Teacher Learning and Professional Development
Vol. 1, No. 2, December 2016, pp. 111 – 116

21st century bunkum: What do we value about kids learning to code, and why?

D. Kevin O'Neill *
Simon Fraser University, Canada

One of today's least questionable educational dogma is the importance of teaching "21st century skills." (Christou, 2016) According to one group of business leaders who catalogued them in the late 2000s, these are the skills that are supposed to make today's students adaptable to the rapidly-changing nature of work in a technological world. The skills as conceived by the Partnership for 21st Century Skills (Partnership for 21st Century Skills, 2009) include perennial favorites like critical thinking, but also the abilities to "adapt to varied roles, jobs, responsibilities, schedules and contexts," and to "develop, implement and communicate new ideas to others effectively." Computer programming (a.k.a. coding) is increasingly referred to as a 21st century skill which every child should be taught (Silcoff, 2016). Recently, Harel (2016) claimed:

> Our country must make a commitment to teaching every child computer science. That doesn't mean teaching watered down content and using simple coding apps but a strong curriculum that leads students to achieve real deep and broad mastery of computer science.

I should elaborate on the term "coding" because what is meant by it is often not clear. Harel's words above are useful because they underscore the important distinction between "coding" and computer science. The former is often used as a synonym for computer programming, while the latter (computer science) is a discipline involving a great deal more than what people generally think of as programming. At its most basic level, coding involves using a computer-interpretable language to describe a procedure that can run successfully and generate meaningful output. However, producing any but the simplest computer programs (and certainly getting paid to do so) involves quite a bit more. The current standards document of the Computer Science Teachers Association (Computer Science Teachers Association, 2016) includes, among many other things, a broad understanding of different data structures and data transfer protocols, analysis of the relative efficiency of programs, the ability to add features to existing programs without breaking them, and the ability to understand the ethical problems introduced by computing technology. None of this is implied in "coding."

Below I want to examine the sense and utility of teaching 21st century skills (including coding). Before I do so however, it is important to point out that in every era there are influential educational notions that are unproven, but go largely unquestioned. In the early 1910s when the sturdy and affordable Ford Model T and the many highly-paid factory workers who produced it were viewed as pinnacles of human achievement (Grandin, 2009), it was a virtually unquestioned truth that schools should be run as much like factories as possible. Today it is a terrible insult to compare any school to a factory; but in 1916 a popular education textbook written by the dean of

* Email: koneill@sfu.ca

education at Stanford University, Ellwood P. Cubberley, explicitly compared a well-run school to a factory without a hint of irony (Callahan, 1962).

Today, 21st century skills have a similar public-relations advantage to the factory of 1910: it is difficult for most people to find fault with such a glass-three-quarters-full view of the future. However, when considering ambitious educational goals like 21st century skills, it is worth keeping in mind what educational research has taught us about what skills are, and how they are developed. Before modern psychology came into its own, Latin and chess were often taught because it was believed that they developed powerful general abilities to think precisely and strategically. Few people doubted these assumptions; but in the 1970s a new body of psychological research began to show that they simply were not true – skills in chess and Latin, however worthwhile on their own, did not automatically generalize to other contexts and tasks (Bransford, Brown, & Cocking, 2000; Mitterer & Rose-Krasnor, 1986).

Interestingly, this evidence emerged just in time for computer programming to make its first appearance on the educational scene as a way to teach general thinking skills (Agalianos, Whitty, & Noss, 2006). Researchers did not have any more luck proving the worth of computer programming as a general approach to learning problem solving than they had had with Latin and chess (Mitterer & Rose-Krasnor, 1986; Pea, Kurland, & Hawkins, 1985), and this is useful to bear in mind now that computer programming is being promoted to the level of a core subject in many Canadian and American elementary and secondary schools, as well as elsewhere in the world.

Given the repeated pattern of soaring enthusiasm and crashing disappointment with regard to the teaching of highly general skills, it makes sense today for us to ask why so many politicians, parents, educators, and even researchers believe in "21st century skills" as teachable. A search of the U.S. Department of Education's database of peer-reviewed journal articles in education for items that refer to "21st century skills" results in over 80 hits, and most of the articles address questions of how to teach these skills, or in what areas of the curriculum they can be taught ("21st century skills in the English classroom!"), rather than inquiring into the more basic questions of whether they are worth teaching, or indeed can be taught. The latter are the more important questions.

