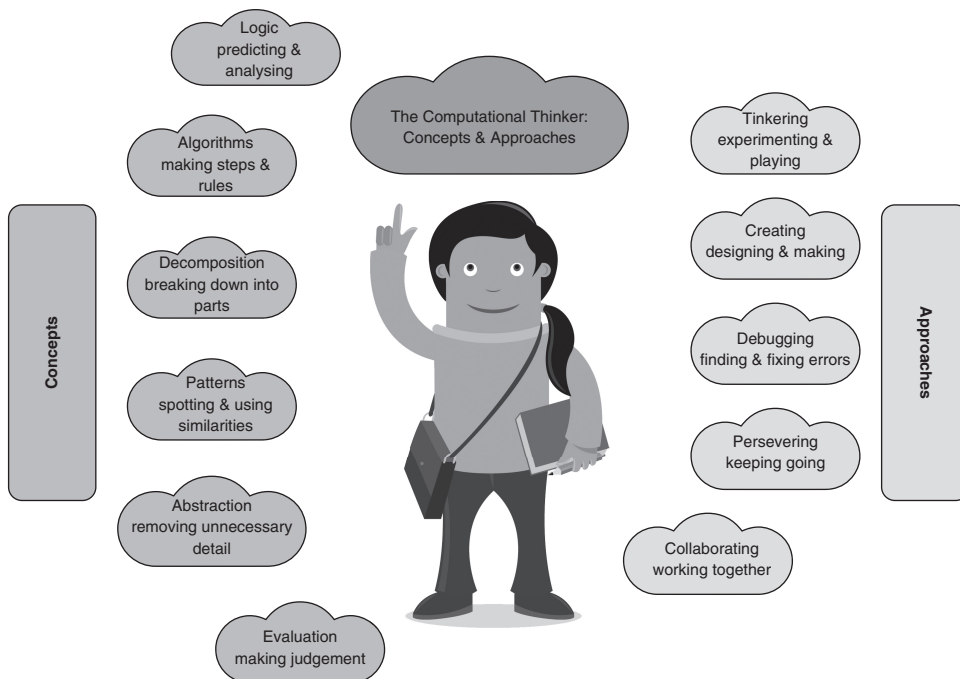


Figure i The computational thinker

Table i Six concepts and five approaches

Concepts: key ideas and methods of thought	Approaches: ways of approaching, tackling, and solving problems
<i>Logic</i> : thinking in terms of rules, applying them to situations, and getting a result.	<i>Tinkering</i> : trying things, seeing what works. Exploring.
<i>Algorithms</i> : defined sequences of steps that achieve some result.	<i>Creating</i> : making something new and innovative, but purposefully.
<i>Decomposition</i> : breaking a problem down into smaller steps, and solving each in turn.	<i>Debugging</i> : spotting mistakes and fixing them.
Finding and using <i>patterns</i> : seeing how a previous solution or approach can be applied here.	<i>Persevering</i> : not giving up when the first attempt fails.
<i>Abstraction</i> : finding what's important, discarding the unimportant, to keep attention on what really matters.	<i>Collaborating</i> : working together, sharing ideas, and building something together.
<i>Evaluation</i> : not only deciding if something works, but deciding what 'works' means in this situation.	

resurrected and redefined the term in 2006 as 'Ways to Think Like a Computer Scientist'. Her idea was elastic and stretched to many definitions. In this book, we use the definition of computational thinking defined by Barefoot (2014) (<http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/>) and *Computational thinking – A guide for teachers* (Csizmadia *et al.* 2015, <http://community.computingatschool.org.uk/resources/2324>)

This is a constellation of six concepts and five approaches. As children develop and master these concepts and approaches, they are better able to grasp a range of problems and develop robust, efficient solutions to them. Very often, these solutions won't involve computers, in much the same way that science is more about a way of thinking and understanding (the scientific method) than any particular fact.

A quick look at these concepts and approaches shows that they are generally useful and applicable across a range of subjects and activities. Computing is a place where these ideas come to the fore, but it's not the only place where they're useful.

This book is about computational thinking. It's about developing these computational thinking skills in students and helping them apply the skills in different situations across the curriculum.

Becoming unplugged

When computational thinking is the central activity in computing, the computers themselves become just a tool for carrying out the processes we've designed elsewhere. From this point of view, the computer can become a distraction. Getting the technology to work can be a fiddly and time-consuming process: we have all at some

point spent far too long tweaking the layout of a document to make it look just right rather than doing something productive.

It is much more difficult to learn a new skill such as computational thinking while dealing with unfamiliar and recalcitrant technology. We have to expend mental effort to practise the new skill while simultaneously trying to operate a device. This combination can easily overwhelm the best of us.

What can be much more useful for learners is to step away from the technology and practise the computational thinking skills in a different setting. If the physical trappings of the new setting are simple and well-known, such as pencil and paper, beanbags and hoops, we don't need to devote any energy to making them work and can instead focus on the new skills and knowledge we are developing: the computational thinking skills and any subject-specific knowledge and vocabulary. Once we've acquired the new abilities, we can apply them to a computing technology, confident that we know what we are trying to achieve and how to go about it; the only difficulty is inherent in the technology, not in us.

Unplugged activities also engage learners in different modalities. Very often, we interact with computing devices by sitting still, looking at a screen, and manipulating keyboard, mouse, and touchscreen. This emphasises visual, logical and mathematical modes of reasoning but leaves out many other modes. Children, especially young children, find it difficult to sit still for long periods, even more so when they're restricted to just a few modes of interaction. The activities in this book explore a much wider range of activity, including musical, kinaesthetic, artistic, intrapersonal and interpersonal (see Gardner, 2011). This variety should assist you with engaging with all the students in your class and maintaining that engagement.

