The Secret History of Women in Coding

Computer programming once had much better gender balance than it does today. What went wrong?

By Clive Thompson

Feb. 13, 2019

As a teenager in Maryland in the 1950s, Mary Allen Wilkes had no plans to become a software pioneer — she dreamed of being a litigator. One day in junior high in 1950, though, her geography teacher surprised her with a comment: "Mary Allen, when you grow up, you should be a computer programmer!" Wilkes had no idea what a programmer was; she wasn't even sure what a computer was. Relatively few Americans were. The first digital computers had been built barely a decade earlier at universities and in government labs.

By the time she was graduating from Wellesley College in 1959, she knew her legal ambitions were out of reach. Her mentors all told her the same thing: Don't even bother applying to law school. "They said: 'Don't do it. You may not get in. Or if you get in, you may not get out. And if you get out, you won't get a job,' " she recalls. If she lucked out and got hired, it wouldn't be to argue cases in front of a judge. More likely, she would be a law librarian, a legal secretary, someone processing trusts and estates.

But Wilkes remembered her junior high school teacher's suggestion. In college, she heard that computers were supposed to be the key to the future. She knew that the Massachusetts Institute of Technology had a few of them. So on the day of her graduation, she had her parents drive her over to M.I.T. and marched into the school's employment office. "Do you have any jobs for computer programmers?" she asked. They did, and they hired her.

It might seem strange now that they were happy to take on a random applicant with absolutely no experience in computer programming. But in those days, almost nobody had any experience writing code. The discipline did not yet really exist; there were vanishingly few college courses in it, and no majors. (Stanford, for example, didn't create a computer-science department until 1965.) So instead, institutions that needed programmers just used aptitude tests to evaluate applicants' ability to think logically. Wilkes happened to have some intellectual preparation: As a philosophy major, she had studied symbolic logic, which can involve creating arguments and inferences by stringing together and/or statements in a way that resembles coding.

Wilkes quickly became a programming whiz. She first worked on the IBM 704, which required her to write in an abstruse "assembly language." (A typical command might be something like "LXA A, K," telling the computer to take the number in Location A of its memory and load it into to the "Index Register" K.) Even getting the program into the IBM 704 was a laborious affair. There were no keyboards or screens; Wilkes had to write a program on paper and give it to a typist, who translated each command into holes on a punch card. She would carry boxes of commands to an "operator," who then fed a stack of such cards into a reader. The computer executed the program and produced results, typed out on a printer.

Often enough, Wilkes's code didn't produce the result she wanted. So she had to pore over her lines of code, trying to deduce her mistake, stepping through each line in her head and envisioning how the machine would execute it — turning her mind, as it were, into the computer. Then she would rewrite the program. The capacity of most computers at the time was quite limited; the IBM 704 could handle only about 4,000 "words" of code in its memory. A good programmer was concise and elegant and never wasted a word. They were poets of bits. "It was like working logic puzzles — big, complicated logic puzzles," Wilkes says. "I still have a very picky, precise mind, to a fault. I notice pictures that are crooked on the wall."

What sort of person possesses that kind of mentality? Back then, it was assumed to be women. They had already played a foundational role in the prehistory of computing: During World War II, women operated some of the first computational machines used for code-breaking at Bletchley Park in Britain. In the United States, by 1960, according to government statistics, more than one in four programmers were women. At M.I.T.'s Lincoln Labs in the 1960s, where Wilkes worked, she recalls that most of those the government categorized as "career programmers" were female. It wasn't high-status work — yet.

In 1961, Wilkes was assigned to a prominent new project, the creation of the LINC. As one of the world's first interactive personal computers, it would be a breakthrough device that could fit in a single office or lab. It would even have its own keyboard and screen, so it could be programmed more quickly, without awkward punch cards or printouts. The designers, who knew they could make the hardware, needed Wilkes to help write the software that would let a user control the computer in real time.
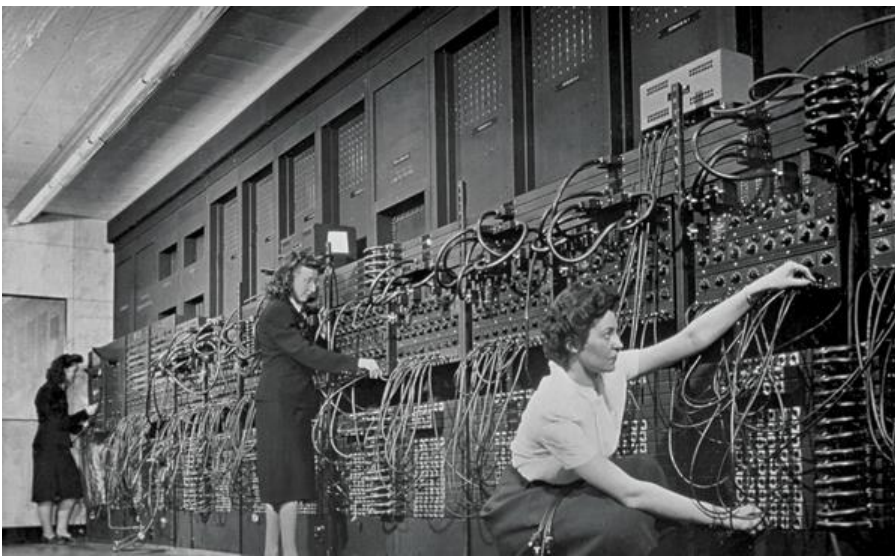
*Computer operators with an Eniac — the world's first programmable general-purpose computer.*CreditCorbis/Getty Images

For two and a half years, she and a team toiled away at flow charts, pondering how the circuitry functioned, how to let people communicate with it. "We worked all these crazy hours; we ate all kinds of terrible food,"

she says. There was sexism, yes, especially in the disparity between how men and women were paid and promoted, but Wilkes enjoyed the relative comity that existed among the men and women at Lincoln Labs, the sense of being among intellectual peers. "We were a bunch of nerds," Wilkes says dryly. "We were a bunch of geeks. We dressed like geeks. I was completely accepted by the men in my group." When they got an early prototype of the LINC working, it solved a fiendish data-processing problem for a biologist, who was so excited that he danced a happy jig around the machine.