The educational philosopher Robin Barrow (1999) suggested that believing in our ability to teach children to think critically about any old thing that comes along is like believing that we can teach them to dribble any old thing – a ball, a chair, or a rabbit – equally well. On the face of them, many of the 21st century skills are equally absurd. Where is there evidence that children can be taught, in a school, to "adapt to varied roles, jobs, responsibilities, schedules and contexts?" Clearly some people have proven themselves more adaptive than others over a lifetime of economic and social change; but is there good reason to believe that this was a function of curriculum or teaching? I have come around to the view that the reason why so many people want to believe in the merits of teaching 21st century skills is that they have become trapped in a situation where they feel little choice but to do otherwise.

Today there seems to be tremendous pressure on students to plan their educational steps, so they do not squander precious time or resources. With postsecondary tuitions continuing to rise and employers increasingly insisting on bachelor's degrees, students as early as the 10th grade are now expected to develop career plans. I have seen the effects of this planning mania on my own students: it tends to make them risk-averse, and squelches exploratory thinking. The current mania for planning is difficult to reconcile with the belief popular among business leaders, politicians and educators that the 21st century economy is a veritable runaway train of innovation that students, teachers and professors should be scrambling full tilt to catch up with (Christou, 2016). Consider for a moment: If the economy really were such a runaway train, would there not be less point than ever in planning? Career plans laid by students today should, in principle, become stale and dated much faster than they did when I was a child.

As I see them, 21st century skills are a hopeful but baseless conceptual invention that is used to reconcile the conflicting beliefs that a) we live in a time of faster-than-ever innovation, and b) today's students should be planning their steps and conserving their resources more carefully than those of previous generations. As the implicit argument runs, our kids would be fools to develop career plans with the expectation that they will find secure, long-term and dignified jobs with decent benefits; but they should plan to go just far enough in the right kind of school to develop a range of general skills and attitudes that will enable them to surf (or even better, create and profit from) the turbulent waves of innovation that are coming.

If, as I have suggested, the very idea of 21st century skills is wishful thinking unsupported by evidence, then our children will have tremendous difficulty coping with the pace of technological and economic change that many expect. I am unconcerned about my own children in this respect though, because there is good evidence that the economy and culture of our time are not actually more dynamic than those of recent history. It is certainly popular to say that we live in a time of unprecedented technological and social change, and some recent social changes (such as marriage equality) have indeed been epochal; but are they really greater than the changes people in our society experienced between, say, 1900 and 1950? That period saw the creation of the first child labor laws in North America, women's suffrage, the rise of powered flight and the automobile, the popularization of radio and television, a worldwide economic depression, two world wars, and huge advances in medicine, including the discovery of penicillin.

Giving credit where credit is due, the creations of Amazon, Apple, Facebook, Google, Netflix, Twitter and the rest of the .com potpourri simply do not compare to the innovations of 1900-1950 in terms of cultural impact. On the basis of jobs created they do not even hold a candle – most of these companies employ remarkably few people relative to their market capitalization and the profits that they generate (Rosoff, 2016). So, while it might be great for one of your students to work at one of these companies, most of them will not get that chance. Altogether, the Cambridge University economist Ha-Joon Chang (2012) argues that that so far, automatic washers and dryers have been a bigger deal in terms of economic and cultural impact than the Internet is, because they have enabled many more two-income families to exist, and have thus done more to change how family life is experienced.

Let us consider in more detail the places where our 21st-century-skilled students will presumably work. It has often been asserted by corporate leaders in high tech that there are not enough graduates in the right fields to fill job vacancies in science and technology. This is demonstrably false (Klein, 2014). If it were true, then by the law of supply and demand, the salaries of the qualified people would be bidded up. On the whole, this effect has not been observed, and this is not only because Adobe, Apple, Google, and Intel have illegally conspired to suppress wages (Isidore, 2015). Rather, it is at least partly because additional competition has been continuously brought into the labor market, either by hiring temporary foreign workers with no negotiating power (Thibodeau, 2014, 2016), or through encouraging more students to earn similar credentials, starting with events like the "Hour of Code" organized through the industry-sponsored Code.org, and aimed at hopeful students, parents and educators. No observant person can escape the asymmetry that students, educators and parents are now expected to be more loyal to the 21st century skills agenda than high-tech companies are to their workers.

