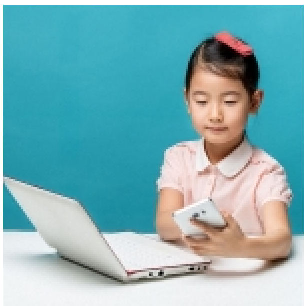

Dear parents, are our children being taught computer coding the wrong way?

Last Thursday, I registered my 11-year-old son Zachary for a free three-hour trial in computer coding. This trial class was held at KidoCode at Mont Kiara, Kuala Lumpur. This preview class was not just for the child but also for the parents, so that both parents and child would have a general idea of how coding was taught at KidoCode and about the center's general philosophy on learning, teaching, and social work.

KidoCode is a coding school where students learn on their own by watching proprietary online tutorials and completing certain number of tasks, which are then scored. Trainers are available should the children get stuck or want advice. (Photo from www.kidocode.com)

Alas, at the end of the three hours, I came away disappointed. What I experienced instead strengthened my suspicion and skepticism that our children are being taught the wrong way about coding.

The late Steve Jobs said over twenty years ago that “Everyone should learn how to program a computer, because it teaches you how to think.” Since then, many others have advocated that our kids must be taught coding, especially in today’s world of ubiquitous technology.

Increasingly more countries today are incorporating coding or computer science into their national school curriculum. Britain, for example, was the first G7 country to do such that for their 5 to 16 year old children. Even Malaysia is cognizant of this importance. As far back as 1988, the then Education Minister, Anwar Ibrahim, announced that coding was to be a part of the national curriculum by 1990. But Malaysia being Malaysia, flip-flop education policies are the norm: Anwar’s national coding policy never materialized. Since then, other similar national coding policies have been reintroduced, only to fail, wane or be quietly abandoned, such as the 2012 1BestariNet project, due to many reasons such as financial and logistics constraints.

Today our national ICT custodian is the Malaysian Digital Economy Corporation (MDEC), and at the risk of failing to learn from history, MDEC too announced in 2016 that it had plans to make coding part of the pedagogy of teaching at national schools, particular for science and math subjects by 2017. The year 2017 came and went, and it appeared that MDEC’s policy too have quietly been abandoned, now replaced by a less ambitious plan to create a Digital Innovators School instead by 2019.

So, yes, we are increasingly being told that our children, the younger the better, should learn coding. Coding schools and online self-learning portals (such as code.org and Codeacademy) are becoming increasingly popular with parents looking to enroll their kids for coding lessons. Even most schools today have at least some coding extracurricular activities to promote coding among students.

My problem is not whether children should learn coding; I firmly believed they should. But my problem is about the way our kids are being taught coding. Coding taught at online self-learning portals and coding schools are like ‘*pop computing*’ – a term coined by Dr. Idit Hazel, CEO of Globaloria, an organization for computer science education. Pop computing refers to a coding culture where coding is taught only to be quick, comfortable, and entertaining but suffers from being superficial, that kids do not have the necessary background or training to develop a deep and independent way of computational thinking.

A popular software to teach kids coding, for example, is Scratch, where kids learn programming by dragging-and-dropping blocks of code in a visual and colorful manner. Fun to use and even more fun to play the game kids develop, but it fails to deliver a multi-dimensional thinking required to apply computer science principles in other contexts.

Scratch is a visual programming platform where children learn programming by dragging and dropping blocks of code. Scratch is created by the Lifelong Kindergarten group at the MIT Media Lab (photo from freeCodeCamp).

My own son once took a Scratch class for seven months, and he immensely enjoyed his classes. He was willing to spend even up to six hours at his then coding school. But even up to today, he still fails to understand, let alone use, fundamental computer coding concepts like logical comparisons, conditional statements, and loops.

In other words, our kids are not taught the *fundamentals*. Coding learning software such as Scratch and Turtle Logo may make programming fun, but they teach software and computer science concepts in a very superficial manner. The fundamentals are made *implicit* rather than *explicit*. Without having a strong grasp of the fundamentals, I fear kids can only build apps they have been "guided" by their schools. MIT researchers Marvin Minsky and Alan Kay remarked that computer literacy is akin to music literacy. Musicians become proficient by listening, improvising and composing, not just playing (or duplicating) other people's composition. Pop computing risks our kids having a poor foundation in coding. Pop computing, I feel, is a more of a marketing rather than educational tool and more catered to hyperactive children or those with short attention spans to keep them focused long enough at the computer (or mobile) screens to learn … something, *anything*.

