"Beards, Sandals, and Other Signs of Rugged Individualism": Masculine Culture within the Computing Professions

Nathan Ensmenger, Indiana University

In 1976 the MIT computer science professor Joseph Weizenbaum published *Computer Power and Human Reason*, a scathing intellectual and moral indictment of the discipline of artificial intelligence, a field that he himself had helped to establish. More than three decades later, his book continues widely read and influential, although not perhaps for the reasons that Weizenbaum had hoped or expected. It was not his carefully constructed philosophical arguments that attracted the attention of most audiences, but rather his lurid descriptions of what he regarded as one of the most dangerous and disturbing phenomena associated with the emerging technology of electronic computing; namely, the increasing prevalence of the compulsive programmer, or the "computer bum."

In computer centers around the world, Weizenbaum argued, these computer bums, "bright young men of disheveled appearance, often with sunken glowing eyes," could be discovered hunched at their computer consoles, "their arms tensed and waiting to fire their fingers, already poised to strike, at the buttons and keys on which their attention seems to be as riveted as a gambler's on the rolling dice."[1] When not otherwise transfixed by their computer screens, these compulsive programmers pored over their computer printouts "like possessed students of a cabalistic text … They work until they nearly drop, twenty, thirty hours at a time. Their food, if they arrange it, is brought to them: coffee, Cokes, sandwiches. If possible, they sleep on cots near the computer." But such respites in the real world were few and far between, and the computer bums never wandered far from their machines. They existed almost entirely in an electronic universe of their own creation, isolated from material concerns and conventional social interactions, haunting the sheltered cloisters of the computer center. "Their rumpled clothes, their unwashed and unshaven faces, and their uncombed hair all testify that they are oblivious to their bodies and to the world in which they move. They exist, at least when so engaged, only through and for the computers."[2]

For Weizenbaum, the disheveled figure of the computer bum represented the embodiment of the dehumanizing effects of pursuing computer power as an end rather than a means: deceived by the illusion of omniscience associated with mastery of this powerful technology, these wasted young men were not scientists uncovering new truths about the universe, nor engineers building useful products to benefit society, but mere junkies in search of their next fix. That such myopic and socially maladjusted tinkerers were being accorded such a prominent and influential

1 Joseph Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (W. H. Freeman, 1976), 116.
2 Ibid.

role in the construction of the essential structures of modern society was, for Weizenbaum, the dangerous and disturbing consequence of the computational imperative. These were not the type of people he wanted to be entrusted with the technological keys to the increasingly computerized kingdom.

Although Weizenbaum's *Computer Power and Human Reason* was early, authoritative, and persuasive (among its many admirers was his fellow MIT professor Sherry Turkle, who would later extend his arguments in her even more popular and influential *The Second Self)* his was not the only, or even the first, mainstream account of the compulsive programmer phenomenon.[3] At the same time Weizenbaum was deriding the obsessive tendencies of the computer bum, a powerful counter-narrative was emerging in which such single-minded focus was lauded as desirable, possibly even heroic. In the interpretation, the glowing screens in the computer centers represented not retreat from the world, but mastery over it.

The best exemplar of this alternative portrayal of the computer bum was actually published four years prior to *Computer Power and Human Reason.* In a rollicking essay in Rolling Stone magazine provocatively entitled "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums," Stewart Brand had heralded the arrival of the electronic digital computer "good news, maybe the best since psychedelics." Where Weizenbaum perceived in computerization efforts the impersonal imperatives of reactionary conservatism, of bureaucracy and authoritarian control, Brand saw revolutionary potential and the ability of individuals to appropriate Cold War technologies for the purposes of progressive social transformation.

Although the ostensible subject of his article was Spacewar!, an early video game developed by students at MIT to demonstrate the capabilities of the then-novel cathode ray tube display, Brand was clearly less interested in Spacewar! than he was in its computer "hacker" developers.[4] Brand acknowledged the double-edged bite of the "hacker" epithet (which he deemed as both "a term of derision and the ultimate compliment") but his representation of the Spacewar hackers was unambiguously positive. Yes, being a computer bum might reflect a "kind of fanaticism," but this was the fanaticism of the artist, the inventor, and the explorer. These "magnificent men in their flying machines" were "scouting a leading edge of technology." They were "brilliant," "revolutionary," and "servants in the human interest." To degree that they violated the norm of conventional society, it was as the heroic outsider or iconoclast. Anticipating the Wild West metaphors that continue to be popular within the Free Software/Open Source Software movements, Brand portrayed computer hackers as the "outlaws," "heretics," and "revolutionaries" of the modern era, fighting to bring computer power to the people.

According to Brand, Spacewar! was not just a computer game but a kind of software-Samizdat, the vehicle though which the subversive hacker subculture was smuggled into the network of ARPA-sponsored research laboratories. The result was the creation of an increasingly global community of technician-radicals. Every

3 Sherry Turkle. *The Second Self: Computers and the Human Spirit* (Simon and Schuster, 1984).

4 Stewart Brand, "SPACEWAR: Fanatic Life and Symbolic Death Among the Computer Bums." *Rolling Stone* (December 7, 1972).

night, "hundreds of computer technicians" in computer centers around the world engaged in an effectively out-of-body, "locked in life-or-death space combat computer-projected onto cathode ray tube display screens, for hours at a time, ruining their eyes, numbing their fingers in frenzied mashing of control buttons, joyously slaying their friend and wasting their employers valuable computer time." These centers were anything but the isolated social wastelands portrayed by Weizenbaum; rather, the computer center that Brand described constituted a vibrant social space, with its own "language and character, its own legends and humor." In fact, as Brand recalled it, his evening spent with the denizens of the Stanford Artificial Intelligence Laboratory was "the most bzz-bzz- busy scene I've been around since Merry Prankster Acid Tests."

These two radically different interpretations of the same phenomenon as portrayed Weizenbaum and Brand neatly capture the perplexed, ambivalent, and conflicted attitudes towards computer programming — and more specifically, computer programmers — that characterized the early decades of electronic computing.

Computer programming was, from its very origins, a mongrel discipline. Originally envisioned as low-status, clerical work, programming soon acquired a reputation as being one of the most complex, arcane, and esoteric of technical disciplines. Although associated with the emerging discipline of computer science, the majority of programmers had no academic training, and did not see themselves as scientists. (And, as the perplexed and dismayed response of Weizenbaum to the computer bums clearly illustrates, many computer scientists did not always know what to do with programmers.) Programmers clearly built things, but they generally did not regard what they did as engineering. They most often described their work and expertise using vague analogies and mixed metaphors: to many of its practitioners, programming was simultaneously art and science, high-tech and black magic, work and play.

If the discipline itself was opaque and incomprehensible to outsiders, so too were its practitioners. They soon acquired a variety of colorful appellations, such as "computer boys," "whiz kids," "hackers," "gurus" — and, of course "computer bums" — that reflected the curious mix of wonder, respect, suspicion, and contempt with which they were regarded by their contemporaries. On the one hand, the technical skills that they possessed were clearly powerful, perhaps even dangerous; on the other, their odd practices (and sometimes personal appearance) and seeming disregard for conventional social norms and authority figures made them bizarre, if fascinating characters. To many, they appeared to be as much a subculture as an occupational group. Indeed, many of the popular accounts of programmers emphasized their innate and idiosyncratic genius. "Excellent developers, like excellent musicians and artists, are born, not made," declared one industry observer, and "the number of such developers is a fixed (and tiny) percentage of the population."[5]

From a contemporary perspective, of course, the association of computer

[5] Bruce Webster, "*The Real Software Crisis*." Byte 21 (1996): 218.

programming ability with a particular personality type is familiar to the point of being cliché. Today, of course, we would call such individuals not computer bums, but computer hackers, or even more likely, computer nerds. Indeed, within a decade of the publication of *Computer Power and Human Reason* the computer nerd would became a stock character in the repertoire of American popular culture, his defining characteristics (white, male, middle-class, uncomfortable in his body and awkward around women) well-established in popular histories of computing such as Tracy Kidder's Pulitzer Prize-winning *Soul of a New Machine* (1981) and Steve Levy's *Hackers* (1984), as well as the 1983 Hollywood blockbuster *Wargames*.[6] During the boom years of the personal computer and Internet revolutions, the business and popular press embraced the nerd identity as key to success in the New Economy. Each carefully constructed "origin story" of a self-respecting high-tech entrepreneur reads as a minor variation on a formula. The "lonely nerd-turned-accidental billionaire" narrative has assumed the mantle of Great American Success Story, as exemplified in the hit PBS documentary *Triumph of the Nerds* (1996) and the Academy Award-winning *The Social Network* (2010).

