DOI:10.1145/2556937

Tim Bell

Education

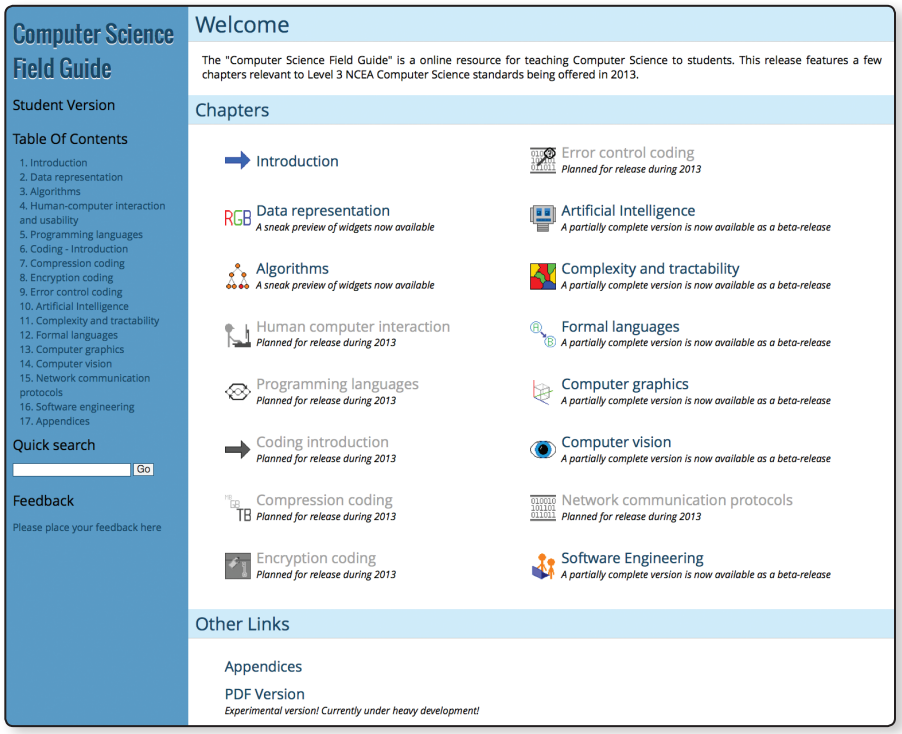
Establishing a Nationwide CS Curriculum in New Zealand High Schools

Providing students, teachers, and parents with a better understanding of computer science and programming.

In 2011, computer science was introduced as a subject in New Zealand high schools with a similar standing to subjects like physics, as part of a new set of education standards under the umbrella term "digital technologies."[2] Since then, at least 150 teachers have increased their skill sets in order to teach these unfamiliar topics, and thousands of students have passed courses in programming and computer science topics.

This rapid introduction of radically new material has not been easy; as well as having to train teachers and develop new teaching material, during the transition students had to prepare for assessments that no other students had done before, and teachers who embraced the changes often had to work with school leaders who had little understanding of what computer science involves.

The details of the implementation are available elsewhere,[1] but the key points are that "Programming and Computer Science" was introduced as an assessable subject for the final three years of high school, and the material was phased in from 2011 to 2013 respectively, so the first cohort of students to have taken the new topics left school in December 2013. With the new content, a course on computer science can include topics like algorithms, HCI, formal languages, complexity and tractability, intelligent systems, software engineering, and graphics and visual computing, in addition to programming. Problem solving and creativity already permeate the New Zealand curriculum, and the new content gives students the opportunity and tools to be creative in new ways.



***Computer Science Field Guide* table of contents (see http://csfieldguide.org.nz).**

When outsiders see topics such as "formal languages" as a school topic, they may wonder if things have been taken too far, and when teachers do an online search for that phrase they are likely to encounter an overwhelming array of teaching material. However, the purpose of the new curriculum is to give students a taste of the field of computer science, not to teach it in great detail. For example, formal

languages can be introduced by working with some simple Finite State Automata and experimenting with some regular expressions, concepts that can be introduced in a few hours of class time. From this students can appreciate the role of formal languages (for example, to find identifiers in a program) without having to grapple with details. To support teachers, a number of resources have been collected on the national teachers' association website (see http://nzacditt.org.nz), and are now being worked into a free interactive online textbook called the *Computer Science Field Guide* (see http://csfieldguide.org.nz).

The broad range of topics is important to help students see the breadth of options computer science offers, and also takes some focus away from merely programming by showing the range of knowledge needed to produce effective software, and the value that people with good human skills can bring to the discipline. For example, Human-Computer Interaction (HCI) is deliberately included early. The way it is presented is focused on evaluating existing interfaces to find design errors, taking into account usability principles and basic psychology relating to interaction. This can level the playing field—students who have been frustrated with computers find a way to articulate what is wrong with them, whereas sometimes students who themselves can develop programs with elaborate but confusing interfaces struggle to see the flaws because they assume the user understands the program as well as they do.

For teaching programming, the choice of language is not specified, although for the final year the language needs to support object-oriented programming and graphical user interface development. Many schools are using Scratch as an introductory language at lower levels, and then switching to a text-based language for the more advanced levels. Python is emerging as the most popular text-based language, but JavaScript, Java, and Visual Basic are also widely used.

Although the new content is targeted at the final three years of high school (when students have exams that become a permanent record of their school achievements), the material is filtering down to lower levels. This is proving to be valuable, as it is easier to come to grips with a subject if it is taught over a number of years rather than suddenly becomes available in the same year it is being assessed for a student's permanent record. Without the gentle introduction the risk is that students become focused on learning just enough to pass assessment with minimal effort, and they can miss out on enjoying and exploring the subject. The new curricula about to be introduced in the U.K. and Australia explicitly cover the full range of levels at school, whereas in New Zealand the curriculum allows it to appear at lower levels, but it is up to the enthusiasm and awareness of a particular teacher or their school leadership to make that happen.

