Why Women Avoid Computer Science

The numbers prove women embrace the “precision” of mathematics. Could it be the ill-defined nature of computing is what drives them away?

Paul De Palma

Women find careers in computing unattractive. A report from the American Association of University Women says that women account for only 17% of the high school students who take advanced placement exams in computer science and earn only 28% of the undergraduate degrees.[1] This confirms an earlier report that noticed a sharp drop in CS degrees going to women between 1986 and 1994.[2]

Since no one really knows why women avoid computer science—or what to do about it—I feel justified in offering a guess of my own.

As it happens, the literature fairly bubbles over with speculation as to why there are so few young women in computer science courses. We hear about math anxiety, violent computer games, the scarcity of mentors, and a supposed female preference for “relational work”.[7] Since no one really knows why women avoid computer science—or what to do about it—I feel justified in offering a guess of my own.

Among the many reasons offered, math anxiety is the most obvious. It is also the least defensible. Commentators never seem to notice that women receive almost half of the undergraduate degrees in mathematics. In fact, they received nearly 40% of them in 1970, well before the women’s movement became a mass phenomenon.[5] Not only do young women not avoid mathematics, they embrace it. What if the precision of mathematics, that “most masculine of subjects” in the words of one study,[7] is exactly what has long invited women? The flip side is that the ill-defined nature of computing is what drives them away.

Young men drawn to computer science, engineering, and physics like to tinker. They enjoy taking things apart and putting them back together. They like kits, gadgets, and screwdrivers. They were the boys who set up the audio-visual equipment in high school 30 years ago, and who now man—the choice of gender is deliberate—the school’s computer network. They are fascinated with anything that moves, especially if it has wheels or wings, and, crucially, is not alive.[4] The men usually given credit for the microcomputer all started with screwdrivers and soldering irons. Bill Gates and Paul Allen built a Basic interpreter to run on their Altair 8800, a computer kit for hobbyists, in the mid-1970s. Steves Wozniak and Jobs, of Apple fame, built their first machine to dazzle pals in Silicon Valley’s Homebrew Computer Club around the same time.

In fact, I claim that microcomputers are responsible for the steep rise in the number of women entering computer science following its introduction, as well as for the steep drop a few years later.[6] In 1971, fewer than 2,400 students received degrees in computer science from a handful of academic departments. By 1986, that number had jumped to nearly 42,000, including almost 15,000 women. It is clear that the dramatic growth of computer science as an academic discipline is due to the microcomputer and, of course, to the extravagant promises that buzz around it. If the number of computer science degrees had continued to grow as it had from 1975 to 1985 (and if the population grew at its average annual rate over the same period), by next year everyone in the U.S. would be the proud holder of one. Lucky for us this didn’t happen. The number of recipients began to drop off sharply in 1987, stabilizing by the mid-1990s at about 24,000.

We know why both men and women entered the field through the academic portal in great numbers in the 1980s. The attention paid to the microcomputer led many to believe it was a talismanic object. Why did these numbers drop, and why more sharply for women than for men? For men, the explanation is obvious. Traditional paths to wealth like law, medicine, and business are more certain, and over the long run, far more remunerative, on average, than computing. Further, computing is not a true profession. One need not suffer through a computer science curriculum to enter the field. Finally, computer science is more difficult than many aspiring young millionaires expect. These reasons serve to drive women away as well. And the tinker factor combined to drive them away in greater relative numbers.

Computing has always had an indeterminate feel to it. With its unreadable (and, now nonexistent) manuals, its feature piled upon fabulous feature, our tools are always more complicated than what they're used for. The old programmer's dictum that we use 10% of the features 90% of the time was true long before the first PC. And the manuals from the glory days of the mainframe were, if anything, more opaque than today's commercially written documentation. This has always selected for success those young men—and they were almost always young men—willing to spend endless hours tinkering with software. The microcomputer only exacerbated a process that had long been in place. It simply added hardware tinkering to the software tinkering that had always defined the testosterone-infused conversations among programmers. Perhaps the day will come when young women find things that roll, fly, and plug into an outlet as fascinating as do young men. The nature/nurture debate is an old one, and, at least in the case of toy trucks and ponies with pastel hair, not likely to be settled soon. One day we may find girls playing with trucks and entering computer science in increasingly greater numbers. Until that happy day, however, I have several suggestions to make the field more hospitable. Let's look in a very unlikely place for what young women seem to prefer. Let's look to mathematics itself. Long before law and medicine opened their doors, a significant fraction of undergraduate degrees in mathematics were earned by women. Judging by these numbers, girls and women have always found mathematics attractive. Why not assume mathematicians have been doing something right and have been doing it right for a long time?

My hypothesis, at least as plausible as what I read elsewhere, is this: to make computer science more attractive to women, make it more like mathematics. Computing, despite the layer upon layer of gadgetry that accumulates like sediment at an archeological site, has its basis in mathematics. How do we get to it? First, teach any girl with an aptitude for symbol manipulation how to program. Teach girls, I say, not to search the Web, use a word processor, install an operating system or, God help us, play computer games. Teach girls how to program. To write a program, like solving a math problem, is to discover a pattern with logic. If girls can do mathematics, and they manifestly can, they can program.