Indeed, in much of popular culture the character of the computer nerd had become so hegemonic that it threatens to erase other cultural representations of scientists and engineers. In the work of the best-selling science fiction writer Neal Stephenson, for example, Isaac Newton and in his contemporaries in Royal Society are represented as early incarnations of the hacker mentality whose mannerisms and motivations are largely indistinguishable from those of the modern open source software community.[7] In the popular genre of steampunk fiction, the Industrial Revolution is reimagined as an abortive first attempt at the Computer Revolution, with Charles Babbage standing in for an early Alan Turing.[8] The perceived connection between the computer "nerdery" and mild forms of autism has stimulated retrospective diagnoses of bookish intellectuals and scientific figures from the fictional Doctor Frankenstein to Newton, Darwin, and Einstein, suggesting a direct line of descent leading directly to the contemporary computer nerd.[9] The remarkable genius and accomplishments of Thomas Edison are now compared to those of Steve Jobs, not the other way around.[10]

Like the 1970s-era computer bum, with whom he shares certain characteristics, the contemporary computer nerd is defined in primarily by his consuming obsession with technology, his lack of conventional social skills, and inattention to his physical health and appearance. Though images of both "bums" and "nerds" were more stereotypical than representative, they are historically significant for the role they played as weapons and resources in the ongoing process of the social construction of the computer professions. As I have argued extensively elsewhere, questions about the identity, expertise, and authority of computer programmers shaped many of the

6 Tracy Kidder, *The Soul of a New Machine*. (Little, Brown, 1981); Steven Levy, *Hackers : heroes of the computer revolution* (Anchor Press, 1984); *Wargames*. DVD. Directed by Wolfgang Petersen. Los Angeles, MGM Studios, 1983.

7 Neal Stephenson, *Quicksilver* (HarperCollins, 2006).

8 William Gibson and Bruce Sterling, *The Difference Engine* (Random House, 1990).

9 Benjamin Nugent, *American Nerd: the Story of My People* (Scribner, 2008).

10 Walter Isaacson, *Steve Jobs* (Simon and Schuster, 2011).

technical, managerial, and professional developments in electronic computing for the first several decades the electronic computer era.[11] The disparate responses of Weizenbaum and Brand to the character of the computer bum are both reflections of this debate and contributions to it; they were not simply describing what they thought computer programmers were like, but arguing for a particular vision of what they ought to be.

In this essay, I explore the history of the most iconic and invariable attribute of the computer nerd stereotype; namely, that he is a he. Of all of the characteristics popularly associated with the figure of the computer nerd, none is more deeply ingrained than the fact that he be male. This is not, of course, to suggest that women do not program computers; in fact, as I will argue, women played an unusually prominent role in the history of computer programming, particularly in its earliest decades. And yet computer programming today is both male dominated and hyper-masculine. Even in an era in which even the most traditionally masculine disciplines such as mathematics, physics, and engineering have opened up opportunities for women, female participation in computing remains dismal. Female enrollments in computer science have actually *decreased* over recent decades and representations of female nerds in popular film, fiction, and history are virtually nonexistent. The notorious misogyny of certain subcultures of the computing community is well documented, as is the discouraging effect that this has on female participation.[12]

To argue that a discipline is dominated by males is not necessarily to suggest that it embodies uniquely masculine characteristics. There are structural, legal, or historical reasons why certain occupations are dominated by men that have little to do with whether the work involved is essentially masculine. In the case of computer programming, however, the dominant assumption is there are certain intellectual and emotional characteristics that are associated with computer programming ability — logical, detached, narrowly focused — that also just happen to be more prevalent in males. The belief that males are much more likely to be anti-social, anti-sensual, and attracted to the "hard mastery" of arcane technology pervades even academic literature, particularly the influential work of Sherry Turkle, who provided the dominant psychoanalytic framework through which the (male) nerd personality has been interpreted.[13] More recently, the perceived association between the "programmer personality" and mild forms of autism (to the point that Asperger's is sometimes referred to as the "geek disorder" or "Silicon Valley Syndrome") have reinforced the notion that there is a natural, historical, and inevitable connection between male forms of sociability, cognition, and virtuoso computer programming ability.

In my historical analysis of the masculinization of computer programming, I will focus on three distinct but related themes. The first is that, contrary to conventional wisdom, the computer industry was initially open to women, who were particularly

11 Nathan Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (MIT Press, 2010).

12 Thomas Misa, ed. Gender Codes: Why Women Are Leaving Computing, Wiley, 2010.

13 Turkle, *The Second Self: Computers and the Human Spirit*; Sherry Turkle. *Life on the screen : identity in the age of the Internet*. New York: Simon & Schuster, 1995, 347 p.

well represented in computer programming. In fact, at its origins, computer programming was a largely feminized occupation. The male computer nerd, far from being a natural or essential form of the computer user, was in many respects a response within the early computing community to uncertainties about the occupational status and gender identity of the computer programmer, and, by extension, about the reputation of the computer industry itself.[14] A remarkable demographic shift occurred in programming over the course of 1960s and early 1970s, a shift that can be explained not only in terms of the professionalization of the discipline, but also by reference to very specific structural mechanisms, such as the use of psychometric testing in hiring process. In this respect, the history of computer programming provides new and unique insights into the different ways the gendering of institutions and practices occurs.[15]

The second intriguing feature of the history of masculinity in the programming professions has to do with the significance of specific sites of practice. Place matters, even in the history of a technology that claims to make place irrelevant. In this case, it was the university computer labs, the sheltered, unsupervised, and subsidized environments, in which the burgeoning computer hacker culture became inextricably linked with the cultural practices of adolescent masculinity. Later, as the locus of hacker activity shifted from the university mainframe to the household personal computer, these practices were recreated in other masculine spaces, such as bedrooms, basements, and dormitories. They persist today in the form of the corporate "campuses" and "play areas" (and even "tree houses") at innumerable tech firms and start-ups.[16]

The final feature of this history concerns the ways in male programmers mobilized multiple, and sometimes even competing, visions of masculine identity. Computer programmers might be predominantly male, but the masculinity of the computer nerd is hardly that of the police office or the football player — or even that of the engineer or scientist. In fact, there was no single, unified masculine ideal that male computer programmers drew upon to establish their authority or elevate their status, Some embraced the asceticism of the "compulsive programmer" while others found it repellent. Weizenbaum clearly deplored the lack of adult male "professionalism" displayed by the "bright boys" of the computer lab; for others, acting the role of the perpetually adolescent "whiz kid" was an attractive proposition. In contrast to the isolated programming nerd, the recent emergence of the frat-boy culture of "brogramming" in certain high-tech companies constitutes still another alternative form of masculinity at play in the computer industry.[17] Despite the

[14] Michael Mahoney, "Boy's Toys and Women'S Work: Feminism Engages Software." in *Feminism in Twentieth-Century Science, Technology, and Medicine*, ed. Angela Creager, Elizabeth Lunbeck, and Londa Schiebinger (University of Chicago Press, 2001).

15 Margaret Rossiter. *Women scientists in America: before affirmative action, 1940-1972* (Johns Hopkins University Press, 1995); Ruth Oldenziel. *Making Technology Masculine: Men, Women and Modern Machines in America, 1870-1945* (Amsterdam University Press, 1999).

[16] Katherine Losse, *The Boy Kings* (Simon and Schuster, 2012); Jim Edwards, "We're Jealous Of This Startup's Hammock-Filled Treehouse Office." *Fortun*e (Dec 14, 2012).

17 Marie Hicks, "De-Brogramming the History of Computing." *Annals of the History of Computing, IEEE* 35 (2013).

variety of forms that it assumed, however, many computer programmers embraced masculinity as a powerful resource for establishing their professional identity and authority.

Information Factories & Feminized Labor

The first computer programmers were women. This is well-established historical fact, and has been much celebrated in recent years by scholars both looking to uncover what Judy Wacjman has called the "hidden history" of women in technology and seeking to engage in contemporary debates about declining female enrollments in computer science programs.[18] These are important issues, but such treatments tend to represent the first female programmers as trailblazers carving out a role for women in a traditionally male-dominated field. As I have argued extensively elsewhere, however, the presence of women in computer programming is not just an historical anomaly or a reflection of a temporary wartime shortage of men; rather, computer programming was a feminized occupation from its origins.[19] In other words, the history of women in early computing says more about the history of computing than it does about women's history. The use of low-wage, low-skilled female programming labor was integral to the design of the larger socio-technical systems of early electronic computation. For the leaders of many of the pioneering computer projects, the natural assumption was that computer programming was women's work. To borrow a metaphor from computer programming itself, the presence of women in early computing was a feature, not a bug.

The shortest path to understanding the imagined role of women computing runs through the University of Pennsylvania ENIAC project, arguably the most visible and influential of the pioneering wartime electronic computer development efforts. In the ENIAC project, the task of configuring the computer to perform a specific series of operations (the task most closely analogous with the modern concept of computer programming) was performed exclusively by women. The expectation was that these women would replicate, in the electronic computer, the elaborate "plans of computation" that were already being performed by human computers in existing large-scale computational efforts. At the time, the ENIAC managers imagined the electronic computer as "nothing more than an automated form of hand computation," and therefore it seemed obvious that the same people who had directed the activities of female "computers" could also be trusted to "set-up" and monitor the operations of their electronic equivalent.[20]

The sexual division of labor at the ENIAC project also reflected contemporary beliefs about the relationship between those who build technologies and those who use them. For the male leaders of the ENIAC project, all of whom were either

18 Judy Wajcman, "Reflections on Gender and Technology: In What State is the Art?" *Social Studies of Science* 30 (2000): 447–464.

[19] Nathan Ensmenger, "Making Programming Masculine." In *Gender Codes: Why Women are Leaving Computing*, ed. Thomas Misa (Wiley, 2010): 115-141.

