Programming at West

Roger Avedon, January 2013
with much content provided by Don Geddis

Part of what made the Macintosh great was that the people working on it were musicians and poets and artists and zoologists and historians who also happened to be the best computer scientists in the world.

Steve Jobs

Almost all Nobel laureates in the sciences actively engage in arts as adults. They are twenty-five times as likely as the average scientist to sing, dance, or act; seventeen times as likely to be a visual artist; twelve times more likely to write poetry and literature; eight times more likely to do woodworking or some other craft; four times as likely to be a musician; and twice as likely to be a photographer.

Michele and Robert Root-Bernstein
A Missing Piece in the Economic Stimulus: Hobbling Arts Hobbles Innovation

How important is programming?

We can think of three broad levels of programming achievement a student might target:

1. *As a hobby*. In the same way that some children take up a musical instrument or a recreational sport, a child can pursue programming as a hobby for their own interest. Jack Dorsey, the founder of Twitter, took up programming as a hobby in junior high school, and wrote programs for taxi dispatching that are still in use.

2. *To achieve familiarity*. Students can be exposed to the basic concepts of programming. There are multiple gradations of familiarity as well. A lot of mathematics can be taught within the conceptual framework of algorithms and computation. The word *computer* itself originally referred to a person who did mathematical computations, and was popularized during World War 2 when women(!) were employed en masse to do the computations necessary to design aircraft wing shapes. Beyond mathematics, students can be exposed to the use of algorithms and programming to perform actions: to manipulate images, words, sounds, scientific data, etc.

3. *To achieve functional ability*. Students can be given training and practice sufficient to create working programs such as games or web applications.

A fourth level of achievement could also be considered, taking initial steps toward the vocation of professional programmer. Yet, in the same way that taking music lessons at early age is neither necessary nor sufficient for someone to become a professional musician, it is certainly not clear how such a vocational goal relates to, or is made more likely, by pursuing any of the three preceding goals, at least at the elementary school level. And if there are programmer-prodigies, who demonstrate profound programming aptitude before high school, they are not well known yet.

What is the appropriate age (range) to learn programming?

If programming is an important life skill in the modern world, does it matter whether the instruction starts in elementary school, as opposed to later in one's education? How young can children actually learn some programming?

Compare, for example, natural language acquisition, which is known to be much easier prior to ages 5 to 7. Is programming like natural language? Many in the field of programming would argue that the skill itself has more similarity with mathematical logic, for example, geometric proofs, which only the highest performing math students in the 8th grade study here in HCSD. Programming is as enrichment activity for mathematically advanced middle school students in some private schools around the Bay Area.

Programming clearly requires focus (on detail) and persistence, so young children, by virtue of their immaturity alone, may find it prohibitively frustrating.

That said, the Hour of Code (www.code.org) is intended to provide a meaningful introduction to programming for children as young as 6. The Robot Turtles board game claims to introduce some of the fundamentals of programming to children as young as 3.

We have little data on the development of profound programming talent in childhood. Individuals like Steve Jobs and Steve Wozniak (Apple), Bill Gates (Microsoft), John Carmack (game programmer), etc., invented the first generation of "personal computers" so they of course they did not have them as children. The next generation: Larry Page and Sergey Brin (Google), Mark Zuckerberg (Facebook), Elon Musk (Paypal, Tesla), etc., despite having access to computers as children, did not demonstrate profound talent until high school or later. One of the few early examples we do have is the aforementioned Jack Dorsey, who did produce commercially important software in middle school--taxi dispatching software that is still apparently in use a decade or more after he wrote it.

How is programming taught to children now? What's up-and-coming?

In the addition to the Hour of Code, there are many other online learning opportunities, some of which are pitched at a child's level.

The Scratch language from MIT is a well-known introductory programming language and environment.

Interest in tablet-based programming environments for education is very timely. Codea is an iPad application for programming based upon Lua, which, in turn, is billed as a powerful but simple programming language.

Khan Academy has an extensive online curriculum for computer programming.

Gamification is the deliberate use of games, usually computer-based, for non-game purposes, such as teaching. One well-reviewed attempt to teach algorithmic logic directly through a game is Cargo-Bot.

Lastly, our own Stephanie Tilenius is piloting a programming class at West through the Hillsborough Recreation department.