In late 1964, after Wilkes returned from traveling around the world for a year, she was asked to finish writing the LINC's operating system. But the lab had been relocated to St. Louis, and she had no desire to move there. Instead, a LINC was shipped to her parents' house in Baltimore. Looming in the front hall near the foot of the stairs, a tall cabinet of whirring magnetic tapes across from a refrigerator-size box full of circuitry, it was an early glimpse of a sci-fi future: Wilkes was one of the first people on the planet to have a personal computer in her home. (Her father, an Episcopal clergyman, was thrilled. "He bragged about it," she says. "He would tell anybody who would listen, 'I bet you don't have a computer in your living room.' ") Before long, LINC users around the world were using her code to program medical analyses and even create a chatbot that interviewed patients about their symptoms.

But even as Wilkes established herself as a programmer, she still craved a life as a lawyer. "I also really finally got to the point where I said, 'I don't think I want to do this for the rest of my life,' " she says. Computers were intellectually stimulating but socially isolating. In 1972, she applied and got in to Harvard Law School, and after graduating, she spent the next four decades as a lawyer. "I absolutely loved it," she says.

Today Wilkes is retired and lives in Cambridge, Mass. White-haired at 81, she still has the precise mannerisms and the ready, beaming smile that can be seen in photos from the '60s, when she posed, grinning, beside the LINC. She told me that she occasionally gives talks to young students studying computer science. But the industry they're heading into is, astonishingly, less populated with women — and by many accounts less welcoming to them — than it was in Wilkes's day. In 1960, when she started working at M.I.T., the proportion of women in computing and mathematical professions (which are grouped together in federal government data) was 27 percent. It reached 35 percent in 1990. But, in the government's published figures, that was the peak. The numbers fell after that, and by 2013, women were down to 26 percent — below their share in 1960.

When Wilkes talks to today's young coders, they are often shocked to learn that women were among the field's earliest, towering innovators and once a common sight in corporate America. "Their mouths are agape," Wilkes says. "They have absolutely no idea."

Almost 200 years ago, the first person to be what we would now call a coder was, in fact, a woman: Lady Ada Lovelace. As a young mathematician in England in 1833, she met Charles Babbage, an inventor who was struggling to design what he called the Analytical Engine, which would be made of metal gears and able to execute if/then commands and store information in memory. Enthralled, Lovelace grasped the enormous potential of a device like this. A computer that could modify its own instructions and memory could be far more than a rote calculator, she realized. To prove it, Lovelace wrote what is often regarded as the first computer program in history, an algorithm with which the Analytical Engine would calculate the Bernoulli sequence of numbers. (She wasn't shy about her accomplishments: "That *brain* of mine is something more than merely mortal; as time will show," she once wrote.) But Babbage never managed to build his computer, and Lovelace, who died of cancer at 36, never saw her code executed.

When digital computers finally became a practical reality in the 1940s, women were again pioneers in writing software for the machines. At the time, men in the computing industry regarded writing code as a secondary,

less interesting task. The real glory lay in making the hardware. Software? "That term hadn't yet been invented," says Jennifer S. Light, a professor at M.I.T. who studies the history of science and technology.

An engraving of Ada Lovelace, the first computer programmer. Credit SSPL/Getty Images

This dynamic was at work in the development of the first programmable digital computer in the United States, the Electronic Numerical Integrator and Computer, or Eniac, during the 1940s. Funded by the military, the thing was a behemoth, weighing more than 30 tons and including 17,468 vacuum tubes. Merely getting it to work was seen as the heroic, manly engineering feat. In contrast, programming it seemed menial, even secretarial. Women had long been employed in the scut work of doing calculations. In the years leading up to the Eniac, many companies bought huge electronic tabulating machines — quite useful for tallying up payroll, say — from companies like IBM; women frequently worked as the punch-card operators for these overgrown calculators. When the time came to hire technicians to write instructions for the Eniac, it made sense, to the men in charge, to pick an all-female team: Kathleen McNulty, Jean Jennings, Betty Snyder, Marlyn Wescoff, Frances Bilas and Ruth Lichterman. The men would figure out what they wanted Eniac to do; the women "programmed" it to execute the instructions.

"We could diagnose troubles almost down to the individual vacuum tube," Jennings later told an interviewer for the IEEE Annals of the History of Computing. Jennings, who grew up as the tomboy daughter of low-income parents near a Missouri community of 104 people, studied math at college. "Since we knew both the application and the machine, we learned to diagnose troubles as well as, if not better than, the engineer."

The Eniac women were among the first coders to discover that software never works right the first time — and that a programmer's main work, really, is to find and fix the bugs. Their innovations included some of software's core concepts. Betty Snyder realized that if you wanted to debug a program that wasn't running

correctly, it would help to have a "break point," a moment when you could stop a program midway through its run. To this day, break points are a key part of the debugging process.

In 1946, Eniac's creators wanted to show off the computer to a group of leaders in science, technology and the military. They asked Jennings and Snyder to write a program that calculated missile trajectories. After weeks of intense effort, they and their team had a working program, except for one glitch: It was supposed to stop when the missile landed, but for some reason it kept running. The night before the demo, Snyder suddenly intuited the problem. She went to work early the next day, flipped a single switch inside the Eniac and eliminated the bug. "Betty could do more logical reasoning while she was asleep than most people can do awake," Jennings later said. Nonetheless, the women got little credit for their work. At that first official demonstration to show off Eniac, the male project managers didn't mention, much less introduce, the women.