[20] David Alan Grier, *When Computers Were Human* (Princeton University Press, 2005).

scientists, engineers, military officers (or, more often than not, all three), the important challenges associated with the development of a working electronic computer system involved the design and construction of the actual machine. The subsequent operation of the computer was considered to be relatively trivial, and therefore work that could be successfully delegated to women. The telephone switchboard-like appearance of the cable-and-plug panels, which were used by the ENIAC programmers to "set-up" the machine, reinforced the assumption that this was work to be done by mere machine operators. The notion that such work would, in fact, be so difficult as to require the construction of an entirely new intellectual discipline, was entirely unanticipated.[21] Programming at the ENIAC project was low-status, largely invisible, and therefore generally performed by women.[22]

Of course, the use of female labor to perform routine tasks was not unique to the ENIAC, or to electronic computing as whole. The combination of mechanization, the division of labor, and a reliance on low-skilled (or at least low-wage) workers is the essence of industrialization, and in the United States at least, women were the first factory workers.[23] This was particularly true of the "information factories" that emerged in the post-Civil War period. From the multi-division firm to the modern nation state, a growing number of information-centric organizations were made possible not only by innovations in information technology, such as typewriters, tabulating machines, mechanical calculators, and vertical filing cabinets, but also by the mass-mobilization of low-wage, low-skill female labor.[24] In fact, by the beginning of the 20th century, women dominated the clerical occupations. The reinvention of the electronic computer as a business machine in the post-war period, driven by office technology firms such as IBM, Remington Rand, Burroughs, and NCR, assured that the gendered division of labor that existed in most business data processing departments was simply mapped onto the new technology of electronic computing. This too would be an office technology designed by men but used by women.

Computer programming in the 1950s was not a job performed exclusively (or even predominantly) by women, but it was nevertheless gendered female. The assumed characteristics of programming work — routine, repetitive, and highly amenable to mechanization (or so it was believed, or at least hoped, by many computer managers at the time) — meant that it was work more likely to be assigned to women that to men. In fact, as one female programmer described the situation, "It never occurred to any of us that computer programming would eventually become something that was thought of as a men's field."[25] For

[21] David Alan Grier, "The ENIAC, the verb to program, and the Emergence of Digital Computers," *Annals of the History of Computing* 18 (1996): 51-55, on 53.

22 See, for example, Jennifer Light, "When Computers Were Women." *Technology & Culture* 40 (1999): 455–483 and Nellwyn Thomas, "Selling the First Computer: The Legacy of the ENIAC's Publicity." Unpublished manuscript (2010).

[23] Thomas Dublin, *Transforming Women's Work (*Cornell University Press, 1994).

24 Margery Davies, *Woman's place is at the typewriter : office work and office workers, 1870-1930*. (Temple University Press, 1982); Sharon Hartman Strom, *Beyond the typewriter : gender, class, and the origins of modern American office work, 1900-1930* (University of Illinois Press, 1992).

[25] Paula Hawthorne, as quoted in Janet Abbate, *Recoding Gender* (MIT Press, 2012).

subsequent generations of male programmers, this early association of programming with women's work would be both threatening and demeaning.

Computer Cowboys

The key assumption behind the association of computer programming with women's work was the notion that the work involved was boring and uncomplicated. The very first written manual on computer programming, published in 1947 by Herman Goldstine and John von Neumann, based on their experience with the ENIAC project, carefully distinguished between the work of the "planner," who did the intellectual labor of analyzing a problem and deciding on a mathematical approach to its solution, and the "coder," who was responsible only for transcribing the thoughts of the planner and mechanically translating this solution into a form that the computer could understand.[26]

In practice, however the planner/coder distinction quickly broke down, even in the case of the ENIAC, and the work of the (female) coders became entangled with the intellectual operations originally carried out by the (male) planners.[27] The tasks that the ENIAC programmers performed turned out to be unexpectedly difficult, requiring the development of creative new techniques, further blurring the boundary between the machine's design and its operation, and that across the hardware-software dichotomy. The same proved to be true at other early electronic computing projects. The limitations of early hardware devices often meant that what was expected to be a simple programming project would quickly escalate into a full-blown research excursion. The optimistic assumption of many computer department managers that programming was simply a matter of having a low-status "coder" implement the plan sketched out by a "planner" was revealed to be simplistically naive.

The intellectual and technical challenges associated with programming a computer increased as the computers themselves became increasingly powerful. By the early 1960s companies like IBM and Remington Rand UNIVAC were manufacturing relatively low-cost, reliable, and mass-produced electronic computers that were economically competitive with earlier forms of information technology. Not only were there many more firms in need of programmers, but also the types of problems that these programmers were being called upon to solve were increasingly varied and complex. Whereas the first generation of experimental electronic computers were largely used for scientific purposes, commercial machines were designed for business applications. Defining an algorithm to solve a differential equation might be difficult, but paled in comparison to the challenge of constructing a computerized accounting system. By the end of the 1960s, the perception of computer programming had changed so dramatically that *Fortune* magazine was declaring it to be "brain business," an "agonizingly difficult intellectual

26 Herman Goldstine and John von Neumann, *Planning and Coding of Problems for an Electronic Computing Instrument* (Institute for Advanced Study, 1947).

27 Ensmenger, *The Computer Boys Take Over*.

effort" that demanded an unusual degree of creativity, insight, and ability.[28]

This newly found appreciation for computer programmers, combined with an increasing demand for their services, was accompanied by an equally dramatic rise in their salaries. Estimates from the mid-1960s suggested that although there were already 100,000 programmers working in the United States alone, there was an immediate demand for as many as 500,000 more.[29] One of the leading industry analysts, in a 1967 article on the "persistent personnel problem" in programming, predicted that salaries for programmers would rise 40-50% over the course of the next 4-5 years. [30] A talented programmer could not only command a high salary, but also possessed an unusual degree of mobility. Annual employee turnover rates in many programming departments were as high as 25%, as programmers of even average ability moved in pursuit of improved compensation of more challenging or interesting projects.[31]

The elevation of both the status and pay scale of computer programmers attracted an increasingly number of men to the occupation. Some of these men drifted in from disciplines with intellectual affinities to computing, such as mathematics, philosophy, or electronic engineering. Others entered via corporate computerization efforts, and had backgrounds in traditional business specialties. In either case, these recent converts to computing brought with them the traditions, practices, and status hierarchies of their former disciplines, often attempting to recreate them in their newly discovered discipline. For them, the lingering association of computer programming with the feminized activities of "coding," corporate data processing, and other forms of clerical work was a source of perpetual career anxiety.[32]

One strategy for dealing with this anxiety was to appeal to traditional forms of masculine identity. For computer programmers, these appeals were most often made in the context of representations of skill: if the problem with programming, at least from an occupational status perspective, is that it was considered to be straightforward and mechanical, then the solution was to reframe the occupation as being active, creative, and unpredictable. As has already been mentioned, programming was often a difficult and intellectually challenging activity. Every computer configuration was unique, and solutions to problems were often necessarily local and specific. The difference between a program that ought, in theory, to work, and one which actually functioned with an acceptable degree of performance was often a function of the idiosyncratic skills and abilities of an individual programmer. For example, on the IBM 650 computer, which was the first of the truly mass-produced computers, (often referred to as the "Model-T" of computing) the main memory was a rotating metal drum covered in magnetic oxides. In order to optimize performance of read-write operations, a skilled programmer would time the operation of critical operations to coincide with the exact moment

[28] Gene Bylinsky. "Help wanted: 50,000 programmers." *Fortune* 75 (1967):141–168.

[29] Stanley Englebardt, "Wanted: 500,000 Men to Feed Computers." *Popular Science* (Jan 1965).

[30] Richard Canning, "The persistent personnel problem." *EDP Analyzer*, 5 (1967):1–14.

[31] John Thompson, "Why is Everyone Leaving?" *Data Management* (1969): 25-27.

32 Edsger Dijkstra. "The Humble Programmer." *Communications of the ACM* 15 (1972): 859–866; RAND Symposium, "Is it Overhaul or Trade-in Time: Part I." *Datamation* 5 (1959): 24–33.

that the desired data had rotated under the read head. Getting this to work properly clearly required a great deal of skill, but what kind of skill was it? It wasn't exactly math, and certainly not a science, and most programmers did not consider what they did do be proper engineering. More often they described their work as a form of directed tinkering, a highly specialized form of puzzle solving that required not only skill and experience, but also an innate genius.

The growing realization that skill was essential to quality programming, combined with the lack of any clear understanding of the essential nature of that skill, was both an asset and a liability for programmers. When John Backus, the IBM researcher who developed the FORTRAN programming language, described computer programming in the 1950s as "a black art, a private arcane matter" in which "each problem required a unique beginning at square one, and the success of a program depended primarily on the programmer's private techniques and inventions," he was not being complementary.[33] Backus meant this description of the idiosyncratic skill and haphazard processes of the programmer to be demeaning, in that it implied that programming was not a proper science. But for many professional programmers, it was precisely this element of personal creativity—and the autonomy and control that went along with the possession of a unique and esoteric skill—that was so appealing about the discipline. To be a devotee of a dark art, or a high priest, or a sorcerer (all popular metaphors used to describe programming in this period) was to be privileged, elite, master of one's of domain.[34] It was certainly preferable to being characterized as a gloried clerical worker or a "mere" technician.

