To Code or Not to Code, That Is the Question

Grady Booch

There Have been many periods in the unfolding of human history when our most learned people and our most treasured institutions have asserted—with unabashed hubris—that it was possible to catalog all that was known or that could be known.

That Was Then
Isadore of Seville was one such person. His magnum opus—Etymologiae—was published around 600 AD and served to codify all that learned men knew at the time (sadly, in his time, women were generally not considered worthy of such learning). The span of his work was breathtaking: across its 20 books, you could read about the saints, the sinners (although, from Isadore’s point of view, that was pretty much anyone who wasn’t a Catholic), the flora and fauna of the world, how to plant a tree, how to wage a war, and much more.

Isadore’s work turned out to be largely plagiarized from earlier sources, but nonetheless it served as a primary repository of knowledge throughout the period of the Middle Ages, losing its efficacy only when the Renaissance and then the Enlightenment brought about a fundamentally new way of thinking about the world. If you were a person of any means in Europe during the Middle Ages, you likely would have studied Etymologiae, or at least have known of its existence.

Today, we know of Isadore as Saint Isadore, canonized around 1600 and considered today by some as the patron saint of the Internet.

While Isadore focused on collecting knowledge from the secular domain, the same phenomenon has compelled the collection of knowledge in the religious domain. Between 180 and 220, the Mishna collected all oral tradition in the Jewish faith. This was followed around 500 by the Gemara, which served to collect all the commentary on the Mishna. Published in 1563, the Shulchan Aruch offered a comprehensive collection of Jewish law. In a manner of speaking, here we have not only the collection of core knowledge but also metaknowledge.

In the 1700s, the Western world saw renewed efforts to catalog all that was known, yielding works such as Denis Diderot’s Encyclopedie in France and the Encyclopedia Britannica in Scotland. The 1800s continued this movement to assemble all human knowledge into one place. Most notably, this was the time of the Oxford English Dictionary, which sought to codify every word in the English language from 1150 on. Initially, the project was expected to take 10 years; it took closer to 50. In any case, every enterprising, learned person in that era would have had a passing familiarity with these works, and certainly would have sought them out as sources of important curated information.

This Is Now
Jumping forward to our generation and shifting to the domain of computer science, we have Don Knuth’s Art of Computer Programming. In a manner of speaking, Knuth set out to codify the essence of computer science in one book. Well, maybe seven, with four of them published and others planned. Knowledge and understanding have this funny way of expanding.
Today, we have Google and Wikipedia. Google’s stated mission is “to organize the world’s information,” while the Wikimedia Foundation’s vision is “a world in which every single human being can freely share in the sum of all knowledge.”

The fundamental problem with any of these efforts, from Etymologiae to Wikipedia, is that they must cope with an ever-expanding corpus and hence can never catch up with the accelerating growth of information. In the case of Isadore’s work, the printed medium made his effort incredibly static; in the case of Google and Wikipedia, they’ve overcome any fully calcified state of knowledge by manifesting their respective repositories as living, breathing snapshots of the world’s information. Still, cultural and situational biases exist: what’s important to know at one place and time isn’t necessarily important in another. To be clear, I’m not saying that Google and Wikipedia are on a fool’s errand. Not at all! Rather, I’m saying that, while organizing all the knowledge of the world is a laudable goal, one must be humble and realistic about such a noble task.

Ours is an age not of horses and fields as it was in Isadore’s time, nor is it an age of ships and machines as it was in Dideroit’s. Back then, depending on who you were, you might need to know what side of a horse to get on, when to harvest your fields, or how to fix a rudder to get through the day. Today, ours is an age of information, and so we have very different life skills to master. To that end we therefore ask, what must a functioning member of our society know about computing?

**Two Epochs**
To answer that question, we return for a moment to the time of Charles Babbage.

To understand Babbage, you must also understand that he was a devoted Anglican. Indeed, early in his marriage, he sought to become a minister, but his requests were rebuffed (largely for political reasons). So, he turned to the world of mathematics. As a man of his times, he accepted the prevailing theology of the presence of a Divine Designer of the cosmos, a position he supported vigorously in writings such as the *Ninth Bridgewater Treatise*. At the same time, as a free-thinking scientist, he was also aware of the work of geologists such as William Smith, whose development of a large-scale geological map of England challenged the Bishop of Usher’s literal reading of the Bible, who had suggested an Earth that was only 6,000 years old.

Babbage was thus a man caught between two epochs of understanding. On one hand, he was very much a product of the church; on the other, he was the consummate scientist with a disciplined point of view that led him to the realization that information was something that could be mechanized. As the story is oft told, while once using an error-filled table of logarithms alongside his dear friend, the great astronomer William Herschel, he’s reported to have exclaimed, “I wish to God these calculations had been executed by steam.”
This was, for the time, a startling point of view. While the Industrial Revolution had brought about a way of thinking that asserted that human labor could be mechanized, Bab- bage made a parallel assertion, that the manipulation of more ethereal things—information—could equally be mechanized. It took a woman—Ada, Countess of Lovelace, and Bab- bage’s sometimes collaborator—to best express this new way of thinking: “We may say most aptly that the Analytic Engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves.”

Celebrating Coding

In our present time, there’s considerable interest in teaching coding as a basic skill. The nonprofit organization Code.org is perhaps the dominant voice in this space. Its vision “is that every student in every school should have the opportunity to learn computer programming. We believe computer science should be part of the core curriculum in education, alongside other science, technology, engineering, and mathematics (STEM) courses, such as biology, physics, chemistry, and algebra.”

Code.org isn’t alone in its efforts: Codeacademy, Girls Who Code, CoderDojo, and many others have added their voices to this worthy cause.

I celebrate each of these efforts; I think it’s a very good thing that everyone has a basic understanding of the technology behind the software-intensive systems with which we exist and are coevolving. I certainly accept my bias as a computer scientist, but I also know that, like the horse and the crops and the sails of earlier times, productive members of society must grasp the essence of the human-made cosmos around them.

That being said, I think we must be careful to not teach coding as just a vocational skill. Coding is important, don’t get me wrong: even Shakespeare had to learn how to spell and construct well-formed sentences before he could write King Lear. Rather, I think it’s essential that we fundamentally teach the notion of computational thinking, a point of view first proposed by Jeannette Wing. As she defines it, “Computational thinking involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science” (“Computational Thinking, Comm. ACM, vol. 49, no. 3, 2006, pp. 33–35”). Wing’s ideas parallel those of Carl Sagan, who once observed that “science is much more than a body of knowledge. It is a way of thinking” (“Why We Need to Understand Science,” Skeptical Inquirer, vol. 14, no. 3, 1990).

For the intellectually curious, of whom most will never program a computer in anger, learning to code is a gateway to understanding how to think computationally and therefore be more effective in an increasingly computed society. For the intellectually curious who may make computing a profession—or more likely, find themselves in a place where they must use computing as an essential tool to do their job—learning to code is just the first step in the journey to become a professional able to effectively collaborate with software-intensive systems.

We must be careful to not teach coding as just a vocational skill.

John Pierce managed the team of William Shockley, Walter Brattain, and John Bardeen who, at Bell Labs, invented the transistor; Pierce coined the term itself. As he so wisely observed, “After growing wildly for years, the field of computing appears to be reaching its infancy.” Indeed, we’re just at the beginning of an amazing journey, and therefore it’s good and reasonable that we remove the mystery of computing and teach the essential skills of computing to the generations that will come after us. ☮