After the war, as coding jobs spread from the military into the private sector, women remained in the coding vanguard, doing some of the highest-profile work. The pioneering programmer Grace Hopper is frequently credited with creating the first "compiler," a program that lets users create programming languages that more closely resemble regular written words: A coder could thus write the English-like code, and the compiler would do the hard work of turning it into ones and zeros for the computer. Hopper also developed the "Flowmatic" language for nontechnical businesspeople. Later, she advised the team that created the Cobol language, which became widely used by corporations. Another programmer from the team, Jean E. Sammet, continued to be influential in the language's development for decades. Fran Allen was so expert in optimizing Fortran, a popular language for performing scientific calculations, that she became the first female IBM fellow.

When the number of coding jobs exploded in the '50s and '60s as companies began relying on software to process payrolls and crunch data, men had no special advantage in being hired. As Wilkes had discovered, employers simply looked for candidates who were logical, good at math and meticulous. And in this respect, gender stereotypes worked in women's favor: Some executives argued that women's traditional expertise at painstaking activities like knitting and weaving manifested precisely this mind-set. (The 1968 book "Your Career in Computers" stated that people who like "cooking from a cookbook" make good programmers.)

The field rewarded aptitude: Applicants were often given a test (typically one involving pattern recognition), hired if they passed it and trained on the job, a process that made the field especially receptive to neophytes. "Know Nothing About Computers? Then We'll Teach You (and Pay You While Doing So)," one British ad promised in 1965. In a 1957 recruiting pitch in the United States, IBM's brochure titled "My Fair Ladies" specifically encouraged women to apply for coding jobs.

Such was the hunger for programming talent that a young black woman named Arlene Gwendolyn Lee could become one of the early female programmers in Canada, despite the open discrimination of the time. Lee was half of a biracial couple to whom no one would rent, so she needed money to buy a house. According to her son, who has described his mother's experience in a blog post, Lee showed up at a firm after seeing its ad for data processing and systems analytics jobs in a Toronto newspaper sometime in the early 1960s. Lee persuaded the employers, who were all white, to let her take the coding aptitude test. When she placed in the 99th percentile, the supervisors grilled her with questions before hiring her. "I had it easy," she later told her son. "The computer didn't care that I was a woman or that I was black. Most women had it much harder."

Elsie Shutt learned to code during her college summers while working for the military at the Aberdeen Proving Ground, an Army facility in Maryland. In 1953, while taking time off from graduate school, she was hired to code for Raytheon, where the programmer work force "was about 50 percent men and 50 percent women,"

she told Janet Abbate, a Virginia Tech historian and author of the 2012 book "Recoding Gender." "And it really amazed me that these men were programmers, because I thought it was women's work!"

When Shutt had a child in 1957, state law required her to leave her job; the '50s and '60s may have been welcoming to full-time female coders, but firms were unwilling to offer part-time work, even to superb coders. So Shutt founded Computations Inc., a consultancy that produced code for corporations. She hired stay-at-home mothers as part-time employees; if they didn't already know how to code, she trained them. They cared for their kids during the day, then coded at night, renting time on local computers. "What it turned into was a feeling of mission," Shutt told Abbate, "in providing work for women who were talented and did good work and couldn't get part-time jobs." Business Week called the Computations work force the "pregnant programmers" in a 1963 article illustrated with a picture of a baby in a bassinet in a home hallway, with the mother in the background, hard at work writing software. (The article's title: "Mixing Math and Motherhood.")

By 1967, there were so many female programmers that Cosmopolitan magazine published an article about "The Computer Girls," accompanied by pictures of beehived women at work on computers that evoked the control deck of the U.S.S. Enterprise. The story noted that women could make $20,000 a year doing this work (or more than $150,000 in today's money). It was the rare white-collar occupation in which women could thrive. Nearly every other highly trained professional field admitted few women; even women with math degrees had limited options: teaching high school math or doing rote calculations at insurance firms.

"Women back then would basically go, 'Well, if I don't do programming, what else will I do?' " Janet Abbate says. "The situation was very grim for women's opportunities."

If we want to pinpoint a moment when women began to be forced out of programming, we can look at one year: 1984. A decade earlier, a study revealed that the numbers of men and women who expressed an interest in coding as a career were equal. Men were more likely to enroll in computer-science programs, but women's participation rose steadily and rapidly through the late '70s until, by the 1983-84 academic year, 37.1 percent of all students graduating with degrees in computer and information sciences were women. In only one decade, their participation rate more than doubled.

But then things went into reverse. From 1984 onward, the percentage dropped; by the time 2010 rolled around, it had been cut in half. Only 17.6 percent of the students graduating from computer-science and information-science programs were women.

One reason for this vertiginous decline has to do with a change in how and when kids learned to program. The advent of personal computers in the late '70s and early '80s remade the pool of students who pursued computer-science degrees. Before then, pretty much every student who showed up at college had never touched a computer or even been in the room with one. Computers were rare and expensive devices, available for the most part only in research labs or corporate settings. Nearly all students were on equal footing, in other words, and new to programming.

Once the first generation of personal computers, like the Commodore 64 or the TRS-80, found their way into homes, teenagers were able to play around with them, slowly learning the major concepts of programming in their spare time. By the mid-'80s, some college freshmen were showing up for their first class already proficient as programmers. They were remarkably well prepared for and perhaps even a little jaded about what Computer Science 101 might bring. As it turned out, these students were mostly men, as two academics discovered when they looked into the reasons women's enrollment was so low.