Anecdotal accounts of the special genius of certain programmers were supplemented by an emerging empirical literature on programmer performance. In one influential IBM study from the late 1960s (which is still cited today, despite its serious methodological shortcomings), suggested that the talented programmer was twenty-six times more productive than his merely average colleague. These uniquely gifted "super-programmers" were "worth an army of programmers of lesser than average caliber," argued one participant at a 1968 NATO Conference on Software Engineering. The conclusion drawn by many corporate managers was that "the major managerial task" they faced was finding and keeping "the right people."[35] "With the right people, all problems vanish."[36] It would be hard to find a more compelling endorsement of the professional power conveyed by the possession of individual expertise. A skilled programmer was effectively irreplaceable.

At first glance it might seem that this focus on individual skill would provide equal

[33] John Backus. "Programming in America in the 1950s — some personal impressions." In *A history of computing in the twentieth century: a collection of essays*, ed. Nicholas Metropolis (Academic Press, 1980): 125–135.

[34] C.A.R. Hoare, "Programming: Sorcery or science?" *IEEE Software* 1(1984):5–16; David Freedman, "Computer Magic." *Proceedings of the 11th annual Computer Personnel Research Conference* (ACM Press, 1973).

[35] Edward David, cited in Peter Naur, Brian Randall, and John Buxton, ed., *Software Engineering: Proceedings of the NATO Conferences* (Petrocelli/Charter, 1976), 33.

[36] Robert Gordon, "Personnel Selection." *Data Processing Digest* (1967): 87-88. 87.

opportunities for both men and women in the programming professions, and to a certain degree that is true. The literature from this period is full of anecdotal evidence about the former secretary or fashion-model who turned out to be an excellent programmer (along with the male mathematician who did not). Women did continue to be hired in disproportionate numbers as programmers. But there were also clearly masculine associations in the language and metaphors used to describe the idiosyncratic and temperamental character ascribed to programming professionals. Tinkering, for example, has long been gendered as a masculine approach to technology use, one in which keeping "close to the machine" was privileged over all other considerations.[37] The computer scientist and Turing Award winner Richard Hamming, for example, has suggested that "real programmers" in the late1950s often resisted the use of higher-level programming languages because they were "sissy-stuff."[38] Even the softer comparisons of computer programming to literary production (comparing programming as poetry was another common analogy) invoked traditionally masculine identities.[39] And the organizational role of "maverick," "hot shot," or "whiz kid" was likely more comfortable for men than for women.

The Right Man for the Job

The changing perception of the status of programmers, along with the rapid rise in demand and salaries, made the occupation more inviting to male participants. But the appearance of new incentives for men to enter an occupation does not necessarily mean that occupation will become masculinized. The same characteristics that made programming attractive to men attracted women as well and, more significantly, female labor remained less expensive and more tractable, in part because employers assumed women would have fewer outside opportunities and no expectations of promotion into management. Neither structural barriers, such as formal educational requirements, nor the nature of the work, such as the need for physical strength or safety concerns, blocked women's entry into the computer professions in this period. So, short of some blatant conspiracy against women, why did programming nevertheless become predominantly masculine?

One significant explanation for the dramatic sex change that occurred in the computer professions in this period has to do with the mechanisms that employers developed to identify and recruit potential programmers. The combination of a massive rise in demand with a general lack of agreement about what made for a good programmer posed a real problem for the computer industry. In the mid-1950s employers convened a series of conferences to strategize about the looming shortage

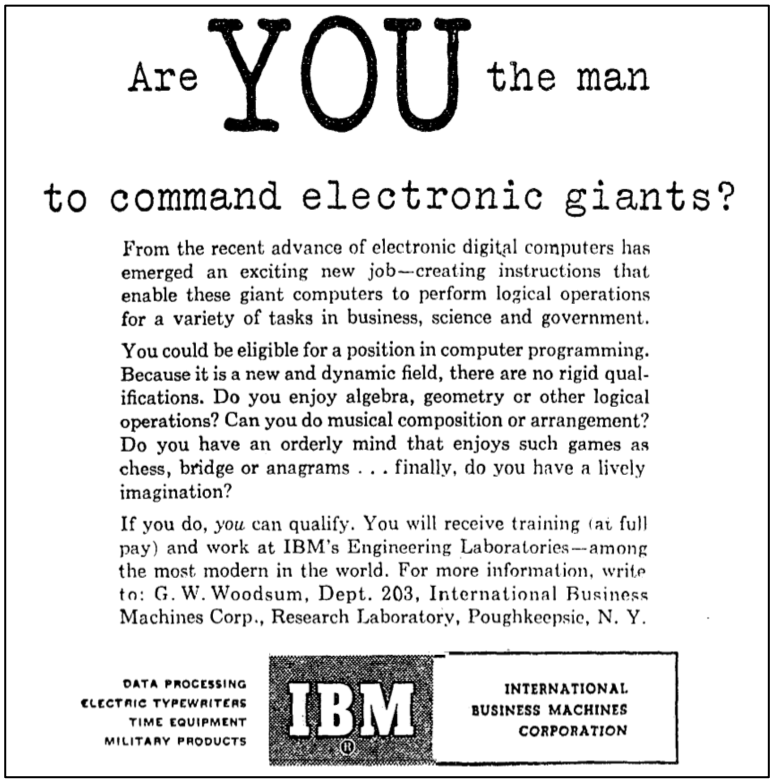
[37] Tine Kleif and Wendy Faulkner, "'I'm No Athlete but I Can Make This Thing Dance!' Men's Pleasures in Technology." *Science, Technology & Human Values* 28 (2003): 296–325; Ruth Oldenziel, "Boys and Their Toys: the Fisher Body Craftsman's Guild, 1930-1968, and the Making of a Male Technical Domain." *Technology and Culture* 38 (1997): 60–96.

[38] Richard Hamming, *The Art of Doing Science and Engineering*. (Gordon and Breach, 2005).

[39] Frederick Brooks, *The Mythical Man-Month* (Addison, 1982).

of trained programmers, and a few years later were talking about the "problem of programming personnel" in terms of a full-blown crisis. The problem was not a lack of workers willing to enter the well-paying and fast-growing new computer industry, but rather the absence of mechanisms for identifying "qualified" programmers. It was one thing to recognize, as did G.T. Hunter of the IBM Corporation, the need for programmers "who were above average in training and ability", but another to specify what kind of training, and what kind of abilities? Prior to the late 1960s there were no formal academic programs in computer science, and even after such programs were established, they never provided more than a small fraction of the programmers required by industry. And the emerging research on programmer performance, although it reinforced the notion of individual genius, provided little in terms of useful guidance for employers about how to identify and develop programmer talent.

Employers struggled with the difficult task of identifying the special "twinkle in the eye" or "indefinable enthusiasm" that separated the genuinely skilled programmer from his or her merely average colleague.[40] As a result the job advertisements from this period are full of such appeals to such vague signifiers of potential programming ability as an affinity for puzzle-solving, chess-playing, and musical performance.


Are YOU the man

to command electronic giants?

From the recent advance of electronic digital computers has emerged an exciting new job—creating instructions that enable these giant computers to perform logical operations for a variety of tasks in business, science and government.

You could be eligible for a position in computer programming. Because it is a new and dynamic field, there are no rigid qualifications. Do you enjoy algebra, geometry or other logical operations? Can you do musical composition or arrangement? Do you have an orderly mind that enjoys such games as chess, bridge or anagrams . . . finally, do you have a lively imagination?

If you do, *you* can qualify. You will receive training (at full pay) and work at IBM's Engineering Laboratories—among the most modern in the world. For more information, write to: G. W. Woodsum, Dept. 203, International Business Machines Corp., Research Laboratory, Poughkeepsie, N. Y.

DATA PROCESSING
ELECTRIC TYPEWRITERS
TIME EQUIPMENT
MILITARY PRODUCTS

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

Figure 1: IBM Advertisement (New York Times, May 31, 1969)

40 Datamation Report. "The Computer Personnel Research Group." *Datamation* 9 (1963): 38–39.

In the absence of clear entry paths into computer programming, employers struggled to create their own tools for recruiting promising potential programmers. The dominant strategy that emerged over the course of the 1960s was the use of psychometric testing (aptitude tests and personality profiles) aimed at identifying trainees who possessed the right stuff to be skilled programmers. Perhaps more than any other factor, it was the use of such testing instruments that contributed to the masculinization of the programming professions, particularly in the corporate environment.

The use of aptitude tests in the programming industry was pioneered in the late 1950s by the Systems Development Corporation (SDC), the RAND Corporation spin-off that was responsible for developing the software for the United States Air Force SAGE air defense system. In 1952 IBM was hired to build the real-time computers that were to serve as the heart of the SAGE network, and in 1955 they contracted with the RAND Corporation to develop the software for these new machines. This was a programming job of unprecedented size and complexity, and within a year there were more programmers at RAND than any other type of employee. When RAND spun-off their software division as SDC, it instantly became the single largest employers of programmers in the world. The SDC staff of 700 programmers representing three-fifths of the available programmers in the United States, and over the course of the next five years, SDC would hire or train 7,000 more.

In order to cope with its sudden demand for programmer trainees, SDC borrowed heavily from the psychometric testing techniques originally developed by the United States military to mobilize large numbers of soldier recruits. Such tests had been adapted for industry use in the 1930s, and in the post-war period had been embraced by the emerging "science" of personnel research.[41] SDC adapted its early programmer aptitude tests from the work of the influential psychologist and psychometrician Louis Thurstone.[42] The central assumption of the Thurstone approach was that for every occupation there were a particular set of personality traits, cognitive skills, and emotional characteristics that could be positively correlated with superior performance. In the absence of well-established measures of actual on-the-job performance, these characteristics could be used to predict aptitude and affinity. This was the idea, at least, and SDC embraced the use of aptitude tests wholeheartedly, if only because it provided at least some method of screening large numbers of potential employees.