Second, when you teach girls how to program, keep things as close to pure logic as possible. Minimize reliance on glitzy software packages, fancy graphical user interfaces, and wildly powerful and complex text editors. This advice is contrary to current practice, even to common sense, but is absolutely correct. These tools form a shell that puts logic at a distance. This is nowhere more clear than with the Linux phenomenon. It is a tinkerer's paradise. It does not surprise me that I've yet to meet a young woman obsessed, as so many young men are, with Linux arcana. To increase the number of women in the field, remove some of the layers. This will not at all harm computer science education and could well have the salutary effect of producing graduates, men as well as women, who can write clear, clean, precise code.

Third, if at all possible, teach computing without microcomputers. Again, this is contrary to received wisdom, but two decades in the field have taught me that microcomputers attract tinkering boys like bees to flowers. Girls, until that glorious day when they begin spending afternoons in the tool aisles at Home Depot, will be driven away. Remove microcomputers and you decrease the distraction from hardware. We are training systems designers, after all; software engineers, not computer technicians.

Fourth, keep the programs short, at least in the early stages. One of the striking features about mathematics education is its reliance on drill. Page through any calculus book and what you will find are thousands upon thousands of nearly identical, comparatively simple problems. Mathematics, to paraphrase a colleague, is the only discipline whose gate is kept by an army of five-minute exercises. Though drill is out of fashion with mathematics reformers, I have long thought it one of the field's charms. Anyone who likes mathematics knows the pleasure of working these problems. They are difficult enough to make one think, but not so difficult as to make one think too much. That is to say, anyone with a reasonable set of gifts can get his or her brain around your garden-variety calculus problem. Since there is something about the determinate nature of mathematics that seems to appeal to girls, I suggest we try to make computing more determinate. Instead of requiring long programming assignments that students must design, code, debug, test, and document, ask students to write many, many small, nearly identical functions. Once they have mastered this skill, they will feel, like students of mathematics, confident about going on to the next level.

Fifth, treat programming languages as notational systems. This means that you should resist the temptation to adopt a new one, no matter how extravagant the promises of its devotees. Remember, the goal here is to try to interest girls in computing. It could well be that Java is the future (as C++ once was, as C once was, as Pascal once was, as, even, Cobol once was), the language of choice for sophisticated systems. Thirteen-year-olds, seventeen-year-olds, or even nineteen-year-olds, don't produce sophisticated systems. However, the way things stand now, they produce simple systems with fabulously complex tools. Remember mathematics. Remember that young women received nearly 40% of the bachelor's degrees without the benefit of feminism. To keep women involved, agree on a programming language appropriate to the task at hand, and don't change it—at least until students have developed a good deal of sophistication.

All of this is speculation, of course. But it is surely no more speculative than exhortations to build girl-friendly computer games[3] or to "prepare tech-savvy teachers".[1] That girls may be drawn to logic more readily than to variations of Mattel's "Barbie Fashion Designer," is counterintuitive. It is certainly out of fashion. Yet a program based on this observation is easy to put together. It requires no grants, no consultants, no expensive outlays for still more equipment. Nor do we have to convince toy manufacturers that women hold up half the sky. We don't even have to reform the high school curriculum. It's odd, I admit, that mathematics, a discipline obsessed with prodigies and timed examinations, would prove so friendly to women. But the numbers are there for anyone to see. Computer science might be looking in all the wrong places.

References

1. American Association of University Women. Tech-savvy: Educating girls in the new computer age, 2000; www.aauw.org/2000/techsavvy.html.
2. Camp, T. The incredible shrinking pipeline. *Commun. ACM 40,* 10 (Oct. 1997), 103–110.
3. Gorriz, C., Medina, C. Engaging girls with computers through software games. *Commun ACM 43,* 1 (Jan. 2000), 42–49.
4. McIlwee, J., Robinson, J.G. *Women in Engineering: Gender, Power and Workplace Culture.* State University of New York Press, Albany, 1992.
5. National Center for Educational Statistics. *Digest of Educational Statistics NCES 98-015.* U.S. Government Printing Office, Washington, D.C., 1997.
6. Oechtering, V., Behnke, R. Situations and Advancement Measures in Germany. *Commun. ACM 38,* 1 (Jan. 1995), 75–82.
7. Perlow, L. *Finding Time: How Corporations, Individuals and Families Can Benefit from New Work Practices.* Cornell University Press, Ithaca, NY, 1997.

Paul De Palma (depalma@gonzaga.edu) is an associate professor of mathematics and computer science at Gonzaga University, Spokane, WA.

Reprinted by permission of the *ABA Journal,* August 1999, pp. 52-56. © 1999 by the American Bar Association.