Keypunch operators at IBM in Stockholm in the 1930s .Credit IBM

One researcher was Allan Fisher, then the associate dean of the computer-science school at Carnegie Mellon University. The school established an undergraduate program in computer science in 1988, and after a few years of operation, Fisher noticed that the proportion of women in the major was consistently below 10 percent. In 1994, he hired Jane Margolis, a social scientist who is now a senior researcher in the U.C.L.A. School of Education and Information Studies, to figure out why. Over four years, from 1995 to 1999, she and her colleagues interviewed and tracked roughly 100 undergraduates, male and female, in Carnegie Mellon's computer-science department; she and Fisher later published the findings in their 2002 book "Unlocking the Clubhouse: Women in Computing."

What Margolis discovered was that the first-year students arriving at Carnegie Mellon with substantial experience were almost all male. They had received much more exposure to computers than girls had; for example, boys were more than twice as likely to have been given one as a gift by their parents. And if parents bought a computer for the family, they most often put it in a son's room, not a daughter's. Sons also tended to have what amounted to an "internship" relationship with fathers, working through Basic-language manuals with them, receiving encouragement from them; the same wasn't true for daughters. "That was a very important part of our findings," Margolis says. Nearly every female student in computer science at Carnegie Mellon told Margolis that her father had worked with her brother — "and they had to fight their way through to get some attention."

Their mothers were typically less engaged with computers in the home, they told her. Girls, even the nerdy ones, picked up these cues and seemed to dial back their enthusiasm accordingly. These were pretty familiar roles for boys and girls, historically: Boys were cheered on for playing with construction sets and electronics kits, while girls were steered toward dolls and toy kitchens. It wasn't terribly surprising to Margolis that a new technology would follow the same pattern as it became widely accepted.

At school, girls got much the same message: Computers were for boys. Geeky boys who formed computer clubs, at least in part to escape the torments of jock culture, often wound up, whether intentionally or not, reproducing the same exclusionary behavior. (These groups snubbed not only girls but also black and Latino boys.) Such male cliques created "a kind of peer support network," in Fisher's words.

This helped explain why Carnegie Mellon's first-year classes were starkly divided between the sizable number of men who were already confident in basic programming concepts and the women who were frequently complete neophytes. A cultural schism had emerged. The women started doubting their ability. How would they ever catch up?

What Margolis heard from students — and from faculty members, too — was that there was a sense in the classroom that if you hadn't *already* been coding obsessively for years, you didn't belong. The "real programmer" was the one who "had a computer-screen tan from being in front of the monitor all the time," as Margolis puts it. "The idea was, you just have to love being with a computer all the time, and if you don't do it 24/7, you're not a 'real' programmer." The truth is, many of the men themselves didn't fit this monomaniacal stereotype. But there was a double standard: While it was O.K. for the men to want to engage in various other pursuits, women who expressed the same wish felt judged for not being "hard core" enough. By the second year, many of these women, besieged by doubts, began dropping out of the program. (The same was true for the few black and Latino students who also arrived on campus without teenage programming experience.)

A similar pattern took hold at many other campuses. Patricia Ordóñez, a first-year student at Johns Hopkins University in 1985, enrolled in an Introduction to Minicomputers course. She had been a math whiz in high school but had little experience in coding; when she raised her hand in class at college to ask a question, many of the other students who had spent their teenage years programming — and the professor — made her feel singled out. "I remember one day he looked at me and said, 'You should already know this by now,' " she told me. "I thought, I'm never going to succeed." She switched majors as a result.

Yet a student's decision to stick with or quit the subject did not seem to be correlated with coding talent. Many of the women who dropped out were getting perfectly good grades, Margolis learned. Indeed, some who left had been top students. And the women who did persist and made it to the third year of their program had by then generally caught up to the teenage obsessives. The degree's coursework was, in other words, a leveling force. Learning Basic as a teenage hobby might lead to lots of fun and useful skills, but the pace of learning at college was so much more intense that by the end of the degree, everyone eventually wound up graduating at roughly the same levels of programming mastery.

"It turned out that having prior experience is not a great predictor, even of academic success," Fisher says. Ordóñez's later experience illustrates exactly this: After changing majors at Johns Hopkins, she later took night classes in coding and eventually got a Ph.D. in computer science in her 30s; today, she's a professor at the University of Puerto Rico Río Piedras, specializing in data science.

By the '80s, the early pioneering work done by female programmers had mostly been forgotten. In contrast, Hollywood was putting out precisely the opposite image: Computers were a male domain. In hit movies like "Revenge of the Nerds," "Weird Science," "Tron," "WarGames" and others, the computer nerds were nearly always young white men. Video games, a significant gateway activity that led to an interest in computers, were pitched far more often at boys, as research in 1985 by Sara Kiesler, a professor at Carnegie Mellon, found. "In the culture, it became something that guys do and are good at," says Kiesler, who is also a program

manager at the National Science Foundation. "There were all kinds of things signaling that if you don't have the right genes, you're not welcome."

An E.R.A./Univac 1103 computer in the 1950s.CreditHum Images/Alamy

A 1983 study involving M.I.T. students produced equally bleak accounts. Women who raised their hands in class were often ignored by professors and talked over by other students. They would be told they weren't aggressive enough; if they challenged other students or contradicted them, they heard comments like "You sure are bitchy today — must be your period." Behavior in some research groups "sometimes approximates that of the locker room," the report concluded, with men openly rating how "cute" their female students were. ("Gee, I don't think it's fair that the only two girls in the group are in the same office," one said. "We should share.") Male students mused about women's mediocrity: "I really don't think the woman students around here are as good as the men," one said.

