Stanford CS Commencement

Alex Aiken

6/17/18

I would like to welcome our graduates, families and guests, members of the faculty, and especially Jennifer Widom, a former chair of the Computer Science Department and the current Dean of the School of Engineering. My name is Alex Aiken; I am the current Computer Science Department chair.

Today we are conferring 315 Bachelor of Science degrees, 264 Masters of Science degrees, and 28 PhDs. The number of BS and MS graduates are both records for the department. We are all here today to celebrate your accomplishments. Congratulations to all of you for a job well done!

In the process of completing your degrees, all of you have navigated a sometimes confusing set of requirements. Now that you are about to graduate, I'd like to take a step back and ask: What was that all about? Did you ever wonder: Why do I have to take a class in THAT? Where did those requirements come from? You may recognize what I hold in my hand: A "program sheet" listing the requirements for a BS degree in Computer Science. Most of you have filled out at least one of these forms, so let's take a moment to talk about what it says.

The first observation is that nowhere on the CS program sheet does it mention that the University has its own requirements separate from the School of Engineering and CS Department requirements. Interestingly, CS program sheets do not even bother to mention the University at all, except for the two words "Stanford University" at the top of the first page. You are just supposed to know. So I have also brought with me a copy of the University's requirements for a Bachelors of Science.

Let's begin with the Writing and Rhetoric requirement. If I asked you how this requirement came to be, I imagine many of you might guess that it is the result of the English Department imposing its will on unsuspecting undergraduates. But you would be wrong. The truth is more interesting.

The first university is generally recognized to have been the University of Bologna, founded in the year 1088. It began as a law school, teaching, of all things, Roman Law. Now, you might wonder, given that the Roman Empire had by this point been defunct in Italy for about 500 years, why anyone in the Middle Ages was even interested in Roman Law, never mind establishing an entire university to teach it. But while the Roman government was long gone some of its institutions were still very much alive, and in particular Roman Law remained the legal framework of the many nations within the former empire and there was great demand throughout Europe for the necessary legal training. And what are the essential requirements of a lawyer? To be able to write and to speak, logically and clearly – Writing and Rhetoric. That's where the writing and rhetoric requirement originated in the university curriculum.

When the first universities were getting started, Constantinople, today's Istanbul, was the rich, sophisticated capital of the Byzantine Empire on the eastern edge of Europe. Scholars from the West began to read and translate really old texts that could still be found in the libraries of the East, and particularly interesting were a very small number of books by a man named Aristotle. The *Analytica Posteriora,* for example, explained propositional logic and why circular reasoning is a really bad idea. Soon Aristotle's works on logic, physics and ethics were incorporated into the standard university education along with writing, rhetoric, mathematics, and the arts in the form of music theory. You can easily identify the modern counterparts of these topics in Stanford's university requirements. The core content of a liberal education, together with the funny robes and hats we wear at graduation ceremonies, are the legacies of those first universities.

We now fast forward to the 19th century. The industrial revolution brought with it greatly increased demand for educated people who not only could think logically and express themselves clearly, but who could also make things. And so engineering began to be taught in universities. The traditional disciplines, validated by eight centuries of incumbency, generally viewed engineering as suspiciously close to vocational trades and at best a second class intellectual activity, a problem that lingers in some parts of academia to this day. In the United States, the great public land grant universities, with their mandate to serve the public interest, embraced engineering. So did a handful of private institutions: MIT, CalTech, Cornell and Stanford. Engineering was different from the traditional disciplines. Dealing with the physical limitations of actual machines and materials was messy and difficult and there was often a gap between what theory predicted and how implementations behaved. A distinctive engineering culture arose that focused on practicality, reliability and efficiency. As more was learned about the science underlying engineering disciplines, more advanced mathematical techniques were introduced and engineering education took on its distinctive combination of abstract theory and practical implementation. And that is where the engineering requirements on the program sheet come from, originally serving as the underpinnings of studies in mechanical and civil engineering, and later electrical engineering, and finally computer science.

And now we come to the most recent chapter, the last 50 years, and the rise of computer science. From the beginning of the first academic computer science departments in the 1960's people recognized that computer science, with its universal machines and highly malleable software, would eventually influence many if not all areas of human activity. Initially computer science had just a few subtopics - algorithms and the theory of computation, programming languages, numerical analysis, architecture and artificial intelligence – but the expansion into entirely new aspects of computing such as operating systems, databases, graphics, security, machine learning, biocomputation, and human computer interaction came quickly. Just ten years ago the field had grown to the point that it was no longer possible to even pretend to teach all of it in a four year degree, and at Stanford we reorganized the computer science major around tracks where students could focus their studies on a subarea that most interested them.