Other employers soon adopted similar practices. Beginning in 1955, the IBM Corporation, in order to help alleviate its clients' desperate quest to recruit programmers, developed the IBM Programmer Aptitude Test, or PAT. By the early 1960s, 80% of all employers used some form of aptitude test, and more than half of these relied on the IBM PAT.[43] In 1967 alone, more than 700,000 aspiring programmers took the PAT, and for at least another decade such tests served as the

41 Henry Eilbert. "The Development of Personnel Management in the United States." *Business History Review* 33 (1959): 345–364.

42 T.C. Rowan. "Psychological Tests and Selection of Computer Programmers." *Journal of the ACM* 4 (1957): 348–353.

43 Charles Lawson. "A Survey of Computer Facility Management."*Datamation* 8 (1962): 29–32.

de facto gateway into the programming profession.[44]

On the surface, this reliance on aptitude testing seems gender neutral, or perhaps even advantageous to women. Many employers administered the test as broadly as possible, believing that they were just as likely to discover programming talent in a secretary as in an engineer or mathematician. After all, you were either born with the aptitude or you were not. As one contemporary article on opportunities for women in computer programming described it, there was "no sex discrimination in hiring" in computing, because "If a girl is qualified, she's got the job." As long as a woman could pass the employer's aptitude test, it didn't matter whether "she was just out of school, or a senior systems engineer …She just had to have the aptitude."[45]

There are indications, however, that below the surface of this seeming meritocracy lurked a series of gender biases. To begin with, these tests emphasized mathematical ability, with most of the questions dealing with number series, figure analogies, and arithmetic reasoning, this despite the fact that by this period most employers no longer perceived any direct relationship between mathematical training and programming ability.[46] But the kinds of questions that could be easily tested using multiple-choice aptitude tests necessarily focused on mathematical trivia, logic puzzles, and word games. The test format simply did not allow for any more nuanced or meaningful or context-specific problem solving. And, in the 1950s and 1960s at least, such questions did privilege the typical male educational experience. (Among other things, as such tests became the passport into the increasingly lucrative field of computer programming, cheating became rampant. Possessing an advance copy of the test almost certainly guaranteed success, and traditional "old boy" networks such as fraternal organizations and social clubs circulated stolen copies.)

Even more gender-biased were the personality profiles that were developed to complement the aptitude tests. The use of such profiles also originated at SDC, and were based on the Strong Vocational Interest Bank (SVIB), which had been widely used in vocational testing since the late 1920s. The basis for the SVIB was a series of four hundred questions aimed at elicit an emotional response (such as "like", "dislike", or "indifferent") to specific occupations, work and recreational activities, types of people, and personality types. Over the course of the late 1950s and 1960s, psychometricians at SDC developed — and published in serious academic journals such as the *Journal of Applied Psychology* and *Personnel Psychology* — the SVIB "keys" they had developed to track the "vocational interests of computer programmers."[47] These personality profiles were used alongside aptitude tests to identify promising programmer trainees at SDC. They also served to establish, in the collective lore of the computer industry, the cultural stereotypes that would come to characterize the

44 W. J. McNamara. "The selection of computer personnel: past, present, future." *in Proceedings of the fifth SIGCPR conference on Computer Personnel Research* (ACM Press, 1967): 52–56.

45 Lois Mandel. "The Computer Girls." *Cosmopolitan* (April 1967): 52–56.

46 William Paschell, *Automation and employment opportunities for office workers; a report on the effect of electronic computers on employment of clerical workers* (Bureau of Labor Statistics, 1958).

47 Dallis Perry and William Cannon, "Vocational Interests of Computer Programmers." *Journal of Applied Psychology* 51 (1967): 28–34.

computer programmer for the next several decades.

For the most part, the personality keys associated with programmers were unremarkable: they enjoyed their work, disliked routine and regimentation, and were especially interested in problem and puzzle-solving activities. The programmer keys resembled somewhat those developed for chemistry and engineering but, interestingly enough, not those of physics or mathematics. Otherwise, the ideal personalities of programmers resembled other white-collar professionals in such diverse fields as optometry, public administration, accounting, and personnel management.[48]

In fact, there was only one really "striking characteristic" about programmers that the SDC researchers identified. This was "their disinterest in people." Compared with other professional men, "programmers dislike activities involving close personal interaction. They prefer to work with things rather than people." In a subsequent study, Perry and Cannon demonstrated this to be true of female programmers as well.[49]

The actual scientific support for this claim that programmers were exception in their lack of people skills was sketchy at best, but corresponded well with anecdotal evidence. As the computer industry analyst Richard Brandon suggested in a 1968 article, the programmer type was "excessively independent," often to the point of mild paranoia. He was "often egocentric, slightly neurotic, and he borders upon a limited schizophrenia. The incidence of beards, sandals, and other symptoms of rugged individualism or nonconformity are notably greater among this demographic group." Tales about programmers and their peculiarities "are legion," Brandon argued, and "do not bear repeating here."[50]

Why such stories were legion is an open question. There were some structural reasons why programmers in this period might have been perceived as scruffy and anti-social mavericks, at least by their white-collar co-workers: for a variety of technical and economic reasons, programmers would often work odd hours and overnight shifts, meaning that on the occasions when they were visible to other employees, they were often unshaved and bedraggled.[51] But there is also evidence that their reputation as outsiders might also have been exaggerated. As I have described elsewhere, many corporate managers in the 1960s were wary of the growing power and authority of the "computer boys," and resented the degree to which technical expertise increasingly seemed to trump more conventional forms of business knowledge and experience.[52] By emphasizing the personal eccentricities and alleged social awkwardness of computer specialists, traditional managers

48 Perry and Cannon, "Vocational Interests of Computer Programmers."

49 Perry and Cannon, "Vocational Interests of Female Computer Programmers." *Journal of Applied Psychology* 52 (1968).

50 Richard Brandon, *"The problem in perspective."* In *Proceedings of the 1968 23rd ACM national conference* (ACM Press, 1968): 332–334.

51 These unusual hours often posed particular barriers to women, as many employers in this period had explicit rules against women being on premise after hours. See Weinberg, *The Psychology of Computer Programming*.

52 Nathan Ensmenger, "Letting the `Computer Boys' Take Over: Technology and the Politics of Organizational Transformation." *International Review of Social History* 48 (2003): 153–180.

sought to limit their influence. "Too frequently these people [programmers], while exhibiting excellent technical skills, are non-professional in every other aspect of their work.''[53]

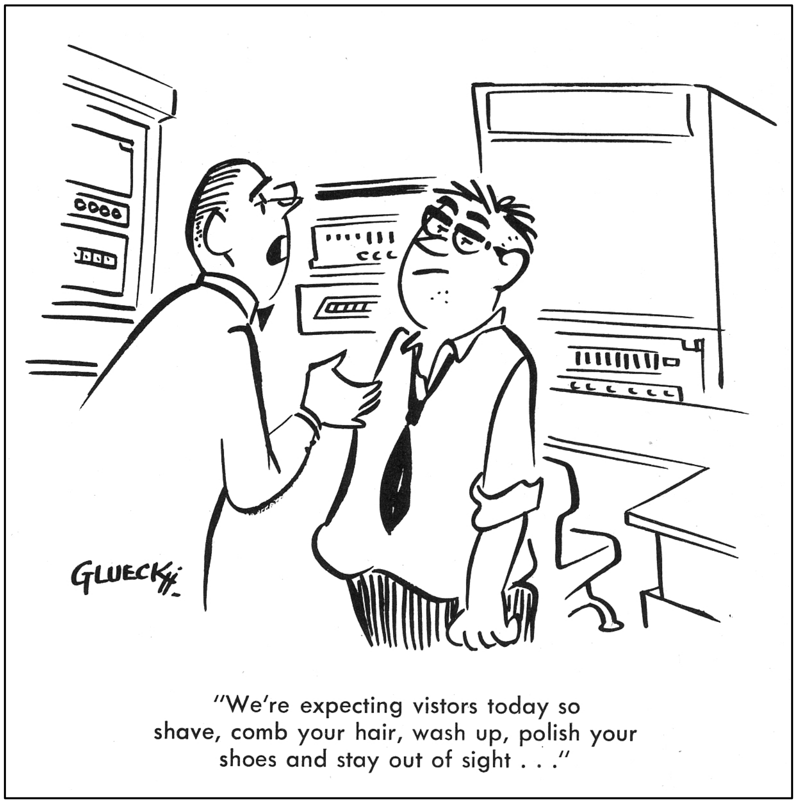

GLUECK
"We're expecting vistors today so shave, comb your hair, wash up, polish your shoes and stay out of sight . . ."

Figure 2: Datamation Magazine, 1963

It is also possible, of course, that there is something about computer programming as a technical activity that makes it particularly attractive to individuals seeking to escape from the complexities of "real world" social interactions into the comforting simplicity of the virtual micro-world contained within a computer, and that such individuals might also express their social non-conformity in their personal appearance. This is certainly the conclusion drawn by scholars such as Turkle, and is the dominant consensus of popular culture.