By then, as programming enjoyed its first burst of cultural attention, so many students were racing to enroll in computer science that universities ran into a supply problem: They didn't have enough professors to teach everyone. Some added hurdles, courses that students had to pass before they could be accepted into the computer-science major. Punishing workloads and classes that covered the material at a lightning pace weeded out those who didn't get it immediately. All this fostered an environment in which the students mostly likely to get through were those who had already been exposed to coding — young men, mostly. "Every time the field has instituted these filters on the front end, that's had the effect of reducing the participation of women in particular," says Eric S. Roberts, a longtime professor of computer science, now at Reed College, who first studied this problem and called it the "capacity crisis."

When computer-science programs began to expand again in the mid-'90s, coding's culture was set. Most of the incoming students were men. The interest among women never recovered to the levels reached in the late '70s and early '80s. And the women who did show up were often isolated. In a room of 20 students, perhaps five or even fewer might be women.

In 1991, Ellen Spertus, now a computer scientist at Mills College, published a report on women's experiences in programming classes. She cataloged a landscape populated by men who snickered about the presumed inferiority of women and by professors who told female students that they were "far too pretty" to be

studying electrical engineering; when some men at Carnegie Mellon were asked to stop using pictures of naked women as desktop wallpaper on their computers, they angrily complained that it was censorship of the sort practiced by "the Nazis or the Ayatollah Khomeini."

As programming was shutting its doors to women in academia, a similar transformation was taking place in corporate America. The emergence of what would be called "culture fit" was changing the who, and the why, of the hiring process. Managers began picking coders less on the basis of aptitude and more on how well they fit a personality type: the acerbic, aloof male nerd.

The shift actually began far earlier, back in the late '60s, when managers recognized that male coders shared a growing tendency to be antisocial isolates, lording their arcane technical expertise over that of their bosses. Programmers were "often egocentric, slightly neurotic," as Richard Brandon, a well-known computer-industry analyst, put it in an address at a 1968 conference, adding that "the incidence of beards, sandals and other symptoms of rugged individualism or nonconformity are notably greater among this demographic."

In addition to testing for logical thinking, as in Mary Allen Wilkes's day, companies began using personality tests to select specifically for these sorts of caustic loner qualities. "These became very powerful narratives," says Nathan Ensmenger, a professor of informatics at Indiana University, who has studied this transition. The hunt for that personality type cut women out. Managers might shrug and accept a man who was unkempt, unshaven and surly, but they wouldn't tolerate a woman who behaved the same way. Coding increasingly required late nights, but managers claimed that it was too unsafe to have women working into the wee hours, so they forbid them to stay late with the men.

At the same time, the old hierarchy of hardware and software became inverted. Software was becoming a critical, and lucrative, sector of corporate America. Employers increasingly hired programmers whom they could envision one day ascending to key managerial roles in programming. And few companies were willing to put a woman in charge of men. "They wanted people who were more aligned with management," says Marie Hicks, a historian at the Illinois Institute of Technology. "One of the big takeaways is that technical skill does not equate to success."

By the 1990s and 2000s, the pursuit of "culture fit" was in full force, particularly at start-ups, which involve a relatively small number of people typically confined to tight quarters for long hours. Founders looked to hire people who were socially and culturally similar to them.

"It's all this loosey-goosey 'culture' thing," says Sue Gardner, former head of the Wikimedia Foundation, the nonprofit that hosts Wikipedia and other sites. After her stint there, Gardner decided to study why so few women were employed as coders. In 2014, she surveyed more than 1,400 women in the field and conducted sit-down interviews with scores more. It became clear to her that the occupation's takeover by men in the '90s had turned into a self-perpetuating cycle. Because almost everyone in charge was a white or Asian man, that was the model for whom to hire; managers recognized talent only when it walked and talked as they did. For example, many companies have relied on whiteboard challenges when hiring a coder — a prospective employee is asked to write code, often a sorting algorithm, on a whiteboard while the employers watch. This sort of thing bears almost no resemblance to the work coders actually do in their jobs. But whiteboard questions resemble classroom work at Ivy League institutions. It feels familiar to the men doing the hiring, many of whom are only a few years out of college. "What I came to realize," Gardner says, "is that it's not that women are excluded. It's that practically everyone is excluded if you're not a young white or Asian man who's single."

One coder, Stephanie Hurlburt, was a stereotypical math nerd who had deep experience working on graphics software. "I love C++, the low-level stuff," she told me, referring to a complex language known for allowing programmers to write very fast-running code, useful in graphics. Hurlburt worked for a series of firms this decade, including Unity (which makes popular software for designing games), and then for Facebook on its Oculus Rift VR headset, grinding away for long hours in the run-up to the release of its first demo. Hurlburt became accustomed to shrugging off negative attention and crude sexism. She heard, including from many authority figures she admired, that women weren't wired for math. While working as a coder, if she expressed ignorance of any concept, no matter how trivial, male colleagues would disparage her. "I thought you were at a higher math level," one sniffed.

In 2016, Hurlburt and a friend, Rich Geldreich, founded a start-up called Binomial, where they created software that helps compress the size of "textures" in graphics-heavy software. Being self-employed, she figured, would mean not having to deal with belittling bosses. But when she and Geldreich went to sell their product, some customers assumed that she was just the marketing person. "I don't know how you got this product off the ground when you only have one programmer!" she recalls one client telling Geldreich.

In 2014, an informal analysis by a tech entrepreneur and former academic named Kieran Snyder of 248 corporate performance reviews for tech engineers determined that women were considerably more likely than men to receive reviews with negative feedback; men were far more likely to get reviews that had only constructive feedback, with no negative material.

