Was computing dumbed down by the arrival of computer science in academia?

Alan Kay
Still trying to learn how to think better · Upvoted by Bill Salefski, Masters Computer Science & Electrial Engineering, University of Illinois at Urbana-Champaign and Kah Seng Tay, Master's and Bachelor's in Computer Science.Author has **687** answers and **8.2M** answer views · Updated 4y

This is an interesting question! My first reaction was “absolutely not, just the opposite” … but this is the old fogey from the 60s reaction. This is because — pound for pound — by far the best “real computer science” * in the 60s was being done in the major universities around the world (e.g. in the UK: Cambridge, Manchester, Imperial, Edinburgh, etc.; in the US: Penn, MIT, Princeton, CMU, Illinois, Stanford, Berkeley, UCLA, etc.; as well as major universities in Europe: ETH, Eindhoven, etc.).

I then started pondering what has happened since computing in academia “went wide” — especially in the US — starting around a little before 1980, moving from a few top places — MIT, CMU, Stanford, etc. — to pretty quickly becoming a subject matter in the ~4000 or so degree granting four year college/universities in the US.

My thought at the time was “How can they be doing this? There aren’t anywhere near 4000 top people in computing in the world — actually many many fewer. Where are they going to get professors of high enough caliber to offer a real college level curriculum in ‘real computer science’?”

Part of the impetus for this rush was another transition that had started in the 60s: the catalysis of the baby boomers and other factors starting to move universities to entities much more like businesses. A good book about this is “Imposters In The Temple”.

This was in full swing and was part of the boom in “CS” departments, regardless of quality.

Another large factor in “business orientation” was a stronger transition in universities toward vocational training rather than “in depth understanding of complex issues” that was the hallmark of education. And part of this had to do with the differential lopsided inflation in parts of human lives (e.g. especially real estate), that got more and more students to aim more for jobs than enlightenment.

I remember being shocked when Stanford CS — over the protests of many of its professors — decided (I think because of “encouragement” from the provost) to have Java be the initial programming language for incoming students. This is an enormous university with an enormous endowment, but now catering to perceived pulls from business rather than having its own special take on actual “sciences of computing” and actual “engineering disciplines for software”.

I don’t know enough about the whole sector of university computing in the US today to answer the question directly.

My own samplings over the years — via talks and visits to many university departments — have given me the prejudice that the students most definitely know almost nothing about the history of computing, and especially computer science.

Right away, this is a clue that “computing is not being taught as a real field” (for example, in Physics you not only learn F = ma but who came up with it).

In this limited anecdotal survey, I found no student or professor had taken the trouble to type “Engelbart” into Google to find out what Engelbart had really done, what he really thought, etc.

This indicated a real lack of both perspective and intellectual curiosity.

It also really pissed me off (a bunch of us had put quite a bit of effort in the 60s and 70s to invent personal computing and world-wide networks primarily to make it easier to become knowledgeable and then enlightened …)

I found no student who could define “computer science” as anything but something like engineering. The term “science” was kind of like the use in “Library Science” — there was no connection to what “Science” has meant since the 17th century.

This deadly embrace is found in NSF (I spent quite a few years on advisory boards there). It deeply and negatively affects high school curricula. And now has invaded the elementary school years with really grotesque and substandard ideas about computing.

My answer to the question: It is quite clear that “computer science” has been disastrously dumbed down. I think academia can be blamed for a lot of the problem (because part of its job is “keeping the flame” preserved against the general dumbing down out in the world). I think business can be blamed for “being dumb”: to the Dunning-Kruger level of not thinking they are, and then inflicting this on the general population in various ways, including schooling at all levels.

* Prompted by Will Razen's comment below, here is how we thought about "Computer Science" in the 60s, when the term was put forth as an aspiration and a question, rather than a done deal.

Science is the attempt to detect and gather phenomena to explain it via creating models (theories) of some kind that produce similar phenomena, and to do this in ways that try to get around what is weak about our senses and abilities to think.

So, if a bridge is built, it exudes phenomena, and can be studied, modeled and better understood. I.e. there can be a "science of bridges" (and of "structures generally"). In the large, this creates "Sciences of the Artificial" ** i.e. sciences that spring up around the artifacts that animals, mostly we, make.

A wonderful thing about "The science of bridges" is that an improved understanding and better models for "bridgeness" can be in turn used to design and make better bridges, which have properties of their own to be studied …

The "sciences of artifacts" are a most delicious art form and pursuit for those who love and are called to the upward lifting adventure of understanding leading to making leading to understanding leading to …

Most of the sciences — whether of nature or artifacts — will use mathematics of some kind — often newly invented — to help with the modeling process. As in Physics, this shouldn't be confused with the science part of things.

When Alan Perlis was asked what "A Science of Computing" could mean, he said it was "The science of processes; all processes". He could as well have said "The science of systems; all systems" (he meant this to be the same answer).

This recognized that algorithms, etc., are a tiny part of what computing is about. Computing is really about understanding, inventing, and building systems. As in many cases in the past in science, when existing math is not up to the task, new math has to be invented. In this case, one of the needs for new ways to think about what's going on stems from the degrees of freedom available and addition of the dimension of time.

The degrees of freedom and extent of dynamic relationships in desired artifacts, generally mean that they have to be debugged rather than proved. (And there are parts of math where proofs have some of the same quality — all proofs have to be debugged, some proofs actually require modeling them on a computer to debug them.)

Some of the earliest pioneers realized that the computer was "meta" in that it could be a great vehicle for modeling ideas about itself, so that much of the new math that was needed could be "extracted" from the "process space" itself. Many computer "theories" are models of processes written as running systems that can be debugged and explored.***

** see the book "Sciences of the Artificial" by Herb Simon (both a Turing Award and Nobel Prize winner, and one of the founders — with Turing awardee Alan Perlis of the CMU Computer Science Department).

*** We are sometimes asked how Xerox Parc could have been so inventive and productive in the 70s with just a few dozen computerists. One of the answers is lurking in the above. We thought in terms of systems of processes, made models of these, and ran these models on computer architectures we invented and constructed. I would call what we did the virtuous uplifting spiral of "Computer Science" understanding things to allow various "Computer Engineerings" to make new things, and etc.

47.1K views · View upvotes · View 19 shares · Answer requested by Austin Conlon 1 of 6 answers

Comments

Comments Recommended

Mark Miller · 4y
I watched a talk you gave at CMU from 2007, and I thought a dialogue you had with one of the students in the Q&A said a lot about the situation with CS in universities. You tried to get the student thinking in terms of "science," as scientists unde (more)

Carl Reynolds · 4y
I agree 100% with your answer. I have been feeling for a long time that CS was sliding in that direction. I remember when getting my Master's Degree having a conversation with another Master's candidate who told me of a knock-down (more)

Tim Elliot, Valuable Antique · 4y
"3 to 4 years of training in how to create HTML, CSS, and JavaScript so they can design web pages." That is today a recipe for a McJob and/or unemployment, because young people still believe that web development is paved with (more)