Review to **Achieve**
Educational Excellence
in Australian Schools

Public submission made to the Review to Achieve Educational Excellence in Australian Schools

Submitter: Computerbasedmath.org
Submitting as a: Other (Private company)
State: Other (UK)

Summary

Integrating modern, computer-based computational thinking into the school curriculum is the single most significant change to achieve educational excellence in Australian schools for STEM and beyond.

It is the right route to motivating students to acquire these vital problem-solving skills which are life-changing and uplifting for them. It is the right route to help boost the Australian economy in an artificial intelligence age.

As CBM have demonstrated, there is a proven approach to create a tightly bound curriculum and digital content that will deliver computational thinking education. It is as close to real life as possible and therefore problem-centric and computer based—the computer undertaking calculations whilst leaving the student to undertake the creative and challenging aspects of problem solving.

Australia is well poised to adopt such an approach with the modern “curriculum cube” and well-developed programmes involving computer technology. Although there are always risks of change, today’s risks of inaction in this area are rising and need to be weighed carefully.

Main submission

Submission: Review to Achieve Educational Excellence in Australian Schools

Fundamentally changing mainstream technical education for the computer age.

About Wolfram: For nearly 30 years, Wolfram has been at the centre of mathematics and computation worldwide in more ways than any other organisation—as employers, suppliers of technology and users of mathematics for creating technology, with customers including the world’s companies, governments and universities. Wolfram is sometimes credited with having strongly contributed to the increased use of mathematics or more widely, computation in the real world during the last few decades. It is out of this uniquely broad basis for understanding the world's mathematics that our views have been formed.

About CBM: Based on this uniquely broad perspective, Computer-Based Mathematics (CBM) was formed to lead a fundamental rethink and redesign of STEM subjects' curricula so they truly reflect today's real-world subjects with their vitally important applications

---

Introduction and case for fundamental change

At no time in history has new machinery threatened to take over from humans as it does now. Previous eras of mechanisation have been largely confined to replacing then scaling up physical activities. Instead, computers are continuing to take over intelligence- and knowledge-based activities—areas previously considered quintessentially human. How should education react? Do we still need to learn skills that computers now perform? If not, what should we learn instead?

It is strongly our view that building on new powers of automation and enabling humans to go further and take on new challenges is the urgent priority, not trying to continue to do tasks that compete with those powers. This means learning to handle harder, more complex problems earlier (to mimic growing complexity in the real world) as well as gaining experience of managing and interfacing with our new machinery (computers and artificial intelligence [AI]). It also means jettisoning most of the skills that computers take over.

In mainstream school education, mathematics is starkly at the centre of this issue: it is the core technical subject—and curricula everywhere still retain hand calculating as their focus. Yet in the real world—where maths skills are so coveted—almost all calculating is by computer, adding much more conceptual complexity and very different approaches for which students are today ill-prepared. Today's school maths is perhaps 80% content that will not be used outside education, and however well that subject is taught with whatever IT provision in its pedagogy, it will still fail to match what is now needed. It is therefore simple to state but harder to manifest the fundamental change that is needed: use computers in schools as we do in the real world, replacing most hand calculating with harder, computer-based problem solving. Invest the time released from hand calculating in to complex problem solving and critical thinking.

Australia needs to make this radical change to its mainstream technical school subject—today called mathematics—as well as the broader STEM curriculum if it is to enable the next generation of children to grasp the opportunities of the AI age. Individually and societally, future economic success will be squandered if the full risk of inaction is not appreciated and weighed against the risks of change. What is more, Australia is at a good position on the starting grid for the change we propose, with a modern, practical outlook on education, the forward-looking “curriculum cube”, and many states with modern technology in the classroom (including Wolfram’s).

In this response, we will be not be addressing all aspects of the Australian F-10 curriculum. Instead we will concentrate on describing how steps can be taken to improve student outcomes and school performance in the acquisition and application of computational thinking skills and knowledge across the curriculum.

What should educational success for Australian students and schools look like?

Computational thinking is key in an AI age, not only as an everyday requirement but also as a crucial skill for top performers. Educational success would mean the widest range of students are empowered to problem-solve computationally across a full range of complex real-world problems, whether they are focussed on STEM subjects or not. The complexity of problems and involvement of computational technology takes us further and further from hand calculations, and this effect will only intensify. Future success means optimising the joint competence of humans and computers to get answers, using the process of computational thinking.

As Jennifer Westacott from the Business Council of Australia stated recently, “We believe school graduates … should be prepared for the world of work–not just a single job or a single employer–carrying the life skills of adaptation, resilience and self-awareness. They need a foundation for future learning and the grounding to become engaged citizens in our society. In short, we should be equipping children for life, not just for sitting tests.” (Address to the National Press Club: Future-Proof, Oct 2017.)

Optimally equipping students with computational thinking is perhaps the most fundamental change that could prepare young Australians for the future world of work.

What is computational thinking?