Lurking beneath some of this sexist atmosphere is the phantasm of sociobiology. As this line of thinking goes, women are less suited to coding than men because biology better endows men with the qualities necessary to excel at programming. Many women who work in software face this line of reasoning all the time. Cate Huston, a software engineer at Google from 2011 to 2014, heard it from colleagues there when they pondered why such a low percentage of the company's programmers were women. Peers would argue that Google hired only the best — that if women weren't being hired, it was because they didn't have enough innate logic or grit, she recalls.

In the summer of 2017, a Google employee named James Damore suggested in an internal email that several qualities more commonly found in women — including higher rates of anxiety — explained why they weren't thriving in a competitive world of coding; he cited the cognitive neuroscientist Simon Baron-Cohen, who theorizes that the male brain is more likely to be "systemizing," compared with women's "empathizing" brains. Google fired Damore, saying it could not employ someone who would argue that his female colleagues were inherently unsuited to the job. But on Google's internal boards, other male employees backed up Damore, agreeing with his analysis. The assumption that the makeup of the coding work force reflects a pure meritocracy runs deep among many Silicon Valley men; for them, sociobiology offers a way to explain things, particularly for the type who prefers to believe that sexism in the workplace is not a big deal, or even doubts it really exists.

But if biology were the reason so few women are in coding, it would be impossible to explain why women were so prominent in the early years of American programming, when the work could be, if anything, far harder than today's programming. It was an uncharted new field, in which you had to do math in binary and hexadecimal formats, and there were no helpful internet forums, no Google to query, for assistance with your bug. It was just your brain in a jar, solving hellish problems.

If biology limited women's ability to code, then the ratio of women to men in programming ought to be similar in other countries. It isn't. In India, roughly 40 percent of the students studying computer science and related

fields are women. This is despite even greater barriers to becoming a female coder there; India has such rigid gender roles that female college students often have an 8 p.m. curfew, meaning they can't work late in the computer lab, as the social scientist Roli Varma learned when she studied them in 2015. The Indian women had one big cultural advantage over their American peers, though: They were far more likely to be encouraged by their parents to go into the field, Varma says. What's more, the women regarded coding as a safer job because it kept them indoors, lessening their exposure to street-level sexual harassment. It was, in other words, considered normal in India that women would code. The picture has been similar in Malaysia, where in 2001 — precisely when the share of American women in computer science had slid into a trough — women represented 52 percent of the undergraduate computer-science majors and 39 percent of the Ph.D. candidates at the University of Malaya in Kuala Lumpur.

Today, when midcareer women decide that Silicon Valley's culture is unlikely to change, many simply leave the industry. When Sue Gardner surveyed those 1,400 women in 2014, they told her the same story: In the early years, as junior coders, they looked past the ambient sexism they encountered. They loved programming and were ambitious and excited by their jobs. But over time, Gardner says, "they get ground down." As they rose in the ranks, they found few, if any, mentors. Nearly two-thirds either experienced or witnessed harassment, she read in "The Athena Factor" (a 2008 study of women in tech); in Gardner's survey, one-third reported that their managers were more friendly toward and gave more support to their male co-workers. It's often assumed that having children is the moment when women are sidelined in tech careers, as in many others, but Gardner discovered that wasn't often the breaking point for these women. They grew discouraged seeing men with no better or even lesser qualifications get superior opportunities and treatment.

"What surprised me was that they felt, 'I did all that work!' They were *angry*," Gardner says. "It wasn't like they needed a helping hand or needed a little extra coaching. They were mad. They were not leaving because they couldn't hack it. They were leaving because they were skilled professionals who had skills that were broadly in demand in the marketplace, and they had other options. So they're like, '[expletive] it — I'll go somewhere where I'm seen as valuable.' "

The result is an industry that is drastically more male than it was decades ago, and far more so than the workplace at large. In 2018, according to data from the Bureau of Labor Statistics, about 26 percent of the workers in "computer and mathematical occupations" were women. The percentages for people of color are similarly low: Black employees were 8.4 percent, Latinos 7.5 percent. (The Census Bureau's American Community Survey put black coders at only 4.7 percent in 2016.) In the more rarefied world of the top Silicon Valley tech firms, the numbers are even more austere: A 2017 analysis by Recode, a news site that covers the technology industry, revealed that 20 percent of Google's technical employees were women, while only 1 percent were black and 3 percent were Hispanic. Facebook was nearly identical; the numbers at Twitter were 15 percent, 2 percent and 4 percent, respectively.

The reversal has been profound. In the early days of coding, women flocked to programming because it offered more opportunity and reward for merit, more than fields like law. Now software has the closed door.

In the late 1990s, Allan Fisher decided that Carnegie Mellon would try to address the male-female imbalance in its computer-science program. Prompted by Jane Margolis's findings, Fisher and his colleagues instituted several changes. One was the creation of classes that grouped students by experience: The kids who had been coding since youth would start on one track; the newcomers to coding would have a slightly different curriculum, allowing them more time to catch up. Carnegie Mellon also offered extra tutoring to all students, which was particularly useful for the novice coders. If Fisher could get them to stay through the first and second years, he knew, they would catch up to their peers.

Components from four of the earliest electronic computers, held by Patsy Boyce Simmers, Gail Taylor, Millie Beck and Norma Stec, employees at the United States Army's Ballistics Research Laboratory. Credit Science Source

They also modified the courses in order to show how code has impacts in the real world, so a new student's view of programming wouldn't just be an endless vista of algorithms disconnected from any practical use. Fisher wanted students to glimpse, earlier on, what it was like to make software that works its way into people's lives. Back in the '90s, before social media and even before the internet had gone mainstream, the influence that code could have on daily life wasn't so easy to see.