And that brings us to today. What are we to make of your Stanford education? Any ideas that survive in the curriculum for a thousand years probably have real value, and the ability to communicate well, think clearly, and reason ethically are skills you will need in everything you do. Your engineering skills will be important in any complex project, particularly the ability to construct models to gain insight while still recognizing that the models are abstractions that can't and shouldn't capture the full complexity of what is really going on. And finally the computer science skills, what will you do with those?

Part of the answer to that question is easy. There is a huge demand for computer science talent, and I am confident that you will all find roles where your abilities as computer scientists will be well utilized. It is a fantastic time to be starting out as a computer scientist, and the continuing expansion of the field and the rate of new discoveries guarantee that it will be a great time to be a computer scientist for many years to come. But with the opportunities will come corresponding responsibilities, and looking to the future, the responsibilities of computer scientists to society will increase greatly.

We are in the midst of a decades long technological revolution driven by computing. The first 50 years were pretty exciting, but the next 50 are really going to get peoples' attention in ways both good and bad. The first web browser appeared in 1993, just 25 years ago and just a few years before the youngest of you were born. Only last year, though, in 2017, did we pass the point where 50% of the world's population is on the Internet. Put another way, almost half of the people in the world still don't even use computers, and of those who do most are recent adopters who know relatively little about the world they have entered.

And the on-line world we have built for them to inhabit has serious challenges. I'll mention just two. The first is cybersecurity. Part of the problem is technical, that we have failed to build systems that can provide basic security guarantees for our on-line lives. Fixing this situation will be a huge and expensive engineering challenge with no silver bullets, but before we can fix it we first have to agree on what a solution should do, and the situation is currently quite confused.

However, the bigger part of the challenge is not technical but social. Even if we had a secure infrastructure, whatever that really means, educating the world's population about how to use it properly is a non-trivial challenge that we have also so far failed to meet. There are lots of people in the world today who are careful to lock the door to their house when they go out in the morning but are also happy to share their bank account with strangers they meet on-line simply because the stranger asks them to. And sophisticated versions of such attacks still routinely ensnare prominent people in business and government who we might assume would know better. More generally, at a time when cyberattacks are rapidly increasing in sophistication, automation and potential harm, most users are unsophisticated and don't understand the threats they face, let alone how to protect themselves. It is not a good situation and unfortunately it is likely to get worse before it gets better.

The second example I'd like to mention is the recent rapid advances in artificial intelligence. I should disclose that I am not at all an expert in AI, but complete lack of qualifications has not been a barrier to anyone else expressing opinions on the subject, and so I'd like to have my turn as well. There is currently a great deal of attention in the popular press on the potential downsides of machines with true human level intelligence, which to me still seems a rather long way off, and much less attention on the more immediate issues around the safety, fairness and interpretability of applications of the machine learning systems we can build right now. The issues are at least recognized and we are working on them, but deployment of machine learning across all aspects of society is proceeding rapidly without those solutions yet in place.

Stepping back, these are just two of many possible examples I could cite where very large numbers of people are experiencing rapid technological change driven by advances in computing. And the problem is that the rate of technological change can be more rapid than the rate at which all of those people and their institutions can adapt. As organizations or even whole societies reach the limit of how quickly they can change, serious stresses become apparent.

And who will they turn to to help them with those problems? This will be your responsibility, and because computer science expertise is scarce and will remain scarce, there will be nowhere to hide. Many of you will find yourselves in situations where you are called on to make decisions that will affect many people, perhaps millions or billions. Use your analytical skills and your ethical judgment to steer those decisions toward good outcomes not just for you and your organization, but for all of society. Many of you will also find yourselves asked for advice from people who know much less than you about the science and engineering, and your ability to communicate clearly and still accurately in those situations will be crucial. Fortunately, you leave here today equipped with the best of all computer science educations, or at least the best that we've been able to come up with in the last one thousand years. The opportunities and the responsibilities that await you are substantial, and know that we both wish you the very best of luck and that we are also counting on you.

Congratulations to the graduates and come back to visit us any time. Thank you for your attention.