We define computational thinking as a mode of thinking in which you apply a rigorous and repeatable problem-solving process to ideas, challenges and opportunities. Although common use of the terminology is relatively new, we would argue that computational thinking approaches have been widespread and spectacularly successful across a range of science, technology and business problems in the real world, driven by the rise of computational technology. Driving this success further needs not only better specialists in computational thinking, but everyone to be able to apply it at a much greater level than they can today. It needs to be used across all subjects, rather as English is. For example, a computational approach to history should be part of learning history. Mathematics could be the core computational thinking subject, but is not hitting the mark—or a new subject could take on this challenge using elements of mathematics and coding with a new approach.

Here's how computational thinking works; it is a four-step process:

1. DEFINE
2. TRANSLATE
3. COMPUTE
4. INTERPRET

These steps are set out below:

DEFINE—Start by defining the question that you really want to address—a step shared with most definitions of "critical thinking".

TRANSLATE—Computational thinking follows question definition with a crucial transitional step 2 in which the question is translated into abstract computational language—be that code, diagrams or algorithms. This has several purposes. It means that hundreds of years' worth of concepts and tools can be brought to bear on the question (usually by computer), because you have turned the question into a form ready for this high-fidelity machinery to do its work. Another purpose of step 2 is to force a more precise definition of the question. In many cases, this abstraction step is the most demanding of conceptual understanding, creativity, experience and insight and is least effectively covered in today's curricula.

COMPUTE—After abstraction comes the computation itself, where the question is transformed into an abstract answer—usually by a computer.

INTERPRET—The abstract answer is transformed back to the real world through interpretation of the results, re-contextualising them in the scope of the original question and sceptically verifying them.

The process rarely stops at that point because it can be applied over and over again, with output informing the next input until you deem the answers sufficiently good. This might take just a minute for a simple estimation or a whole lifetime for a scientific discovery. A visual representation of this can be seen at www.computerbasedmath.org/maths-process-poster.

Whilst emphasising the process end of computational thinking, its power of application comes from (what are today) very human qualities of creativity and conceptual understanding. The "magic" is in optimising how process, computer and human can be put together to solve increasingly tough problems. Rather than spending a large proportion of curriculum time devoted to teaching hand calculation and algebraic manipulation, students would be better served by allowing this time to be spent on defining, translating and interpreting the problem and solution, optimising the human, real-world part of the computational thinking cycle.

There is another key difference between a traditional maths way of thinking about a problem and modern computational thinking: the cost-benefit analysis between the four steps of the process.

Before modern computers, step 3—computation—was very expensive because it had to be done by hand. Therefore you would try very hard to minimise the amount of computation at the expense of much more upfront deliberation in question definition (step 1) and abstracting (step 2). It was a very deliberate process. Now, more often than not, you might have a much more scientific or experimental approach, with a looser initial question for question definition (like "Can I find something interesting in this data?") and an abstraction in step 2 to a multiplicity of computations (like "Let me try plotting correlation of all the pairs of data.") because computation is so cheap and effective you can try it repeatedly and not worry if there is wastage at that step. Modern technology has dramatically shifted the effective process.

Outcomes, measurement and relationship to Australia’s curriculum capabilities

One of the key challenges is defining, clearly and explicitly, the educational outcomes that drive and complement the students' learning. We have mapped the required skills of problem-solving as identifiable outcomes within our computational thinking cycle www.computerbasedmath.org/outcomes; we have begun to map out the primary contexts too.

The Australian Curriculum describes its outcomes using three dimensions:

- Skills, knowledge and understanding of subjects
- Skills, knowledge and understanding of general capabilities across the subjects
- Skills, knowledge and understanding of cross-curriculum priorities

Computational thinking, as a broad, cross-curriculum paradigm, interleaves the current three dimensions on a number of levels and provides a mechanism for delivering success to the widest range of students. If implemented correctly, it is accessible to the full range of student abilities through well-chosen, motivating and interesting problems. It does not disadvantage the student motivated by pure mathematics, as the problems are open-ended, realistic and can be highly conceptual; it has genuine value, from humanities applications right through to STEM.

Our outcomes heavily support the Australian Curriculum’s set of seven general capabilities by adding detail for core computational thinking.

A few examples:

- Literacy: There is greater emphasis on reporting and communicating solutions in CBM. Presenting results and interpretations are a key part of the CBM approach.
- Numeracy: Mental arithmetic is often exercised, but with higher-order mental imagery too. For example, how proportional change will be manifested, how exponential relationships behave and how dimensions can be visualised, but in far more complex cases than today's curriculum.
- Information and communication technology (ICT) capability: Core to CBM is that computers are intrinsic to computational thinking, not just to support pedagogy of today's subject. Knowing how to apply the wide range of capabilities of computers for computation, e.g. machine learning, coding, emphasising data science of large datasets, managing processes of human-computer interaction and communicating findings with interactivity.
- Critical and creative thinking: The outcomes of CBM and computational thinking overlap strongly with the descriptors of critical and creative thinking but further include manifestation with technology.
- Personal and social capability: Working in teams and managing oneself are skills developed in all CBM materials. Collaboration is a common thread throughout.
- Ethical understanding: This is covered by many of the topics in the STEM modules, including building in questioning of computational results. The learning is done in context, and is not solely about the mathematics, so students have to consider ethical issues when interpreting and understanding their results.
- Intercultural understanding: This is developed at a number of different levels, including the universality of maths and computation across all cultures as a common language, diverse collaboration on tasks and the pooling and analysing of data from dispersed groups.