Faculty members, too, adopted a different perspective. For years some had tacitly endorsed the idea that the students who came in already knowing code were born to it. Carnegie Mellon "rewarded the obsessive hacker," Fisher told me. But the faculty now knew that their assumptions weren't true; they had been confusing previous experience with raw aptitude. They still wanted to encourage those obsessive teenage coders, but they had come to understand that the neophytes were just as likely to bloom rapidly into remarkable talents and deserved as much support. "We had to broaden how faculty sees what a successful student looks like," he says. The admissions process was adjusted, too; it no longer gave as much preference to students who had been teenage coders.

No single policy changed things. "There's really a virtuous cycle," Fisher says. "If you make the program accommodate people with less experience, then people with less experience come in." Faculty members became more used to seeing how green coders evolve into accomplished ones, and they learned how to teach that type.

Carnegie Mellon's efforts were remarkably successful. Only a few years after these changes, the percentage of women entering its computer-science program boomed, rising to 42 percent from 7 percent; graduation rates for women rose to nearly match those of the men. The school vaulted over the national average. Other schools concerned about the low number of female students began using approaches similar to Fisher's. In 2006, Harvey Mudd College tinkered with its Introduction to Computer Science course, creating a track specifically for novices, and rebranded it as Creative Problem Solving in Science and Engineering Using

Computational Approaches — which, the institution's president, Maria Klawe, told me, "is actually a better description of what you're actually doing when you're coding." By 2018, 54 percent of Harvey Mudd's graduates who majored in computer science were women.

A broader cultural shift has accompanied the schools' efforts. In the last few years, women's interest in coding has begun rapidly rising throughout the United States. In 2012, the percentage of female undergraduates who plan to major in computer science began to rise at rates not seen for 35 years, since the decline in the mid-'80s, according to research by Linda Sax, an education professor at U.C.L.A. There has also been a boomlet of groups and organizations training and encouraging underrepresented cohorts to enter the field, like Black Girls Code and Code Newbie. Coding has come to be seen, in purely economic terms, as a bastion of well-paying and engaging work.

In an age when Instagram and Snapchat and iPhones are part of the warp and weft of life's daily fabric, potential coders worry less that the job will be isolated, antisocial and distant from reality. "Women who see themselves as creative or artistic are more likely to pursue computer science today than in the past," says Sax, who has pored over decades of demographic data about the students in STEM fields. They're still less likely to go into coding than other fields, but programming is increasingly on their horizon. This shift is abetted by the fact that it's much easier to learn programming without getting a full degree, through free online coding schools, relatively cheaper "boot camps" or even meetup groups for newcomers — opportunities that have emerged only in the last decade.

Changing the culture at schools is one thing. Most female veterans of code I've spoken to say that what is harder is shifting the culture of the industry at large, particularly the reflexive sexism and racism still deeply ingrained in Silicon Valley. Some, like Sue Gardner, sometimes wonder if it's even ethical for her to encourage young women to go into tech. She fears they'll pour out of computer-science programs in increasing numbers, arrive at their first coding job excited and thrive early on, but then gradually get beaten down by industry. "The truth is, we can attract more and different people into the field, but they're just going to hit that wall in midcareer, unless we change how things happen higher up," she says.

On a spring weekend in 2017, more than 700 coders and designers were given 24 hours to dream up and create a new product at a hackathon in New York hosted by TechCrunch, a news site devoted to technology and Silicon Valley. At lunchtime on Sunday, the teams presented their creations to a panel of industry judges, in a blizzard of frantic elevator pitches. There was Instagrammie, a robot system that would automatically recognize the mood of an elderly relative or a person with limited mobility; there was Waste Not, an app to reduce food waste. Most of the contestants were coders who worked at local high-tech firms or computer-science students at nearby universities.

Despite women's historical role in the vanguard of computer programing, some female veterans of code wonder if it's even ethical to encourage young women to go into tech because of the reflexive sexism in the current culture of Silicon Valley.
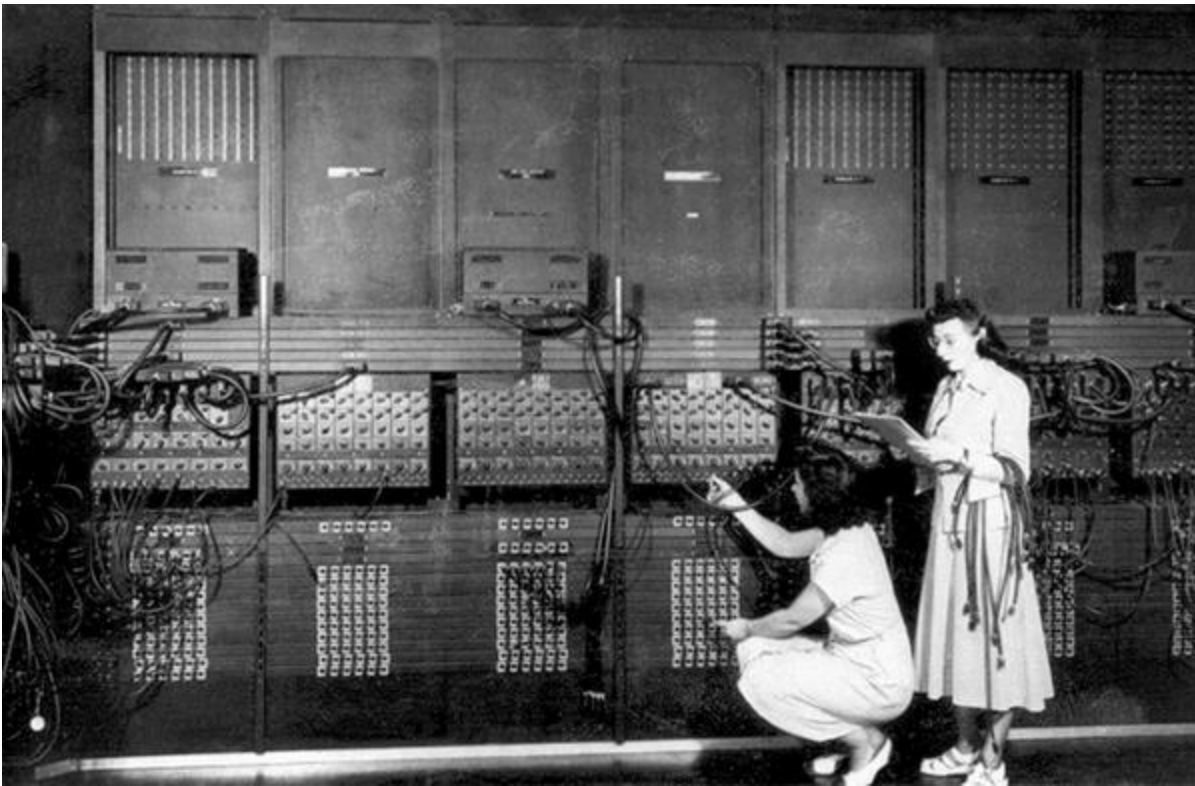
Despite women's historical role in the vanguard of computer programing, some female veterans of code wonder if it's even ethical to encourage young women to go into tech because of the reflexive sexism in the current culture of Silicon Valley. Credit Apic/Getty Images