Some coding schools also pack in their curriculum, probably to make their curriculum appear comprehensive and holistic for parents. KidoCode, for instance, aim that their students learn very advanced computer science topics like object-oriented programming, software design patterns, and web programming (such as by using Flask). The school also attempts to teach four programming languages: Python, C++, JavaScript, and HTML. This is extraordinary considering that C++, in particular, is notorious even among experienced programmers for being a difficult language to learn. Also included in their curriculum is building electronic gadgets with Raspberry Pi and Arduino. All of these are to be completed in about 100 hours (which works out to three hours of class per day in a span of only a month).

Consequently, I fear coding schools teach is a case of "a little of this and that" but each topic in a very superficial, inadequate manner.

Furthermore, coding schools may boast of having several hundreds, even thousands, of enrolled students, but nothing is made known to the public about how many of their enrolled students actually follow through to the end of the course or about the quality of their "graduated" students. These schools are also neither accredited nor the quality of their teaching or lesson materials independently accessed.

My son Zachary never has a problem with his attention span, and he would actually want to understand what he is doing rather than just "duplicating" or making small adjustments to the provided coding guide or lesson. So, I see the usual, albeit more fun, route of pop computing is not for him. I have actually started to teach him coding, starting with Python programming.

For instance, for his first lesson, I taught my son the following classic example:

```
print('Hello, world!')
```

My son actually sees more meaning in this single line of code than his trial coding lesson at KidoCode such as follows:

```
tom = Turtle()
tom.shape('turtle')
tom.speed(100)
tom.color('green')
for c in [1,2,3,4]:
    tom.forward(100)
    tom.right(90)
```

where the concept of object-orientation is actually taught but in an implicit manner. Furthermore, would kids actually understand what `[1,2,3,4]` or even `Turtle()` mean? Sure, the above seven lines code help to make a cute animated turtle move and make a square on the screen — but I suspect most kids would secretly wonder about `[1,2,3,4]` — *what sorcery is this?*

Herein lies part of the problem: the lack of fundamentals being taught.

There is actually a difference between computer science and coding. Coding is really just our written instructions to the computer so that the computer does exactly what we want it to do. Computer science, on the other hand, is not just about coding but also a way of thinking that involves problem solving through a logical and methodological manner. Yevgeniy Brikman, who is a software engineer and writer on technology matters, went even as far to say that learning about coding or technology is less important than learning how to think. Sure, technology is ubiquitous, Brikman remarked, but that does not mean we must study about technology in schools.

"For example, we all fly in airplanes," Brikman further explained, "but getting a pilot license is not part of the K-12 [school] curriculum …. but the tools you need to understand how to think about flying should be part of the curriculum."

In other words, learning the fundamentals is crucial. Learning how to fly a plane should not be compulsory for everyone, but we all should learn the fundamentals related to flying. We should learn physics and math because they teach us about gravity, forces, pressure, velocity, friction, and lift. We should also learn biology because it teaches us about the effects of high altitudes on our human bodies, and we all should also learn history because it helps to explain about the invention of airplanes and their effect and role in societies.

The crux then for my son – and your children – is to understand the fundamentals, not just learn to duplicate or ‘modify’ guided code (or just build some apps because everyone seems to be doing it now). Once my son has nailed down his fundamentals, then I might consider sending him to coding schools to learn other topics and even build apps. Yes, the school’s lessons would probably be more fun and flashier than his dad’s, but at least my son would actually understand and appreciate what he is doing whilst having great fun.

Updated: 29 June 2018

References

- “American schools are teaching our kids how to code all wrong” by Idit Hazel, May 15, 2016, Quartz
- “Don’t Just Learn to Code, Learn How to Think Like a Computer Scientist” by Melanie Pinola, March 7, 2014, Lifehacker
- “Don’t learn to code. Learn to think” by Yevgeniy Brikman, May 19, 2014
- “Should kids learn to code?” by Gaby Hinsliff, The Guardian, Dec 3, 2015