The prevalence of what would soon come to be known as "nerds" in computer programming might also, however, be an unintended consequence of what Richard Brandon described as the "Darwinian selection mechanisms" of the computer industry hiring practices. In this interpretation, the embodiment in the hiring practices of the industry of the ideal of the "detached" (read anti-social, mathematically inclined, and male) programmer created a feedback cycle through

[53] Malcom Gotterer, "The Impact of Professionalization Efforts on the Computer Manager." in *Proceedings of 1971 ACM Annual Conference* (ACM Press, 1971).

which anti-social, mathematically inclined males became over-represented in the programmer population, which in turn reinforced the original perception that that programmers *ought* to be anti-social, mathematically-inclined, and male.

Whatever the reasons for its origins, the association of masculine personality characteristics with a presumed inherent programming ability helped created an occupational culture in which female programmers were seen as exceptional or marginal. Only by behaving less "female," could they be perceived as being acceptable. Many women still did continue to be hired as programmers and other computer specialists, but they did so in an environment that was becoming increasingly normalized as masculine, and in which the selection mechanisms privileged male candidates. Even today, companies such as Google and Microsoft are notorious for their reliance on confrontational interview techniques in which logic and math puzzles play a prominent role — despite the substantial evidence that such techniques are severely gender biased.[54]

Identity Crisis?

Although the growing emphasis on individual genius and manly mastery provided many computer programmers with useful resources with which to protect their autonomy and authority, concerns about status and identity continued to plague some programmers (and in this case, male programmers in particular). Although there were plentiful opportunities for horizontal mobility within the industry—even a moderately skilled programmer would never lack for a programming position—opportunities for vertical mobility up through the ranks was difficult if not impossible. The emerging stereotype of the programmer personality as being "disinterested in people" hindered their advancement into more managerial positions, as did their general lack of formal educational or professional credentials. By the end of the 1960s, the "question of professionalism" had become a subject of frequent debate within the computing literature.[55] While many programmers continued to relish their role as technological savants, others pursued more mainstream approaches to establishing a professional monopoly of competence. These more corporate- or academically-oriented aspiring computer professionals, the majority of them male, worked to establish professional societies, publish academic journals, develop credentialing programs, and lobby employers and governments for recognition and legitimacy.

As Margaret Rossiter and others have demonstrated, professionalization generally implies masculinization.[56] Consider for example the Data Processing

[54] William Poundstone, *Are You Smart Enough to Work at Google? Trick Questions, Zen-like Riddles, Insanely Difficult Puzzles, and Other Devious Interviewing Techniques You Need to Know to Get a Job Anywhere in the New Economy* (Hachette Digital, 2012).

[55] Nathan Ensmenger. "The 'Question of Professionalism' in the Computer Fields." *IEEE Annals of the History of Computing* 4 (2001): 56–73.

56 Margaret Rossiter. *Women scientists in America : struggles and strategies to 1940* (Johns Hopkins University Press, 1982); Jeffrey Hearn, "Notes on Patriarcy, Professionalization and the Semi-Professions." *Sociology* 16 (1982).

Management Association (DPMA), which in the early 1960s established the Certified Data Processor (CDP) program, which was modeled after the widely recognized Certified Public Accountant (CPA). In the case of the CDP program, the masculine bias of professional standards were particularly apparent: the requirement of formal educational credentials, a minimum of three years industry experience, and the possession of "high character qualifications" (the specifics of this requirement were vague, and rarely enforced, but appeared to involve letters of recommendation from other established "professionals") privileged not only males, but males with an established commitment to a corporate managerial culture. The majority of CDP holders were middle managers, an organizational role that was often explicitly denied to women in this period, or at the very least was implicitly associated with masculine characteristics.[57] The more computer professionals were seen as not just technical experts, but also potential corporate managers, the more women were excluded.

The principle alternative to the business-oriented DPMA was the Association of Computing Machinery (ACM), which was founded in 1947 as an outgrowth of an academic conference, and which continued afterwards to focus on the concerns of professional academics. As might be expected from an explicitly academically oriented professional society, the ACM was even more stringent in its educational requirements. In 1965, a period when the ratio of male to female college undergraduates was close to 2-1, it imposed a strict four-year degree requirement for its members.[58] The ACM was also notorious for its disdain for business-oriented programmers, and in turn was castigated by many working programmers as "dominated by, and catering to, Ph.D. mathematicians."[59] There were obviously even fewer female Ph.D. mathematicians than there were women with undergraduate degrees. To the extent that belonging to the ACM or possessing a computer science degree was considered an essential component of being a "professional" programmer, programming was increasingly a masculine identity. A survey from the late 1970s showed that fewer than 10% of ACM members were women.[60]

The ACM was also responsible for setting the agenda for the emerging discipline of computer science. A comprehensive scholarly history of academic computer science has yet to be written, but for the purposes of this essay it is sufficient to note only that A) the institutionalization of computer science as an academic discipline was well-underway by the late 1960s and B) that it involved a turn towards the theoretical, the mathematical, and the abstract. This latter agenda sometimes alienated computing practitioners and industry employers, who criticized the computer scientists for being "too busy teaching simon-pure courses in their

57 Marie Hicks, "Meritocracy and Feminization in Conflict: Computerization in the British Government." In *Gender Codes: Why Women are Leaving Computing*. Ed. Thomas Misa (Wiley, 2010).

58 Claudia Goldin, Lawrence Katz, and Ilyana Kuziemko. "The Homecoming of American College Women: The Reversal of the College Gender Gap." *Journal of Economic Perspectives* 20 (2006): 133–156.

59 Datamation Editorial. "The Cost of Professionalism." *Datamation* 9 (1963): 23.

60 Thomas D'Auria*,* "ACM membership profile report." *Communications of the ACM* 20 (1977): 688–692.

struggle for academic recognition," but the pursuit of theory and abstraction were effective strategies within the academy, and the ACM quickly became dominated by those who perceived their professional identity in terms of the academic research scientist.[61] This identity was less accessible to women and other minorities, whose participation rates in both academic computer science and academically oriented professional societies were lower than their rate of participation in the computer industry more generally.[62]

It is important to note that although the academic discipline of computer science was indeed masculine, it was masculine in ways that were typical of most of academia in this period. The traditional masculinity of the academic professions had little to do with the distinctively gendered nature of computing in the corporate world. To the degree that computer scientists were decried as "eggheads" divorced from the needs and realities of the "real world", it was in terms of the traditional critique of academics as "ivory-tower" types that had little to do with the emerging stereotype of the "computer cowboy" or "whiz kid." In fact, in many respects the academic computer science persona was cultivated in direct opposition to the emerging stereotypes of the computer programmer as an idiosyncratic genius. What the academic computer scientist wanted was to establish his discipline on a firm foundation of theoretical knowledge.[63] The long-standing association of computer programming with individual aptitude, machine-specific techniques, and arcane knowledge was anathema to the computer scientist. It was, after all, an MIT professor of computer science who launched the first major attack against the emerging phenomenon of the "computer bum." These compulsive and unsystematic tinkerers, no matter how brilliant, represented everything that the rigorous and conscientious computer scientist was not. That the emergence of the pathologically undisciplined "computer bum" was a direct consequence of the academic institutionalization of computer science is therefore a particularly delicious irony.

Computer Centers

The "computer bum" of the late 1970s superficially resembled his corporate cousin, the "computer boy." He too possessed a skill that was innate, idiosyncratic, and individual—to the point of being as much a personality type as an aptitude. He too was scruffy and unkempt, anti-social, and out-of-sync with the prevailing organizational norms of professional behavior. And finally, he too was "more interested in machines than in people," and in mastering technology for pleasure rather than in the pursuit of some larger purpose. But although the computer bum represented the extreme end of a spectrum that had already been defined in the corporate setting, this particular extreme could only be achieved *outside* the corporation. The computer bum of the late 1970s was the product of a distinctive

[61] Hal Sackman, W.J. Erickson, and E.E. Grant. "Exploratory experimental studies comparing online and offline programming performance." *Communications of the ACM*, 11(1968):3–11.
62 D'Auria, "ACM membership profile report."
63 Dijkstra, "The Humble Programmer." *Communications of the ACM* 15 (1972): 859-866.

combination of technology and place, a combination that was unique to the research university but which developed outside of, or perhaps parallel to, academic programs in computer science. Without the computer lab, the computer bum would not have existed. In these unconventional and unruly places, bright young students were allowed almost unlimited—and largely unsupervised—access to cutting-edge experimental electronic digital technologies, and where the already gendered stereotypes associated with computer culture would become inextricably linked to adolescent masculinity. The norms, ethos, and practices established in the university computer center in the 1970s formed the basis for the emergent computer hobbyist culture of the 1980s (and beyond) and would be perpetuated and recreated in similarly masculine spaces, from the bedrooms of pimply teenage computer hackers to the couches and erstwhile dormitories of innumerable Internet start-ups, to the studiously informal work/playgrounds of corporate "campuses" at Apple, Microsoft, and Google, where free sodas and foosball tables are seen as being as essential to the production of software as product labs and computer workstations.

The computer center was a social and technological space unique to the Cold War research university, although its origins predate the advent of the electronic digital computer. Beginning in the early 1930s, several major research universities had established, often in collaboration with equipment manufacturers, computational service bureaus aimed at providing computational support for scientific researchers. It would be these computer centers that built (or later purchased) most of the early electronic computers, and in many cases, the first formal academic training in electronic computing was provided through these centers, rather than via traditional departments.