As I mentioned at the opening, in any age there is a lot about how education is practiced that is not evidence-based. This is inevitable, because you cannot build a whole complex system of education on the basis of the little bit we know that is convincingly proven. Some decisions about curriculum are purely values-based, and we are forced to make some assumptions where evidence for decision-making is either impossible or too expensive to provide. However, at the very least we should not be betting our students' futures on ideas that actually run counter to what has been convincingly proven. In light of decades of research on skills transfer, and clear evidence of what is going on in the tech jobs market, bets on efforts to teach highly general 21st century skills to all students, including coding, seem very unlikely to pay off. On the contrary, the kid coding agenda seems likely to make our children unwitting pawns in a chess game played by high-tech entrepreneurs, who appear far less inclined to share the wealth than Henry Ford once was.

As a teacher educator who majored in computer science and has spent years developing software for use by teachers, I would not argue that coding should not be taught in at least some schools, at some grades. I would not even argue against teaching coding at an early age, starting with Scratch and other tools like the ones Harel has disparaged. However, insisting that every child must master computer science may be overreaching quite a lot given current trends in the labor market.

We have seen panicky overreach in curriculum building before. Following the launch of Sputnik by the USSR in 1957, a mania for improved science education developed in the western world, whose unfortunate legacy remains with us today. Commenting on the perceived crisis of national security brought about by the apparent superiority of soviet science, the U.S. nuclear scientist Dr. Edward Teller (a key contributor to the development of the first atomic bombs), argued "we have to strike at the root of the difficulty…by educating the whole population of 170 million people [in science]" (Hoffman, 2012). At this time the National Science Foundation became the major funding body for educational research in the world, and it became relatively difficult to sustain research on teaching and learning in the arts, social sciences and humanities.

In historical retrospect, it is stunning that so few Americans questioned Teller's leap in logic about the necessity of the whole population being better educated about science. Presumably, they did not question him because they respected his achievements and were terrified of the Soviets dominating the world – though of course this did not happen. When making decisions about what should be mandatory for all children to learn, it is essential to remember that nobody really knows what the future will be like, despite their fervent claims to know, and despite our own desire for a safe and predictable world (Gardner, 2010; Samuel, 2009). As a culture, we seem to be forgetting this lesson yet again – with the strong encouragement of industry groups whose potential profits rest on their predictions being believed.

I hope that the foregoing makes clear that the choice of whether every child should learn to code is one that cannot, in principle, be supported by rational economic planning or educational research alone. Knowledge of coding will not necessarily translate into any kind of well-paid employment for the majority of today's students, nor will it easily translate into improved problem solving in other areas of life. Despite this, we could still justify teaching it – but it is not clear whether the majority of educators, parents or students would be persuaded by this justification.

When we teach woodworking in schools, it is not with the assumption that every child will become a cabinetmaker, or will furnish their own home. When we teach about electricity, it is not with the assumption that our students will one day be employed as electricians, wire their own home extensions, or repair their own appliances. In a parallel way, coding can be viewed as a kind of craftwork that underlies the workings of the world we live in to an ever-increasing degree – even if it does not employ an equally rapidly increasing workforce. One might argue that like woodworking and electrical circuits, coding is worth learning about because it enriches our appreciation of the built environment around us, and to some limited degree, our ability to operate thoughtfully within it. Notice however that this is not an argument based on economic utility or general skills, but one based on the fundamental value that appreciating is better than not appreciating, that knowledge is better than ignorance. It is quite unclear to me how broadly this value is shared today. On the whole, politicians seem to value understanding quite a lot less than they value employability. I am really not sure about parents and teachers.

In my view we are long overdue for an honest, informed, and broad-based public discussion about whether coding as a universal skill (much less computing science as a discipline) is something that a substantial proportion of parents, teachers and educational leaders really value – regardless of whether it makes students more employable or more generally skilled problem-solvers, which in most cases it won't. Even if we judge that coding is worth learning about as a kind of craftwork, it does not follow that every child should learn it, and that every elementary teacher should therefore learn to teach it. At the moment there is scarcely any broad-based public discussion on this matter, and politicians and school leaders appear to be taking their cues entirely from industry (Christou, 2016), under the mistaken assumption that the computing industry can continue growing forever.