Pedagogy

The playful nature of many of these unplugged activities emphasises the constructivist pedagogy that underlies this approach. Away from the computer, and the performance anxiety and stress it could cause, learners are free to experiment and explore the activity and the thinking that underlies it. This will help them construct their own understanding of the topic, assisted by suitable scaffolding from the teacher. The physical nature of many of the unplugged tasks results in learners making some artefact that either embodies or represents their learning, such as a written-out, debugged algorithm or even a bowl of fruit salad, thus supporting both Papert's constructionism and Piaget's constructivism (see Ackerman, 2001).

The collaborative nature of many of the activities encourages social constructivism, with learners developing, testing, and refining their ideas together in a social setting (Vygotsky, 1980).

By definition, computing unplugged develops skills in one context (unplugged) that must be applied to another (plugged, using the technology). While this requires more

of the learners than simply learning a new skill to be applied in the same context, it should develop more metacognitive awareness in learners as they apply existing learning to a new context and talk about the transfer of skills.

Computing, despite its abstract and mathematical basis, is a strongly practical activity. Going back to an earlier analogy, one does not become a competent bricklayer from just reading about it: competence only comes from direct activity. All the approaches of computational thinking are active ones, requiring engagement and interaction by the learner. Most of the activities in this book have a strong experiential flavour, with the learners moving through Kolb's (2014) experiential learning cycle at least once in the activity.

Cross-curricular links

As we said earlier, computational thinking is a general approach to thinking about and tackling problems. While it is most apparent and distinctive when it comes to computing, the same concepts and approaches are present in many disciplines and subjects. We are seeing this in the wider world, with computers and smart devices moving into many areas from fitness tracking and photography to romance and autonomous transport.

Computing can apply to other domains in two different, but complementary, ways. The digital literacy strand of computing is about the ability to use technology tools to assist the normal practice of a discipline, such as providing new tools for drawing and manipulating images. This is one way that people think computing can be applied to the world.

But computational thinking provides a different approach. It can be used as a lens through which to view a subject, such as understanding a piece of art by crafting an algorithm that can create it, or finding a new recipe for fruit salad by finding the common and distinct elements of a range of existing ones.

Here are a few cross-curricular examples:

- *Numeracy*: many standard calculations (addition, division) can be expressed as algorithms. Patterns can be found in relating things like properties of shapes. Co-ordinates can be illustrated by making 'robots' move (see Chapter 1).
- *Literacy*: algorithms are often written as sets of instructions, which must be clear and precise. Phonics have rules for how sounds are written and how letters make sounds.
- *Art and Design*: art often abstracts details away from life to create a representation. Making complex pieces can be decomposed into a sequence of steps.
- *Science*: performing experiments requires following algorithms. Making predictions and drawing deductions from them is making patterns and involves logical thinking.

- *Geography*: algorithms appear as ways to find, communicate and follow directions. Grid references are co-ordinates and, again, can be used to move people or objects around a map.
- *History*: the small details of a period of history can be abstracted away to see the broader sweep, with events grouped into periods.
- *PE*: rules of sports are full of algorithms and decision-making processes, which can be drawn out as algorithms. Dances are sequences of movements and can be analysed similarly to songs (see Chapter 3).
- *PSHE*: We all follow algorithms for staying healthy, follow routines when entering or leaving school, and so on. When things go wrong, the processes can be debugged to make them better or find where problems arose.

About this book

Each chapter in this book explores how computational thinking relates to a different activity. The topics were chosen to show the range of applications of computational thinking, and to draw out important aspects of it. Above all, the topics are meant to be fun and engaging.

The activities throughout the chapters in this book show a large number of ways that computational thinking can be used across the curriculum, such as through drama, the outdoors, art, music, puzzles, games and practical hands-on activities. We suggest ways of tackling computer science concepts in a collaborative way by taking on the roles of robots, musicians, artists, explorers, code breakers, magicians, gamers, cooks and scientists. This demonstrates that computational thinking techniques can be applied across curriculum subjects so that they are embedded as a creative problem-solving tool.

Each chapter follows a similar format. It starts by outlining the key computational thinking aspects that are developed and explored in that chapter, including key terms and concepts used. There are three classroom activities presented as outline lesson plans that give concrete examples of how the concepts can be explored in a classroom setting. Sections discussing the computational thinking aspects in more detail and inviting you to reflect on other ways they can be incorporated into your teaching follow these. There is some discussion of pedagogical strategies for implementing the activities across a range of age groups and abilities, along with suggestions for extension activities and for follow-on 'plugged' activities using a range of free online resources. Examples are given of how the concepts are relevant in real-world situations such as medical devices or weather forecasting. This reinforces the idea that a computing lesson or set of lessons might introduce concepts through an unplugged activity, apply them via some 'plugged' digital making and also give consideration to their use in real-world contexts.

Resources

Computing in the national curriculum: a guide for primary teachers. www.computingschool.org.uk/data/uploads/CASPrimaryComputing.pdf

Computing in the national curriculum: a guide for secondary teachers. www.computingschool.org.uk/data/uploads/cas_secondary.pdf

Computer Science in a Box: Unplug your curriculum. A set of seven unplugged lesson ideas from the National Center for Women and Information Technology (ncwit.org) www.ncwit.org/sites/default/files/resources/computerscience-in-a-box.pdf

References

Ackermann, E (2001) Piaget's constructivism, Papert's constructionism: What's the difference. *Future of learning group publication*, 5(3), 438.

Gardner, H (2011). *Frames of mind: The theory of multiple intelligences*. Basic books.

Kolb, DA (2014) *Experiential Learning: Experience as the source of learning and development*. FT press.

Vygotsky, LS (1980). *Mind in Society: The development of higher psychological processes*. Harvard University Press.