Even after the establishment of independent computer science departments, a separation of computer operations from computer science research was typical of most universities. In part this represented the logic of capital: it was difficult and expensive to purchase and operate a large-scale computer facility (a situation that remains true today), and so it made sense for universities to centralize computing and distribute the costs across multiple departments. But it was also true that the nascent discipline of computer science was not particularly interested in controlling its own computing resources. In fact, computer scientists worked hard to distance themselves from "service" connotations of the computer center, and indeed, from any association with actual computers. After all, one of the strongest objections made to the establishment of their discipline in the first place was that what they did was not science at all, but technology. It was in their professional interest to focus on the computer as a logical abstraction rather than an embodied technology. The last thing that research-oriented computer scientists wanted to be associated with were the "mere technicians" who tended the machinery, which explains both the continued existence of the autonomous computer center and the great antipathy with which academic researchers viewed the activities of the "computer bums" with whom these centers were increasingly identified.

In their physical configuration, the academic computer center closely resembled its nearest cousin the corporate data processing department: the size, expense, and power requirements of computers in this period demanded the construction of dedicated computer rooms with raised floors, reinforced cooling systems, and securely locked doors. But whereas in the corporate context the enforced segregation of computer equipment and personnel served to reinforce the elite and privileged status of the computer experts—the literature from this period is replete with references to "high priests" of computing carefully controlling access to the "air-conditioned holy of holies" of the computer room—the marginal location of the computer center encouraged experimentation and exploration. In this sheltered but unsupervised environment, the links between electronic computing and the culture and practices of adolescent masculinity would be firmly established. During the day the university computer centers were run by staff technicians in the service of faculty research projects. At night, however, the computer centers were turned over to the use of undergraduates, either explicitly or with the implied consent of the faculty and administration. It was the after-hours activities of unofficial computer enthusiasts that would establish the distinctive computer "hacker" identity.

The association between the social architecture of the computer center and the expression of the computer bum personality was first made public by the psychologist Lucy Zabarenko and her colleague Ellen Williams at the 1971 ACM Conference on Personnel Research. In doing their empirical research on programmer education, Zabarenko and Williams had noticed a "special cultural phenomenon" distinctive to the computer center—a culture so distinctive that they thought it worthy of further study by anthropologists.[64] There was something "especially compelling" about nature of computer programming, they argued, that absorbed its practitioners to such as degree that they lost their sense of time and place. In their quest to "get [time] on" the machine, the inhabitants of the computer center stayed up late at night, slept all day, and lost all interest in their other academic work. Their obsession would cause them to neglect their bodies, to the point that "many of these men appeared poorly nourished and all were thin," subsisting as they did "mainly on coffee and carbohydrates." These practices, originating from necessity, soon became part of the "invariant custom" of the "computer bum," who increasingly associated only with others of his kind, making it a point "to be informally dressed, elaborately unaware of time, and constantly underfed." For Zabarenko and Williams, the computer bum was an unsavory character, one that threatened, rather than advanced, the advancement of computer technology. "Can we teach young children computer skills," they worried, "without also transmitting the beliefs and values of the computer center?" They believed the pervasive presence of the disheveled computer bums in the computer center would deter more "normal" programmers.

We have already seen how Stewart Brand, just a year after the publication of the Zabarenko and Williams report, provided a radically different assessment of the relative virtues and vices of the computer bum culture. But Zabarenko, Williams, and Brand (and, just a few years later, Joseph Weizenbaum) were in surprising agreement about the nature and causes of the phenomenon. What made the computer bum possible was not simply the availability of computer technology, but

64 Lucy Zabarenko and Ellen Williams. "The Computer Center as a Subculture." *Council on Anthropology and Education Newsletter* 2 (1971): 5–8.

the combination of technology, culture, and environment. This was a combination peculiar to the university computer center. It did not exist within the corporate data processing department, despite their apparent similarities.

Three features of the academic computer center significantly contributed to the formation of its unique culture. To begin with, the computer center was an isolated, and therefore largely unsupervised, environment where students had an unprecedented degree of access to the equipment. In the corporate setting even the most ardent computer enthusiasts were limited in their ability to engage directly with the machine. Rarely if ever was this access individual or unmediated. After hours in the university computer center, it was possible to exercise what came to be known as the “hands-on imperative” (a practice that would later be elevated to the status of central tenet of the “hacker ethic” in a popular and sensational account of the history of the computer center at MIT, revealingly entitled *Hackers: Heroes of the Computer Revolution*). Even if direct access to the machine was officially forbidden, motivated and creative student programmers could usually find a way. At MIT, for example, the long-standing tradition of “lock hacking” proved a useful resource to a new generation of aspiring “computer hackers.” In an era of mainframe computers that occupied an entire room, this was as close as you could get to the experience of a “personal computer.” It is no wonder that computer centers tended to attract the type of personality who found one-on-one interactions with a computer particularly compelling.

Secondly, the students who frequented the computer center were sheltered from the economic realities—and consequences—of their actions. In the corporate world computer time was expensive and therefore carefully rationed and monitored. In addition, corporate programmers were being paid for their work, and as such were accountable to managers, budgets, and schedules. Student programmers, on the other hand, were largely free to pursue their own interests, agendas, and aesthetics. This last was particularly significant: while industry employers had long complained that graduates of computer science programs had only learned to write “trick programs” rather than real applications, the code that the computer bums obsessed over did not generally serve a pedagogical purpose and were rarely related to their academic studies; in fact, the very best of the bums were notorious for not completing their course work, even when it related to their computer science curriculum. Quite a number failed out of university—but nevertheless continued to frequent the university computer labs. In an era in which many academic computer centers were saturated with grant money (largely from the Department of Defense), a skilled computer bum could pick up enough work to support his habit almost indefinitely. And, in stark contrast to the present era, the work that went on in the computer center was not intended to kick start a commercial project. The goal of becoming the next Steve Jobs or Mark Zuckerberg would not become the dominant obsession of the aspiring computer nerd until a later generation.

When the bums in the computer centers did write programs, they were often for trivial purposes or for problems that had already been solved effectively. But what interested the computer bums was not utility, but creativity, elegance, and competition. For example, one popular challenge was to attempt to solve a given problem in as few instructions as possible. Programmers would spend hours, even days, eliminating (or "bumming," as it was called) a single of code. Whether the resulting program ran quickly or efficiently, or even solved some useful or interesting problem, was irrelevant. What mattered was that the code was beautiful according to the prevailing aesthetic of the community. The truly elegant program listing would be "bummed to the fewest lines so artfully that the author's peers would look at it and almost melt with awe."[65] These listings would be passed around the computer center to be shared, admired, envied. To win such a competition was to establish one's mastery of the machine, and to establish one's place within the community hierarchy.

This brings us to the last of the three features of the university computer center that made it so distinct and significant, and which was noted by all of its observers, whether with admiration or disdain: despite the stereotype of the computer person as individualistic and "disinterested in people," the computer center was a profoundly social space. To be sure, the computer bums came to the computer center to indulge their fascination with the machine, and it was in part the machine that kept them glued to their screens and keyboards. But they were more than simply working alone, together. In practice, computer centers were abuzz with conversation and other forms of social interaction.

In fact, in his 1971 analysis of *The Psychology of Computer Programmers*, Gerald Weinberg used the sociology of computer center as an argument for study programming as a social activity. He found that even small disruptions to the social and spatial networks of the center (for example, the relocation of the soda machine) proved enormously disruptive to learning and productivity. Programmers learned through conversation, by watching one another code, in telling one another stories over Chinese food at three in the morning. Even practical jokes and pranks could serve a purpose: Stewart Brand, for example, relates the story of an MIT hacker who wrote a program called 'The Unknown Glitch,' "which at random intervals would wake up, print out I AM THE UNKNOWN GLITCH. CATCH ME IF YOU CAN, and then it would relocate itself somewhere else in core memory, set a clock interrupt, and go back to sleep." Searching for the Glitch was at once a form of collective entertainment, a lesson in computer architecture, and a rite of passage. Although in the sheltered womb of the computer room the computer bums might be isolated from the outside world, they were in intense interaction with one another.

The incorporation of video display units into computer terminal technology, which began in the 1960s, created new opportunities for socialization within the computer center. Hackers could now demonstrate their programs to others more readily, and tinker with the computer's graphical capabilities. As Steward Brand noted, they could develop, and then play against one another, competitive games such as Spacewar. The virtual violence of the computer video game, at this point available only within the confines of the university computer center, provided the link between abstract and disembodied activities of the computer hacker with more traditional, physical forms of masculine competition. For example, in one popular

65 Levy, *Hackers : heroes of the computer revolution*.

computer center competition, which was known as "sport death," programmers challenged one another to push the limits of sleep deprivation. As one self-reflective male hacker noted, "women are not so into sport death."[66]

Finally, these graphical displays could be used to display pornography. The earliest documented computer "girlie pics" date from the mid-1950s, but no doubt this was just the first of many. In fact, one widely distributed digital scan of a 1960s-era Playboy pinup, the so-called "Lena image," became a reference image for researchers in computer graphics, and has been cited in hundreds of academic papers. Looking at girls on computer screens—as opposed, for example, to pursuing them in real life—might be a particularly compensatory and adolescent masculinity, but it was masculinity nonetheless. And at the very least, sharing such images with your friends in the computer lab created yet another opportunity for male sociability.