The magnitude of the changes is easy to underestimate, and the teachers have borne the brunt of the transition.

Challenges with the Implementation

Before the 2011 changes occurred, computing in schools was focused on teaching students how to *use* computers, which led to courses that were not regarded as academically challenging, and thus were avoided by the students who were most likely to do well in a career in computer science. The changes have transformed it into a challenging subject, and this has created some problems in the transition as students, parents, and even student advisors are unable to understand the difference, resulting in the wrong students joining these courses (in New Zealand, students have a lot of choice over which courses they take in high school). In a 2012 survey, one year after the new courses started, 46% of teachers reported their courses attracted a significant number of students who lacked the academic ability to perform well, which reflects a difficulty communicating the nature of the new courses.[4] An important challenge is to help school management and student advisors to understand what computer science is, its value for students, and the kind of students who would do well to study it.

Marketing the subject is always a challenge. It is reasonable to let students know about the high demand for computer science graduates, the high salaries that are available, and the great work environments, but we also must ensure students do not overlook the fact that there is a lot of work to be done to get to the point of being hired, and that computer science can be a demanding subject. Learning to program is not trivial, and we need to straddle the line between not dressing it up as an easy option, but also encouraging students to try out the subject to find out if it is indeed a strength they did not realize they had. Also, money may not be a motivator for all students; for some students the best message may be that they can make a difference in the world, designing software that helps people in areas such as medicine, communication, and safety.

The Key Role of Teachers

The magnitude of the changes is easy to underestimate, and teachers have borne the brunt of the transition. In New Zealand, the new material was introduced with very little lead time, and for various reasons there was no opportunity to train a new generation of teachers, and very little time and resourcing to increase the skills of existing teachers. The majority of existing computing teachers are over 50 years old, yet most have embraced the change, primarily because of wanting to do the right thing for the students and the country, rather than due to any directives from management.[4]

The rapid transition has been somewhat unsettling for many teachers. From the New Zealand experience, we have found that teachers are willing to learn, but they need to be valued and supported. For those without a background in computer science (and this is the majority of teachers in New Zealand, as many had started as typing or commerce teachers), they could feel like an imposter, and it is important to value what they bring—

A key factor with the new material has been to avoid covering too much new content.

their teaching experience, wisdom, passion, and ability to relate to students—and provide them with learning opportunities in a form and at a time that they can access them. Some teachers will pick up things quickly if they have already had some experience in programming, while others will need some time.

The single most popular source of help has been a national mailing list run by the teachers themselves, where many thousands of email messages have been exchanged sharing ideas, teaching material, and peer support. This is a powerful tool because peers best understand the issues that each other face. In addition, several Google CS4HS events[3] have been run that provided intense training sessions focused entirely on preparing for the changes. More recent initiatives have been a formal postgraduate course teachers can take in their own time, and having university CS students work alongside teachers, helping them to understand the content while the teacher provides the classroom management skills.

Changes like this require a strong message to school management, who need to support it with resourcing, time for teachers to get up to speed, putting structures in place to enable students to make informed decisions about the new courses (such as having lower-level classes with some input from a computer science teacher), and providing the right advice to students and parents. One teacher commented that the management assumes that "all students know how to surf the Net and operate a smart device and therefore will find computer science easy." Changing this attitude will make the introduction of computer science considerably easier for teachers, and without it the changes can backfire, with the wrong students taking the subject, leading to a vicious cycle of low pass rates, students avoiding the subject, smaller classes in computing, and threats to computing teachers' jobs.

A key factor with the new material has been to avoid covering too much new content. Setting expectations too high could backfire by causing schools and students to avoid the subject; the changes will be a success if high school students simply find out what CS and programming are about. We do not need to teach them everything, and do not need to produce expert programmers, since there are plenty of opportunities to do that beyond high school. If students just find out they have a passion for the subject, the changes will have been a success. For some this may involve deciding that although they like computers, they do not like computer science, while others may find it is much more interesting than they imagined. The key is that they make informed choices about their career.

The overarching issue that continually arises is the lack of understanding of what computer science is about. If students, school administrators, teachers, and parents gain a good idea of what it really means, this will have the biggest impact on the skills pipeline. C

References

1. Bell, T., Andreae, P., and Lambert, L. Computer science in New Zealand high schools. In ACE '10: *Proceedings of the 12th Conference on Australasian Computing Education (Australian Computer Science Communications), volume 32*. T. Clear and J. Hamer, Eds. Australian Computer Society, Inc., Brisbane, Australia, 2012, 15–22.
2. Bell, T., Andreae, P., and Robins, A. Computer science in NZ high schools: The first year of the new standards. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (Raleigh, N.C., 2012) L.A. Smith King, D.R. Musicant, T. Camp, and P. Tymann, Eds. ACM, New York, 343–348.
3. Blum, L. and Cortina, T.J. CS4HS: An outreach program for high school CS teachers. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2007* (Covington, KY, 2007) I. Russell, S.M Haller, J.D. Dougherty, and S.H. Rodger, Eds. ACM, 2007, 19–23; http://dblp.uni-trier.de/db/conf/sigcse/sigcse2007.html\#BlumC07.
4. Thompson, D., Bell, T., Andreae, P., and Robins, A. The role of teachers in implementing curriculum changes. In *Proceedings of IGCSE 2013* (Denver, CO, 2013), 245–250.

Tim Bell (tim.bell@canterbury.ac.nz) is a professor in the Department of Computer Science and Software Engineering at the University of Canterbury in Christchurch, New Zealand.