

### **Textbook history as I lived it\***

I started teaching the intro CS course in the late 1960s as a graduate student. The languages were Elastic, an assembly language, and Fortran. We used a manual for Elastic and a thick intro textbook for Fortran. The only author I remember is Elliot Organic (sp?). Daniel McCracken's Fortran book came out in the 70s, and then there was the almost universal switch to Pascal as the intro language. The switch was made before any intro books were written, so we used Grogono. (I'm sure you all remember Grogono—a good book, yes—but an intro text? I don't think so.) I believe Mike Clancy's *Oh Pascal* was the first true intro Pascal text, followed by Elliot Koffman's book in 1980, followed by Dale/Orshalick in 1982. I believe Walter Savitch's Pascal book came next.

From 1983 on, our revisions were not so much content driven, and competition driven. Elliot would come out with a new edition, and Walter and I would follow. Our publishers might have thought that the used-book market was a big influence, but at least to me it wasn't nearly as important as what Elliot and Walter were doing. Also, I think we all become more conscious of teaching problem solving. The Pascal language didn't change much, but our perceptions of what was important pedagogically did.

### **Textbook future as I view it**

Since the switch away from Pascal that began in the early to mid-1990s, the language changes have driven the need for new editions. Not long after the first series of C++ intro books came out, the new ANSII standard was adopted, requiring new editions. I don't think we can make "embracing rapid change" or "working on fundamentals" an either/or proposition. Certainly the changes in languages will require more frequent editions, but we should take the opportunity to always fine-tune the presentation of the fundamentals with each new edition.

How profound is the change to object-orientedness? If you had asked me 6 months ago, I would have said "not very much." But that was before being immersed in working on our new Java intro book. Our decision to make it truly object-oriented has required me to rethink the problem-solving process. And it is certainly more of a shift than just changing vocabulary such as "sending a message" for "invoking a function" or "application" for "program."

I look forward to hearing what my colleagues on the panel have to say about these issues.

Nell Dale

\*The dates are approximate, based on my memory and not on fact.