If we do commit ourselves to teaching every student to code, we need to organize ourselves to do it well. We have already had the chance to see how teachers teach coding with poor preparation or none – during the first "kid coding" craze of the 1980s, when I was a youth. Some of this teaching was productive and inspirational; but quite a lot proved pointless and ineffective (Agalianos et al., 2006; Papert, 1993). Mandating the teaching of coding today will therefore demand quite a lot of high-quality (read: expensive) professional development for teachers in the field, as well as those coming through teacher certification programs. Thankfully, today we know quite a lot more about how to teach computer programming effectively than we did in the 1980s. The question we need to ask ourselves is, do we really want to?

References

Agalianos, A., Whitty, G., & Noss, R. (2006). The social shaping of LOGO. *Social Studies of Science, 36*(2), 241–267.

Barrow, R. (1999). The higher nonsense: Some persistent errors in educational thinking. *Journal of Curriculum Studies, 31*(2), 131-142.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How People Learn: Brain, Mind, Experience, and School* (expanded ed.). Washington, D.C.: National Academy Press.

Callahan, R. E. (1962). *Education and the cult of efficiency: A study of the social forces that have shaped the administration of the public schools.* Chicago: University of Chicago Press.

Chang, H.-J. (2012). *23 Things They Don't Tell You About Capitalism*. New York: Bloomsbury Press.

Christou, T. M. (2016). 21st-century learning, educational reform, and tradition: Conceptualizing professional development in a progressive age. *Teacher Learning and Professional Development, 1*(1), 61-72.

Computer Science Teachers Association Task Force (2016). *CSTA K-12 Computer Science Standards. New York: Association for Computing Machinery*. Retrieved from http://www.csteachers.org/?page=CSTA_Standards

Gardner, D. (2010). *Future babble: Why expert predictions fail - and why we believe them anyway*. Toronto: McClelland & Stewart.

Grandin, G. (2009*). Fordlandia: The rise and fall of Henry Ford's forgotten jungle city*. New York: Henry Holt and Company.

Harel, I. (2016). American schools are teaching our kids how to code all wrong. Retrieved from http://qz.com/691614/american-schools-are-teaching-our-kids-how-to-code-all-wrong/

Hoffman, D. (Writer). (2012). The Sputnik Moment: Varied Directions, Inc.

Isidore, C. (2015, March 4, 2015). Google, Apple and other tech giants settle antipoaching case for $415 million. CNN.com. Retrieved from http://money.cnn.com/2015/01/16/technology/google-apple-antipoaching-settlement/index.html

Klein, K. (2014). The truth about the great American science shortfall. Los Angeles Times. Retrieved from http://www.latimes.com/opinion/opinion-la/la-ol-stem-science-math-shortage-20140224-story.html

Mitterer, J., & Rose-Krasnor, L. (1986). LOGO and the transfer of problem solving: An empirical test. *Alberta Journal of Educational Research, 32*(3), 176-194.

Papert, S. (1993). *The Children's Machine: Rethinking school in the age of the computer*. New York: Basic Books.

Partnership for 21st Century Skills. (2009). P21 Framework Definitions. Retrieved from Washington, DC: http://www.p21.org/storage/documents/docs/P21_Framework_Definitions_New_Logo_2015.pdf

Pea, R. D., Kurland, D. M., & Hawkins, J. (1985). Logo programming and the development of thinking skills. In M. Chen & W. Paisley (Eds.), *Children and microcomputers: Formative studies* (pp. 193-212). Beverley Hills, CA: Sage.

Rosoff, M. (2016). Here's how much each employee at a big tech company like Apple or Facebook is worth. Business Insider. Retrieved from http://www.businessinsider.com/revenue-per-employee-at-apple-facebook-google-others-2016-2?op=1

Samuel, L. R. (2009). Future: A recent history. Austin: University of Texas Press.

Silcoff, S. (2016). B.C. to add computer coding to school curriculum. The Globe and Mail. Retrieved from http://www.theglobeandmail.com/technology/bc-government-adds-computer-coding-to-school-curriculum/article28234097/

Thibodeau, P. (2014, March 4, 2015). Displaced IT workers are being silenced. Computerworld. Retrieved from http://www.computerworld.com/article/2855642/displaced-it-workers-are-being-silenced.html

Thibodeau, P. (2016). IT layoffs at insurance firm are a 'never-ending funeral'. Computerworld. Retrieved from http://www.computerworld.com/article/3077302/it-careers/it-layoffs-at-insurance-firm-are-a-never-ending-funeral.html