To the degree that the computer center was a social environment, therefore, it was almost exclusively a homosocial environment. Again, this is a stark contrast to the corporate computing experience. Although the stereotype of the bearded, sandaled computer programmer was well in place by the late 1960s, in actual practice women were still very much present in most corporate computer departments, particularly when we include in our count computer operators and keypunch operators (by then the most feminized of computer specialties). There were certain situations, particularly when programmers worked after-hours to avoid conflict with mission-critical applications, from which women were explicitly excluded from the sanctum sanctorum (ostensibly for their personal safety), but in most corporations women represented at least 25-30% of all computer personnel. Not so in the university computer centers, particularly during the night hours when the most interesting activity was occurring. In part this was simply a reflection of the demographics of the student population—at some of earliest universities to develop computer centers, such as Princeton and Columbia, women were not even able to enroll until the late 1960s (or even later). But even as female enrollments in formal academic computer science programs increased, their participation in the informal computer center culture did not. The male camaraderie defined by inside jokes, competitive pranks, video game marathons, and all-night code-fests simply was unfriendly to a more mixed-gender social environment, a fact noted by many women who cited the male-dominated culture of the computer center as an obstacle to their continued participation in computing.[67] These centers were key sites of learning and the central nodes in the informal networks of knowledge exchange. The exclusion of women from these places is therefore significant.

Conclusions

[66] Sherry Turkle, "Computational Reticence: Why Women Fear the Intimate Machine." In *Technology and Women's Voices: Keeping in Touch*, ed. Cheris Kramarae (Pergamon Press, 1988).

67 Karen Frenkel. "Women and computing." *Communications of the ACM* 33 (1990): 34-46.

By the end of the 1970s, when Joseph Weizenbaum first published his scathing critique of the computer bum, the unique combination of technology, culture, place, and practices that had created the phenomenon was already coming to an end. The expensive mainframe computers that had justified the existence of the computer center were being replaced by smaller, personal computers. As the computer centers were reconstituted and reconfigured (socially, technologically, and institutionally) as "computer labs," they lost some of their sense of mystery, seclusion, and sacredness. But many of the norms and practices that had been established in the computer centers had become so thoroughly integrated with hacker culture that they continued to persist long after their original reasons for being disappeared. Life in the new computer labs continued to be nocturnal, despite the fact that there was no longer any real competition for computer time during daylight hours. The all-night coding sessions continued, reinterpreted as a rite of passage and a cultural marker rather than a structural necessity.

The slow accretion of cultural forms may be the best way to explain the ongoing process of masculinization in the computing professions. In a wide variety of contexts, from the corporation to the academy to the computer center, male programmers mobilized masculine identity as a means of pursuing professional recognition. The most conspicuous contemporary feature of this masculine identity, the association of computer programming with the "computer nerd" personality type, is not a reflection of the essentially gendered nature of the activity — an emphasis on individual technique, tacit knowledge, and even personality characteristics is common in many technological occupations, particularly those in the early stages of their development — but rather the byproduct of attempts by early programmers to elevate the status of their discipline. Many male programmers saw the role of the eccentric and idiosyncratic computer genius as a desirable alternative to that of a lowly, routinized, and feminized "coder." Although there were some downsides to being categorized as a "whiz kid" or a "computer boys," most particularly the stigma of being narrowly-focused, anti-social, and corporate unfriendly, this identity nevertheless provided programmers with many of the perceived benefits of professionalization: the establishment of barriers to entry to the discipline; the possession of a "monopoly of competence"; and mastery over an esoteric body of knowledge.[68] Skilled programmers might be "industrial theocrats," "prima donnas," and the "the most unmanageable and the most poorly managed specialism in our society," but they were also effectively irreplaceable.[69] Even for those programmers who aspired to more conventional professional identities the emerging stereotype of the undisciplined computer bum provided a useful foil to position oneself against. To be a professional computer scientist or software engineer was to *not* be a hacker, maverick, or bum. The existence of such amateurs was nevertheless assumed, or at least asserted, in the rhetorical

[68] Magali Sarfatti Larson, The Rise of Professionalism: A Sociological Analysis (University of California Press, 1977).

[69] Avner Porat and James Vaughan. "Computer personnel: The new theocracy — or industrial carpet-baggers?" *Personnel Journal*, 48 (1968):540–543; Herbert Grosch, "Programmers: The Industry's Cosa Nostra." *Datamation* 12(1966):202.

construction of one's chosen disciplinary agenda.

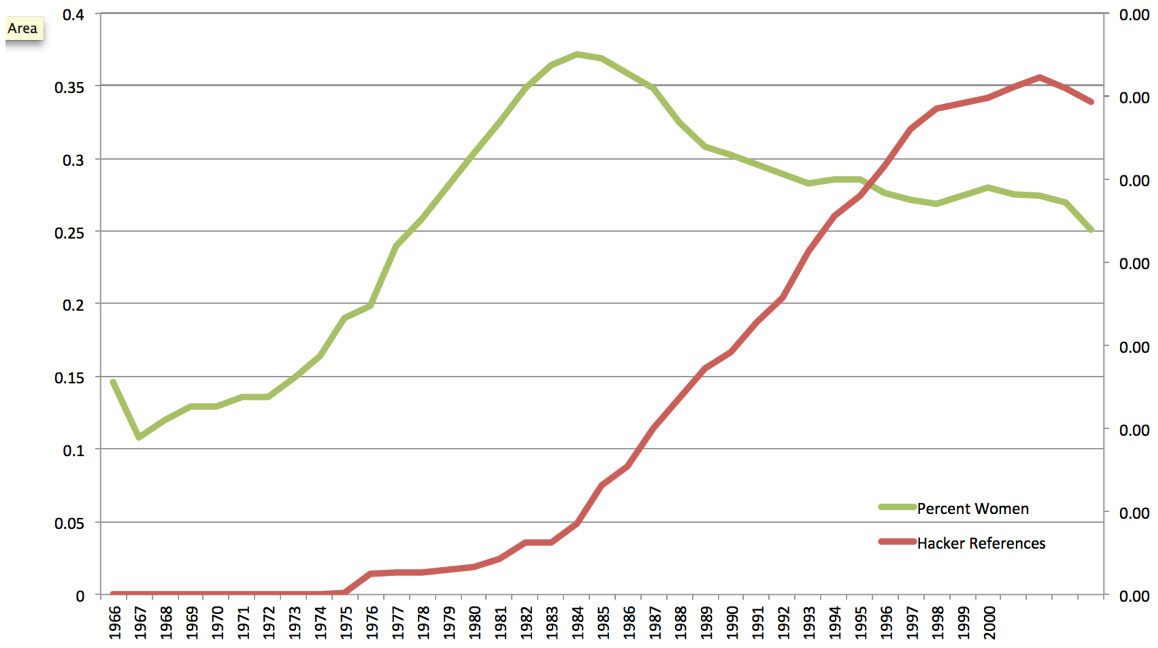


Figure 1: Female enrollments in undergraduate computer science climbed steadily until 1982, when they suddenly started declining. For years the explanation for this has been something of a mystery. The dramatic rise of media representations of the computer hacker might be the explanation.

What is different about computer programming is the degree to which such values become embedded in the mechanisms of personnel selection, the social architecture of university computer centers, and in media representations of the emerging computer culture. By the early 1980s, the personal computer had helped create a new generation of computer hacker. These hackers, the majority of them adolescent males, introduced yet another layer of masculine identities and practices to the increasingly male-dominated computing subculture, this time borrowed from ham radio and hobby electronics. These were already activities dominated by men. In fact, as Susan Douglas and Kristin Haring have convincingly demonstrated, many of the characteristics and practices that are commonly assumed to have originated in 1980s computer culture were actually well-defined decades earlier by ham radio operators, for similar reasons and via similar processes.[70] In the case of the personal computer, the intersection of computer technology, hobbyist sensibilities, and adolescent masculinity, reinforced and accelerated the widespread dissemination of a masculine hacker culture.

The personal computer elevated the computer hacker into a truly mainstream cultural phenomenon. Although actual hackers comprised only a small percentage of those who defined themselves as computer users and/or programmers, they attracted a disproportionate degree of popular attention. Prior to the 1980s, references to computer hackers were rare and largely confined to the specialist literature. Beginning in 1982, the popular media in the United States exploded with stories about young, male computer hackers. In this period, the stereotype of the computer nerd became firmly established, and came to dominate the popular imagination of what computer programmers are like, what they do, and who can (and more significantly, who cannot) participate. For many women, the computer hacker became a metaphor "for all the things they did not like about computer science: the style of work, the infatuation with computers leading to neglect of normal non-study relations, and the concentration on problems with no obvious relation to the outside world."[71] The same stories about the nature and origins of their discipline that male programmers find comforting and empowering are, for many of their female counterparts, profoundly limiting narratives.

70 Susan Douglas, *Inventing American Broadcasting,1899-1922* (Johns Hopkins University Press, 1987); Kristen Haring, *Ham Radio's Technical Culture* (MIT Press, 2006).

[71] Tove Hapnes and Knut H. Sorensen, "Competition and collaboration in male shaping of computing: A study of a Norwegian hacker culture," in *The Gender-Technology Relation: Contemporary Theory and Research*, eds. Keith Grint and Rosalind Gill (London: Taylor and Francis, 1995).