DONALD E. KNUTH

Scholar with a Passion for the Particular

by Karen A. Frenkel

- Fletcher Jones Professor of Computer Science, Stanford University, Stanford, California.
- B.A. in mathematics, 1960, Case Institute of Technology, now Case Western Reserve University, Cleveland, Ohio.
- Ph.D. in mathematics, 1963, California Institute of Technology, Pasadena, California, thesis on finite projective geometry. "One of the two branches of mathematics that I've so far never applied to computer science," says Knuth. "I've seen some use for almost every other branch."
- Received the first Grace Murray Hopper Award in 1971 for volumes 1 and 2 of *The Art of Computer Programming*, Addison-Wesley, Reading, Massachusetts, 1968 and 1969. To date, volume 3 has also been written, second editions of volumes 1 and 2 are complete, and four more volumes are planned.
- Other ACM awards: 1974 A.M. Turing Award, Turing Lecture: "Computer Programming as an Art"; 1986 Computer Science Education Award; and 1986 Software Systems Award.

"Age 30 is kind of appropriate because I got the first copy of volume 1 from the publisher nine days after my 30th birthday. So, a large part of the work had been done when I was 30 years old. They already were working on typesetting the second volume."

Commenting on his books' influence, Knuth says, "It's been phenomenal from my point of view. In 1976 a study was done of how many people writing papers on computer science made a reference to my book somewhere in their articles, and it was found that about 30 percent of the papers in *Communications, Journal of the ACM*, and *SIAM Journal on Computing* cited the book. So it has an impact in that way." What about sales? Knuth notes that publishers may joke about professors whose books never sell, but they don't apply here. "I know that people buy the book. I don't know how many read it. But the sales have been incredible. I think something between 1000 and 2000 copies [have been sold] per month for 20 years."

In 1977, upon reading galley proofs of the second edition of *The Art of Computer Programming*, Donald E. Knuth reacted boldly. "When I realized that you could make beautiful books just by patterns of zeros and ones, I couldn't help but try to set up zeros and ones the way I wanted. I wanted my books to look good, or else I didn't want to write them." The technology had changed—the new computer-based systems were producing pages with bad spacing and limited typefaces, which were worse than those produced by old manual techniques—and the resulting galleys were "just awful," he says. So keen and so offended was his aesthetic sense that Knuth resolved to improve the typographical state of the art by creating his own system. The new printers with tiny dot matrices had to be instructed as to whether ink was or was not wanted on each part of the page. That is the same as putting zeros and ones into computer memory, where the ones correspond to ink and the zeros correspond to blank space. A 10-year hiatus in writing the remaining volumes of *The Art of Computer Programming* resulted while Knuth engaged in a flood of prolific programming to create TEX and METAFONT—two typographical systems that place letters and symbols on the page and create letter and symbol designs, respectively.

Developing these systems was not only a matter of taste, but also a computer science problem whose solution has influenced Knuth's programming practices. It was a wonderful surprise, says Knuth, that research on "an application way out there" somehow "came right back into the center of computer science." Having a good tool for typography, Knuth was able to write better programs because he could document the programs better than before. Knuth says, "I could now completely change my way of writing computer programs to one that would make the process very much like writing literature—like writing a book."

MINIMIRACLES AND DAUNTING DETAIL

Uncovering such a link, and the unexpected application to his own work, was a "small miracle," says

Knuth, one of many he has experienced in writing his books. Though small in scope, these miracles have been and can be enormously important, says Knuth. "Usually the real changes in computer science seem to grow out of a lot of small things rather than one big thing." Reflecting on the three finished volumes, and the huge amount of material he had to summarize, Knuth says each part had its own attraction because each developed differently, unexpectedly, and even mysteriously: "I would start out to write a section, and there would be seemingly much, much more to be said than could possibly fit in 70 pages. I would start out and think, 'Oh, my goodness, I'll never be able to do this.' And then, as I started working on the section in an intensive mode, all of a sudden, like magic, some unifying principle that wasn't originally obvious would emerge that would make it possible to condense a lot of the other literature. . . . I would boil it down to something like 50 pages and feel like I had been reasonably complete."

Carolyn Caddes, from *Portraits of Success*

DONALD E. KNUTH

Initially Knuth's tomes, regarded by many as classics, were to have been a single book on how to write compilers. The assignment came from an Addison-Wesley editor while Knuth was still a second-year graduate student at Cal Tech in January 1962. Explaining the ambitious outcome, Knuth says, "I figured, as long as I'm going to do a book on compilers, I should include a few other chapters on basic techniques that people would use before they got all the way to compilers. So I threw in a chapter on everything I was interested in." A few days later he went to contract.

To prepare, Knuth used lecture notes for a course he was teaching that would culminate with writing compilers. The following summer, after graduating, Knuth realized that he would need to expand the topic further. He wrote to his publishers and said, "Do you mind if I make this book a little bit longer, because I think there's a need for explaining these things in somewhat more detail." And they said, "Oh no, go right ahead. Make it as long as you feel necessary." He finished the first draft, which was handwritten in pencil, in 1967. "It was 3000 pages," says Knuth, "and my handwriting's very small." They didn't realize how badly he had estimated the relationship between his handwritten notes and the printed page, Knuth recalls. The original editor, who had since been promoted three times, explained that Knuth had taken his instructions very literally and that, if the first chapter was any indication, the whole book would be impossible to package. Negotiations as to how to cut it down proceeded, and for a few months, there were plans to produce only three volumes. But readers looking over the material liked seven of the twelve chapters, says Knuth, and the publisher "probably figured that they wouldn't be able to talk me out of the other five, so they decided to package it in seven volumes with one 'good' chapter in each volume."

Does this mean that you cannot write about compilers without including almost all the basics of programming? Yes, says Knuth, although a few chapters could have been left out then but would have been grave omissions now. In 1962, chapter 7, "Combinatorial Searching," for example, was not necessary. Says Knuth, "I put it in because it was the most fun for me. I enjoyed writing those programs more than any others, and there were hardly any combinatorial algorithms known in 1962." Now that he has resumed writing just that chapter, the field has "mushroomed to the point where that chapter's going to be more than one physical volume in size."

A FAITHFUL SPOKESMAN

Keeping up-to-date is a kind of scholarship for Knuth. "What I'm trying to do in these books is be a spokesman for the entire computer science community and present everyone's ideas in a consistent way as faithfully as I can," Knuth says. "I do this [keep up-to-date] by what I call 'batch processing'; that is, I work on one area intensely at a time and put everything else out of my mind. I read 100 papers on that one subject, spending three months on it, and then write up that part of the book; and then another three months on another part." But presenting viewpoints requires originality, Knuth notes. "To be consistent," he says, "I have to take two authors' treatments of their work and answer for each one the questions that the other author has asked."

Although the process can be quite deliberate, it can, on the other hand, flow in a more natural and almost inevitable manner. Of chapter 4, volume 2, which was originally called "Miscellaneous Utility Routines," Knuth says, "I didn't have a good title. I needed to throw in a bunch of neat ideas, and I didn't see what else to call them. Four years later, after I had written