

# White Paper for ICER 2006

Stuart Reges, Computer Science & Engineering, University of Washington

I expect to be the curmudgeon of the group. I love the idea of finding new ways to teach computer science, but I don't see the compelling need for change and I'm skeptical about the likelihood of success. Despite my pessimism, I don't advocate doing nothing. I recognize that experiments are already underway and that others are sure to follow. I hope to be part of that process.

I think that the recent decline in cs enrollment will reverse itself once students realize that not all computing jobs are being outsourced to India. If, as most of us believe, there continues to be a strong job market for cs graduates, then the word will spread and eventually this situation will turn around.

The ICER proposal asks us to look forward 5 years, but I started by looking backwards. Ten years ago David Kay published a SIGCSE paper "Bandwagons Considered Harmful, or The Past As Prologue in Curriculum Change" in which he warned us about the dangers of experimenting too much with our introductory CS classes. He identified several different directions people were exploring at the time: (less programming, Scheme, OOP, C++). I myself was part of these experiments exploring the use of C++, OOP and Java.

Around three years ago I found myself thinking that after all that experimentation, I had ended up with a CS1/CS2 sequence that was inferior to the one I had been teaching in the late 1980's. My new course was taught in Java instead of Pascal and included some important concepts like encapsulation that were not supported in Pascal, but the courses themselves were not nearly as good as they had been before. Students were learning less and, more importantly, students were enjoying the classes less than they used to. In the last two years I have been exploring a "back to basics" approach where I am trying to find a reasonable middle ground. The early results are encouraging.

I also find myself skeptical because it seems that so many universities teach computer science badly. Assignments are often silly (who would want a program that does that?) or at the wrong level of difficulty (too easy or too hard). Instructors require students to buy expensive textbooks that they often don't follow. Faculty who have little commitment to introductory classes are forced to teach them anyway ("It's your turn"). Introductory classes often get the least motivated and least capable TAs ("We have to support him somehow"). And the language and other technologies like IDEs are changing so rapidly that students are stymied by technical glitches that have nothing to do with computer science ("What does NoClassDefFoundError mean?").

In terms of "dreaming big" and imagining what could have a significant impact, I'd ask for the following:

Make sure that cs1 and cs2 at every university is taught by excellent teachers who inspire young people by assigning them challenging homework that requires computer science reasoning to solve and by providing them with the resources they need to succeed.

Notice that I didn't say anything about whether the course is taught in Java or whether it is "objects early" or whether it is even a programming course. The aspects I have listed above (excellent teaching, inspiring young people, challenging homework, resources to succeed) are so much more important that these other issues become minor details. Unfortunately, I think that we fail miserably at achieving this goal even without curricular experimentation. With experimentation the situation goes from bad to awful.

Of course, diversity is an important issue to consider. I did a careful study of attitudes among the students taking the CS1 course at the University of Washington. I found that the women who performed best in the class were not seriously considering computer science as a major. There was nothing new in my results, but it allowed me to focus on individuals to get a sense of what was going on and what can be done. I was able to convince many of these talented women to reconsider computer science as an option. I think the strongest argument was that I could point out that students in the class were solving really difficult problems that these women were particularly good at solving (unlike many of the other students, they *enjoyed* solving these hard problems). That appealed to them and I think was more important than issues like whether or not the problem was a practical application. Of course, overall perception of computer science as a field was important and there it was important to be able to point out where a computer science degree can lead you.

I would say that my successes have come from teaching a high-quality course and giving special encouragement to nontraditional students who do well in the course. I have been helped by the fact that our CS1 class is required in many undergraduate majors. So I might get a woman who is thinking about Bioengineering or Mathematics who discovers that she has a strong aptitude for CS. I could be even more successful if I had more students in my CS1 class and if they took it early enough in their undergraduate career that they could switch to computer science as a major.

One of my primary goals, then, would be to get more students into the CS1 class and to get them to take it early in their undergraduate career. If we then teach an excellent class, we have a good chance of attracting talented people to the field.

My two pages are almost filled, so I'll add one final thought as long as I'm trying to dream big. Cultural phenomena are very powerful. Right now we are suffering from the misperception that all programming jobs are going to India. Crime shows like CSI have increased interest in forensic science and the book *Freakonomics* has increased interest in Economics. I don't foresee a TV show focusing on computer science (although you never know), so I'm investing my hope more in a book. I am considering attempting it myself by collecting together some of the ideas I have explored over the years with small groups of honors students (many of whom were convinced to major in Computer Science). My working title is *Virtual Things: Seeing the World Through Computer Science Glasses*.