Computer Science, or Computing Science as the late Edsger Dijkstra called it, is unique among the sciences. It has a complex nature, being both a scientific and an engineering discipline. It contains within it the practice of programming, which in addition to being a rigorous mathematical and engineering endeavor, is also an art akin to poetry or literature. In my teaching I strive to do justice to each of these aspects.

Science can be understood as "the effort to reveal the kinds and laws of nature to the faculties of independent reason." The notion of independent reason is beautifully captured by the motto of the Royal Society as "nullius in verba", i.e. “nothing on the word”. I strive to make this a central tenet of my teaching. To instill this as a value in my students, I work to teach them that it is insufficient to simply assert the correctness of an algorithm or its realization as a concrete program. Neither is our intuition sufficient, nor is it enough to demonstrate correct execution on a handful of test cases. Rather, we must proceed from a careful and deep understanding of the specification, and through stepwise refinement, disciplined reasoning, and an understanding of the principles involved, develop our algorithms in conjunction with proofs of their correctness.

Students need to know and become a part of the broader Computing Science tradition by understanding its history, and finding heroes within it. Our field has roots that go as deep as Pythagoras, and students need to have an appreciation for the intellectual tradition that they are inheriting. Towards this end I pepper my lectures with illustrative stories, and teach them about great computer scientists such C.A.R. Hoare, Niklaus Wirth, Edsger Dijkstra, John McCarthy, Alan Turing, and others.

Engineering can be understood as "the application of scientific knowledge to the design of technology." I frequently develop programs in class as a part of my lectures. In doing so, I emphasize the careful design of a program from its specification. In addition, I tie in to each program's development a discussion of the underlying principles involved: What kind of problem are we trying to solve? What is its nature? How is our choice of data structure guided by our answers to the previous questions? How does that impact our algorithm and vice versa? How do our choices affect the efficiency of the resulting algorithm and our ability to prove its correctness? Etc. Students therefore can see that our choices have consequences, and we can begin to develop a methodology for comparing programs, and a feel for the art and aesthetics of programming. These lessons are further illustrated through anecdotes from my tenure at eBay by showing how various decisions had far reaching and practical effects in an industrial setting.

One definition of art is "something that is created with imagination and skill and that is beautiful or that expresses important ideas or feelings". One means by which I have my students develop the art of programming is to teach them to read different programs which implement the same specification. This practice was developed as a consequence of my tenure at eBay, where I often had to read already developed programs in order to extend or correct them. By learning how to read a program, and by reading different implementations of the same specification, students can learn about and develop their own programming styles. By providing them with examples of well written programs to study, students can learn how to express themselves in their programs with both clarity and elegance.

In addition, I try to instill a sense into my students that they have an obligation, both professional and moral, to strive for a high standard. It has been my experience that through a disciplined and concerted effort, most students can achieve such a high standard, and that even those who fail are well served by the experience. I spend a great deal of time working individually with students outside of the classroom as well in furtherance of this goal, and to serve as an example. One of my great joys while teaching is to see a student who after a great deal of effort finally achieves mastery of a concept. I have had several students who after failing their first semester with me have retaken the course and were at the top of their class due to their efforts.