However a curriculum is structured, learners will progress better if they are engaged in their learning. CBM makes this engagement easier by removing the need to learn complex, out-of-context calculation skills by hand. Instead the learning can be framed within problems that are of local relevance, easily adapting a problem from one context to another. (Example: "Are girls better at maths?", where two datasets are compared, might become "Which state has the most expensive houses, NSW or ACT?".) Being able to frame problems in local contexts enables a computational thinking curriculum to support the three cross-curriculum priorities, generating personalised learning contexts that can have relevance to all Australian communities, including Aboriginal and Torres Strait Islanders.

What can we do to improve and how can we support ongoing improvement over time?

In the narrative above we have emphasised what needs to change for success and why.

The challenge then is how best to deliver change and continuing improvement through a computational thinking educational strategy.

Fundamentally, there are two options:

- Introduce a new computational thinking subject.
- Integrate computational thinking into STEM and broader curricula.

Whichever approach is most effective, what is initially needed is a radical showcase of how a CBM approach can drive the agreed-upon principles of the Australian Curriculum. This could be achieved initially by picking key stages in the curriculum to run the radically new subject in a way that builds upon more traditional work beforehand and prepares for today's subsequent curriculum. If this showcase is successful, there will be pressure to drive it to other years of the curriculum and adjust assessments.

The approach outlined is contrasted with trying to implement a less radical change agenda across a wider range of the curriculum at the start. Because of how fundamentally different the new subject matter needs to be, this broader approach is unlikely to cause the vision reset that will really deliver results.

A key question is how to manifest delivery of a new, computer-based subject. Our approach has been to build interactive student-teacher materials which collectively, through association with agreed outcomes, mark out the curriculum. This contrasts with traditional approaches of drawing up a curriculum specification and building materials such as textbooks based on it, and was adopted for several reasons. Firstly, to make problems realistic. Even with our expertise, it is hard to predict upfront which areas of maths will be utilised, and if you lock into these upfront, the problems become contrived. Secondly, it is more cost effective overall. Thirdly, it very directly manifests to stakeholders—students, teachers, parents—what the new subject entails. Fourthly, we have found that interactive student-teacher materials boost engagement of both teachers and students. In order to achieve optimum contextual and cultural resonance, co-design with local educators and key stakeholders is worthwhile.

Problems we have selected have been drawn from across the STEM curriculum, built around topics as diverse as how to win a bicycle race, marketing the "best" mobile phone, controlling a quadcopter or deciding whether boys are better than girls at mathematics! The aim should be to choose problems which will:

- Be as realistic as possible, similar to problems they will actually face
- Be accessible with minimal preamble to motivate students to enjoy mathematics and want to learn more
- Build mathematical skills by introducing increasingly complex concepts, rather than increasingly complex procedures
- Build an understanding of and competence in using an iterative four-step problem-solving methodology that has broad applicability
- Give students as broad an experience as possible of today's mathematical tools (e.g. machine learning)
- Develop complementary broad-based coding skills
- Address a rather different set of mathematics outcomes than has been seen in traditional mathematics education

The Australian Curriculum’s Technologies curriculum has made a recent first step into specifying the nature of coding skills to be learned. This can be built upon and extended from the initial requirements within the realms of data science and user interface to have a new emphasis on the general capabilities required of students. We would recommend a change to the critical and creative thinking capability to include the use of computation to provide a commonality between the two brands of thinking—using computation for critiquing and using computation for being creative whilst solving problems. The two are different, as described in the introductory description, but being able to apply computational thinking provides a common basis for both.

We think a core subject that teaches computational thinking is the most likely route to success. That is because computational thinking needs knowledge of what is possible, experience of how you can apply it and know-how of today’s machinery for performing it. You need to know which concepts and tools there are to translate and abstract to computational language. We do not think you can only learn this in other subjects; there needs to be an anchor where these modern-day basics (learnt in a contextualised way) can be fostered.

Are there barriers to implementing these improvements?

Delivering a computer-based curriculum primarily through statements of skills, knowledge and understanding is problematic because translating these statements into classroom experience has particular complexities for a new computational thinking curriculum. To enable the delivery of a new subject that is beyond the experience of many career teachers, materials will need to be produced that both deliver the subject in its intended manner and also provide the teacher with detailed guidance on how to deliver it, both in pre-planning and real time. CBM has evolved a well-received method for doing this, combining the student materials with intended outcomes, stages in the problem-solving cycle and teacher support.

A key change required to measure the success of a computational thinking curriculum will be to allow the use of computers during any student assessments that take place. A large proportion of the outcomes we have listed can be assessed without a computer, but to get the complete picture of a student's ability to be creative and solve problems independently, a computer needs to be available. The pilot of VCE Mathematical Methods CBE in Victoria has shown that it is possible to assess similar skills using isolated computers during timed sessions.