The winning team, though, was a trio of high school girls from New Jersey: Sowmya Patapati, Akshaya Dinesh and Amulya Balakrishnan. In only 24 hours, they created reVIVE, a virtual-reality app that tests children for signs of A.D.H.D. After the students were handed their winnings onstage — a trophy-size check for $5,000 — they flopped into chairs in a nearby room to recuperate. They had been coding almost nonstop since noon the day before and were bleary with exhaustion.

"Lots of caffeine," Balakrishnan, 17, said, laughing. She wore a blue T-shirt that read "WHO HACK THE WORLD? GIRLS." The girls told me that they had impressed even themselves by how much they accomplished in 24 hours. "Our app really does streamline the process of detecting A.D.H.D.," said Dinesh, who was also 17. "It usually takes six to nine months to diagnose, and thousands of dollars! We could do it digitally in a much faster way!"

They all became interested in coding in high school, each of them with strong encouragement from immigrant parents. Balakrishnan's parents worked in software and medicine; Dinesh's parents came to the United States from India in 2000 and worked in information technology. Patapati immigrated from India as an infant with her young mother, who never went to college, and her father, an information-tech worker who was the first in his rural family to go to college.

Drawn to coding in high school, the young hackers got used to being the lone girl nerds at school, as Dinesh told me.

"I tried so hard to get other girls interested in computer science, and it was like, the interest levels were just so low," she says. "When I walked into my first hackathon, it was the most intimidating thing ever. I looked at a room of 80 kids: Five were girls, and I was probably the youngest person there." But she kept on competing in 25 more hackathons, and her confidence grew. To break the isolation and meet more girls in coding, she

attended events by organizations like #BuiltByGirls, which is where, a few days previously, she had met Patapati and Balakrishnan and where they decided to team up. To attend TechCrunch, Patapati, who was 16, and Balakrishnan skipped a junior prom and a friend's birthday party. "Who needs a party when you can go to a hackathon?" Patapati said.

Winning TechCrunch as a group of young women of color brought extra attention, not all of it positive. "I've gotten a lot of comments like: 'Oh, you won the hackathon because you're a girl! You're a diversity pick," Balakrishnan said. After the prize was announced online, she recalled later, "there were quite a few engineers who commented, 'Oh, it was a girl pick; obviously that's why they won.' "

Nearly two years later, Balakrishnan was taking a gap year to create a heart-monitoring product she invented, and she was in the running for $100,000 to develop it. She was applying to college to study computer science and, in her spare time, competing in a beauty pageant, inspired by Miss USA 2017, Kara McCullough, who was a nuclear scientist. "I realized that I could use pageantry as a platform to show more girls that they could embrace their femininity and be involved in a very technical, male-dominated field," she says. Dinesh, in her final year at high school, had started an all-female hackathon that now takes place annually in New York. ("The vibe was definitely very different," she says, more focused on training newcomers.)

Patapati and Dinesh enrolled at Stanford last fall to study computer science; both are interested deeply in A.I. They've noticed the subtle tensions for women in the coding classes. Patapati, who founded the Women of A.I. group with an Apple tech lead, has watched as male colleagues ignore her raised hand in group discussions or repeat something she just said as if it were their idea. "I think sometimes it's just a bias that people don't even recognize that they have," she says. "That's been really upsetting."

Dinesh says "there's absolutely a difference in confidence levels" between the male and female newcomers. The Stanford curriculum is so intense that even the relative veterans like her are scrambling: When we spoke recently, she had just spent "three all-nighters in a row" on a single project, for which students had to engineer a "print" command from scratch. At 18, she has few illusions about the road ahead. When she went to a blockchain conference, it was a sea of "middle-aged white and Asian men," she says. "I'm never going to one again," she adds with a laugh.

"My dream is to work on autonomous driving at Tesla or Waymo or some company like that. Or if I see that there's something missing, maybe I'll start my own company." She has begun moving in that direction already, having met one venture capitalist via #BuiltByGirls. "So now I know I can start reaching out to her, and I can start reaching out to other people that she might know," she says.

Will she look around, 20 years from now, to see that software has returned to its roots, with women everywhere? "I'm not really sure what will happen," she admits. "But I do think it is absolutely on the upward climb."